

Przysposobienie informatyczne

Michał Basa Robert Dąbrowski Marcin Engel
Janusz Jabłonowski Piotr Krzyżanowski

Warszawa, 2006

Edytor

Robert Dąbrowski

Autorzy

Michał Basa Podstawy internetu i poczty, USOS i BUW

Robert Dąbrowski Podstawy komputera, prezentacje

Marcin Engel Skład tekstu

Janusz Jabłonowski Arkusz kalkulacyjny

Piotr Krzyżanowski Strony WWW

Copyright © Uniwersytet Warszawski, Wydział Matematyki, Informatyki i Mechaniki, 2005–2006. Wydanie 1.1. Niniejszy plik PDF został utworzony 28 listopada 2006.

Skład w systemie L^AT_EX, z wykorzystaniem m.in. pakietów **beamer** oraz **listings**, w wykonaniu autorów. Szablony podręcznika i prezentacji: Piotr Krzyżanowski, projekt: Robert Dąbrowski

Spis treści

1	Wprowadzenie	12
1.1	Informacje organizacyjne	12
1.2	Cele zajęć	13
1.3	Harmonogram zajęć	14
1.4	Materiały dydaktyczne	15
1.5	Zasady zaliczania	16
2	Zasady działania komputera	17
2.1	Wprowadzenie	17
2.1.1	Popularne systemy operacyjne	17
2.1.2	Autoryzacja użytkowników	19
2.1.3	Podstawowe narzędzia	20
2.2	Zasoby komputera	24
2.2.1	Eksplorektor systemu	25
2.2.2	Katalogi i pliki	25
2.2.3	Skróty	28
2.2.4	Zasoby sieciowe	29
2.3	Panel sterowania	30
2.3.1	Klawiatura	31
2.3.2	Mysz	32
2.3.3	Ekran	32
2.3.4	Data i godzina	33
2.3.5	Opcje regionalne i językowe	33
2.4	Drukarki i faksy	34
2.4.1	Kolejka wydruku	34

2.4.2	Parametry wydruku	35
2.5	Menedżer zadań	36
3	Podstawowe aplikacje	38
3.1	Wprowadzenie	38
3.2	Eksploreator systemu	38
3.2.1	Opcje folderów	38
3.2.2	Widoki	39
3.2.3	Drzewo katalogowe	40
3.2.4	Wyszukiwanie	40
3.3	Kompresja	41
3.3.1	Tworzenie folderów kompresowanych	41
3.4	Akcesoria	42
3.4.1	Notatnik	42
3.4.2	Wordpad	43
3.4.3	Kalkulator	44
3.4.4	Paint	44
3.4.5	Gry	44
4	Przeglądarki internetowe	46
4.1	Wprowadzenie	46
4.1.1	Krótką historia przeglądarek internetowych	47
4.1.2	Protokoły komunikacyjne	48
4.1.3	Standardowe protokoły sieciowe	48
4.1.4	URL (Uniform Resource Locator)	48
4.1.5	Technologie obsługiwane przez współczesne przeglądarki	50
4.2	Używanie przeglądarki internetowej	52
4.2.1	Główne okno programu Firefox	53
4.2.2	Wyszukiwanie, panele, zakładki	53
4.2.3	Zakładki	54
4.3	Konfiguracja przeglądarki internetowej	56
4.3.1	Ustawienia ogólne	56

4.3.2	Prywatność	58
4.3.3	Parametry przeglądania	63
4.3.4	Pobieranie plików	66
4.3.5	Ustawienia zaawansowane	67
4.3.6	Protokół FTP w przeglądarce internetowej	67
4.4	Inne popularne przeglądarki internetowe	68
4.4.1	SeaMonkey	69
4.4.2	SeaMonkey Composer	69
4.4.3	Microsoft Internet Explorer	69
4.5	Bezpieczeństwo danych przesyłanych protokołem HTTP	70
4.5.1	Algorytmy szyfrujące stosowane w protokole HTTPS	70
4.5.2	Używanie protokołu HTTPS w przeglądarce Firefox	73
4.6	Wyszukiwarki internetowe	78
4.6.1	Zasada działania wyszukiwarki internetowej	78
4.6.2	Podstawy obsługi wyszukiwarki internetowej	79
4.6.3	Zaawansowane wyszukiwanie	80
4.6.4	Szukanie zaawansowane	83
4.6.5	Ustawienia wyszukiwania	84
4.6.6	Narzędzia językowe	84
4.6.7	Problemy z używaniem wyszukiwarek internetowych	85
4.7	Pobieranie dokumentów i konwersja formatów	85
4.7.1	Odczyt dokumentów formatu PDF	86
4.7.2	Tworzenie dokumentów formatu PDF	86
4.7.3	Konwersja formatów graficznych	87
5	Poczta elektroniczna	88
5.1	Wprowadzenie	88
5.1.1	Zalety poczty elektronicznej	88
5.1.2	Wady poczty elektronicznej	89
5.1.3	Bardzo krótka historia poczty elektronicznej	89
5.1.4	Mechanizmy funkcjonowania poczty elektronicznej	89
5.1.5	Adresy poczty elektronicznej	91
5.2	Konfiguracja	92

5.2.1	Parametry potrzebne do konfiguracji klienta poczty	92
5.2.2	Konfiguracja programu Mozilla Thunderbird	93
5.2.3	Używanie programu Thundrbird	97
5.2.4	(Trochę) bardziej zaawansowana konfiguracja	102
5.3	Zagadnienia związane z bezpieczeństwem poczty elektronicznej	106
5.3.1	Zagrożenia dla poczty elektronicznej	106
5.3.2	Sposoby wyeliminowania zagrożeń	106
5.3.3	Algorytmy szyfrujące w zastosowaniu do poczty elek- tronicznej	106
5.3.4	Szyfrowanie poczty	107
5.3.5	Podpisywanie poczty	107
5.3.6	Sprawdzanie autentyczności klucza publicznego	108
5.3.7	Włączenie szyfrowania w programie Thunderbird	111
5.3.8	Ochrona antywirusowa	112
5.3.9	Ochrona antyspamowa	112
5.4	Webmail	112
5.4.1	Korzystanie z Webmaila	113
5.4.2	Interfejs przykładowego Webmaila	114
6	Wstęp do arkuszy kalkulacyjnych	115
6.1	Wprowadzenie	115
6.2	Zaczynamy	116
6.2.1	Co to jest arkusz kalkulacyjny	116
6.2.2	Zalety arkuszy kalkulacyjnych	116
6.2.3	Struktura arkusza kalkulacyjnego	117
6.3	Wygląd arkusza kalkulacyjnego	118
6.3.1	Elementy interfejsu użytkownika w typowym arkuszu	119
6.3.2	Wiersz wprowadzania w typowym arkuszu	120
6.3.3	Korzystanie z pomocy	120
6.4	Podstawowe operacje	120
6.4.1	Pliki używane przez arkusz	120
6.4.2	Wpisywanie danych do arkusza	121
6.4.3	Nawigowanie po arkuszu	124

6.4.4	Obszary arkusza	125
6.5	Formatowanie arkusza	126
6.5.1	Operacje na wierszach i kolumnach	126
6.5.2	Formatowanie arkusza — komórki	126
6.5.3	Ćwiczenie z formatowania arkusza	127
6.5.4	Kopiowanie danych w arkuszu	128
6.5.5	Przenoszenie danych w arkuszu	129
6.6	Użyteczne dodatki	129
6.6.1	Tworzenie serii danych	129
6.6.2	Zabezpieczanie arkusza	130
6.6.3	Praca z dużymi arkuszami	131
7	Arkusz zaczyna działać	133
7.1	Obliczenia w arkuszu — pojęcia podstawowe	133
7.1.1	Formuły	133
7.1.2	Zasady pisania formuł	134
7.1.3	Tryby adresowania	134
7.1.4	Ćwiczenia z trybów adresowania	134
7.1.5	Operacje na datach	135
7.1.6	Definiowanie nazw	135
7.2	Najpopularniejsze funkcje z arkusza kalkulacyjnego	135
7.2.1	Funkcja SUMA	135
7.2.2	Funkcja JEŻELI	135
7.2.3	Ćwiczenie z funkcjami SUMA i JEŻELI	136
7.2.4	Proste funkcje statystyczne	136
7.2.5	Funkcje zliczające	136
7.2.6	Funkcja ZAOKR	136
7.2.7	Funkcje wyszukiujące	137
7.2.8	Ćwiczenie z funkcją WYSZUKAJ.PIONOWO	137
7.2.9	Proste funkcje tekstowe	137
7.3	Tworzenie arkusza - podsumowanie	137
7.3.1	Dodatkowe uwagi o pisaniu formuł	137
7.3.2	Błędy obliczeń sygnalizowane przez arkusz	138

7.3.3	Pętle adresowe	138
7.3.4	Etapy tworzenia arkusza kalkulacyjnego	139
7.3.5	Ćwiczenie z dużym przykładem	139
7.4	Operacje bazodanowe	140
7.4.1	Sortowanie	140
7.4.2	Filtrowanie	140
7.4.3	Ćwiczenie z sortowaniem i filtrowaniem	140
8	Dalsze operacje na arkuszach	141
8.1	Wykresy w arkuszu kalkulacyjnym	141
8.1.1	Tworzenie wykresów za pomocą kreatora	141
8.1.2	Miejsce na wykres	142
8.1.3	Typ wykresu	144
8.1.4	Format wykresu	145
8.1.5	Opisy wykresu	147
8.1.6	Wykres gotowy!	148
8.1.7	Tworzenie wykresów — co dalej?	148
8.1.8	Ćwiczenie z tworzenia wykresu	150
8.2	Drukowanie arkusza kalkulacyjnego	151
8.2.1	Drukowanie arkusza	151
8.2.2	Nagłówki i stopki	152
8.2.3	Zamrażanie nagłówków kolumn i wierszy	153
8.2.4	Dopasowywanie rozmiarów wydruku	154
8.2.5	Dodatkowe możliwości związane z drukowaniem	157
8.3	Kilka nieco bardziej zaawansowanych tematów	159
8.3.1	Funkcje bazodanowe	159
8.3.2	Ćwiczenia z funkcji bazodanowych	160
8.3.3	Analizy co-jeśli	160
8.3.4	Uwagi o pisaniu i nagrywaniu makr	160
9	Podstawy redagowania dokumentów elektronicznych	162
9.1	Wprowadzenie	162
9.2	Uruchamiamy i poznajemy edytor tekstu	163

9.3	Wprowadzamy tekst	167
9.4	Wędrujemy po tekście	171
9.5	Wprowadzamy poprawki	171
9.6	Zaznaczamy fragmenty tekstu	172
9.7	Operacje na fragmentach tekstu	174
9.8	Proste operacje formatowania	176
9.9	Zapisujemy swoją pracę	180
9.10	Linijka	182
9.11	Przygotowujemy dokument do druku	186
9.12	Formatowanie akapitu, strony i dokumentu	187
9.13	Tabele	196
9.14	Grafika	198
9.15	Wyszukiwanie wzorca w tekście	199
9.15.1	Wyszukiwanie	200
9.15.2	Zastępowanie wzorca w tekście	204
10	Strukturalizacja dokumentu i złożone operacje na tekście	207
10.1	Strukturalizacja dokumentu	207
10.2	Style i ich stosowanie	208
10.3	Modyfikacja stylu znaków	214
10.4	Definiowanie nowego stylu znaków	216
10.5	Style akapitów: tworzenie i modyfikacja	218
10.6	Style z automatyczną numeracją	220
10.7	Szybka zmiana stylu akapitu	224
10.8	Szybkie definiowanie nowego stylu akapitu	224
10.9	Konspekt i automatyczna numeracja rozdziałów	224
10.10	Spis treści i inne spisy	228
10.11	Wyszukiwanie różnych atrybutów tekstu	230
10.12	Wyrażenia regularne	231
10.13	Indeks	233
10.14	Odsyłacze	238
10.15	Bibliografia	240

11 System składu dokumentów \LaTeX	244
11.1 Wprowadzenie do systemu \LaTeX	244
11.2 Schemat pracy z systemem \LaTeX	245
11.3 Struktura pliku źródłowego	247
11.4 Strona tytułowa	253
11.5 Podział dokumentu na rozdziały	254
11.6 Wyróżnianie, zmiana czcionki	255
11.7 Wyliczenia	256
11.8 Tabele	259
11.9 Spis treści i odsyłacze	261
11.10 Bibliografia	264
11.11 Formuły matematyczne	265
12 Hipertekst i HTML	268
12.1 Wprowadzenie	268
12.1.1 Cele zajęć	269
12.1.2 Materiały dydaktyczne	270
12.1.3 Przeglądarka	270
12.2 Hipertekst	271
12.2.1 Specyfika hipertekstu	273
12.2.2 Porównanie wybranych charakterystyk formatów HTML i PDF	275
12.2.3 Przeglądarki dokumentów hipertekstowych	275
12.3 Pierwsze strony WWW	276
12.3.1 Znaczenie znaczników	277
12.3.2 Najprostszy kompletny dokument w formacie HTML	280
12.3.3 Narzędzia tworzenia dokumentów w formacie HTML	285
12.3.4 Jak stworzyć stronę w formacie HTML — i sprawdzić, co wyszło?	289
12.3.5 Uwaga na literówki!	292
13 Struktura strony WWW	295
13.1 Struktura logiczna	295
13.1.1 Szkielet tekstu	298

13.1.2	Listy	303
13.1.3	Tabele <table>	306
13.1.4	Podsumowanie	308
13.2	Znaczniiki nadające tekstowi dodatkowy sens	309
13.2.1	Linki <a>	309
13.2.2	Wyróżnienia	311
13.2.3	Przydatne drobiazgi	313
13.2.4	Podsumowanie	315
13.3	Publikacja	316
13.3.1	Poprawność językowa stron HTML	316
13.3.2	Publikacja na serwerze WWW	319
13.3.3	Niezależność od sprzętu i przeglądarki	321
13.3.4	Konserwacja stron WWW	322
14	Atrakcyjne strony WWW	323
14.1	Multimedia	323
14.1.1	Grafika	323
14.1.2	Formaty grafiki WWW	325
14.1.3	Nagrania dźwiękowe i filmy	326
14.1.4	Kilka słów o dobrych obyczajach	326
14.1.5	Zanurzanie multimediów w naszej stronie WWW (<i>nie-obowiązkowo</i>)	327
14.1.6	Wielkość <i>ma</i> znaczenie	333
14.2	Formatowanie stron WWW	335
14.2.1	Potrzeba oddzielenia formy od treści	335
14.2.2	Arkusze CSS	337
14.2.3	Struktura arkusza CSS	341
14.2.4	Klasy (<i>nieobowiązkowo</i>)	342
14.2.5	Wybrane parametry formatowania CSS	343
14.2.6	Kilka przykładów praktycznych zastosowań CSS	352
14.2.7	O potrzebie umiaru	354
14.2.8	Media	355
14.2.9	Gotowe arkusze CSS	356

14.2.10	Poprawność arkusza CSS	356
14.3	Projektowanie stron WWW	357
14.3.1	Specyfika stron domowych	358
14.4	Czego <i>nie należy</i> publikować na WWW?	362
14.5	Dynamiczne strony WWW (<i>nieobowiązkowo</i>)	363
14.5.1	CSS	364
14.5.2	JavaScript	364
14.5.3	PHP	365
14.6	Kilka tricków przydatnych w życiu (<i>nieobowiązkowo</i>)	365
14.6.1	Data aktualizacji strony	365
14.6.2	Niezauważalne wklejenie zawartości zewnętrznego pliku	366
14.6.3	Formatowanie dokumentu do wydruku	366
14.6.4	Szablony stron	368
14.6.5	Konwertery HTML	370
14.6.6	Tworzenie dokumentów PDF	371
14.7	Przyszłość HTML	372
15	Prezentacje	373
15.1	Wprowadzenie	373
15.1.1	Przygotowywanie prezentacji	373
15.1.2	OpenOffice Impress	374
15.2	Kreator prezentacji	375
15.2.1	Kreator prezentacji — krok 1	376
15.2.2	Kreator prezentacji — krok 2	376
15.2.3	Kreator prezentacji — krok 3	377
15.2.4	Nową prezentację należy od razu zapisać	378
15.3	Podstawowe możliwości OpenOffice Impress	378
15.3.1	Dodawanie obiektów do slajdu	379
15.3.2	Gotowe układy slajdu	380
15.3.3	Dostosowywanie paneli do swoich potrzeb	380
15.3.4	Pokaz slajdów, zapisywanie i eksportowanie	381
15.4	Dalsze możliwości OpenOffice Impress	382
15.4.1	Panel zadania	382

15.4.2	Ustawienia pokazu slajdów	383
15.4.3	Widoki	385
15.4.4	Wstawianie wykresów	389
15.4.5	Edycja wzorca stron	392
16	Wprowadzenie do systemu USOS. Zasoby BUW dostępne drogą elektroniczną	396
16.1	Wprowadzenie	396
16.2	Uniwersytecki System Obsługi Studiów	396
16.2.1	Architektura systemu USOS	396
16.2.2	Dostęp do aplikacji USOSweb	397
16.2.3	Główne okno aplikacji USOSweb	397
16.2.4	Logowanie	398
16.2.5	Moduły systemu USOSweb	398
16.3	Zasoby BUW dostępne drogą elektroniczną	414
16.3.1	Strona główna BUW	414
16.3.2	Katalog komputerowy	415
16.3.3	Wydawnictwa elektroniczne	416
17	Egzamin	418
17.1	Zasady przeprowadzania egzaminu	418
17.2	Cele egzaminu: lista umiejętności	419
17.3	Przykładowe zestawy egzaminów	419
17.3.1	Baza każdego egzaminu	419
17.3.2	Przykładowy egzamin	420
17.3.3	Inne pomysły na zadania egzaminacyjne	429

Rozdział 1

Wprowadzenie

Przedmiot *Przysposobienie Informatyczne* ma wyznaczać minimum umiejętności posługiwania się komputerem oraz poszerzać i ugruntowywać umiejętności i wiedzę studentów zdobytą w szkole na przedmiocie Technologie Informatyczne.

Uczestnictwo w kursie podstawowym umożliwi studentom poznanie najnowszycy technologii komputerowych, ze szczególnym uwzględnieniem technologii przydatnych w procesie kształcenia na Uniwersytecie Warszawskim.

Poprzez kursy rozszerzające program Przysposobienie Informatyczne umożliwi studentom poznanie zaawansowanych technik informatycznych związanych bezpośrednio z poszczególnymi dziedzinami kształcenia i badań.

1.1 Informacje organizacyjne

Przedmiot Przysposobienie Informatyczne jest prowadzony przez pracowników jednostek dydaktycznych Uniwersytetu Warszawskiego. Kontakt z instruktorami jest możliwy za pośrednictwem poczty elektronicznej lub bezpośrednio w terminach konsultacji instruktorów.

- Instruktor prowadzący
 - Imię i nazwisko
 - Adres poczty elektronicznej
 - Strona domowa instruktora
- Konsultacje
 - Numer pokoju, budynek
 - Godziny
- Strona główna przedmiotu

<http://pi.mimuw.edu.pl>

Zajęcia z przedmiotu Przystosowanie Informatyczne są prowadzone w pracowniach komputerowych przygotowanych na potrzeby przedmiotu. Pracownia wzorcowa została przygotowana na Wydziale Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego. Podobne pracownie komputerowe powstają w obrębie innych jednostek dydaktycznych uczestniczących w programie.

Na potrzeby przedmiotu Przystosowanie Informatyczne uruchomiona została również witryna internetowa, na której na bieżąco publikowane są informacje o przebiegu prac zmierzających do uruchomienia przedmiotu. Witryna przedmiotu będzie również miejscem, w którym publikowane są materiały dydaktyczne opracowane w ramach przedmiotu.

1.2 Cele zajęć

Zajęcia modułu podstawowego są realizowane w następujących pakietach tematycznych.

- Zasady działania komputera i sieci
- Podstawowe narzędzia biurowe
- Wykorzystanie zasobów Internetu
- Arkusz kalkulacyjny
- Redagowanie dokumentów
- Tworzenie stron WWW
- Przygotowywanie prezentacji multimedialnych
- Zasoby sieciowe UW

- Zasady działania komputera i sieci
 - pojęcie systemu operacyjnego, aplikacji
 - organizacja zasobów w komputerze, pliki, katalogi, skróty i podstawowe operacje na nich
 - pojęcie przeglądarki sieciowej, podstawowe zagadnienia związane z zasobami w Internecie
 - użytkownicy (nazwy, hasła), logowanie w systemie operacyjnym Windows a logowanie w systemie operacyjnym Linux
 - porównanie popularnych systemów operacyjnych (Windows, Linux)
- Podstawowe narzędzia biurowe
 - kompresja plików
 - posługiwanie się podstawowymi aplikacjami (w tym aplikacjami biurowymi)

- podstawy edycji i formatowania tekstu
- korzystanie z oprogramowania otwartego, informacje o prawie własności intelektualnej
- Wykorzystanie zasobów Internetu
 - korzystanie z przeglądarek sieciowych i narzędzi do wyszukiwania dokumentów w Internecie
 - pobieranie z sieci dokumentów i utrwalanie w różnych formatach konwersja formatów (w tym konwersja DOC do PDF oraz PS)
 - posługiwanie się programami pocztowymi
 - przysyłanie pocztą elektroniczną załączników w różnych formatach
 - podpisywanie i szyfrowanie poczty elektronicznej
- Arkusz kalkulacyjny
 - posługiwanie się formułami do tworzenia skomplikowanych obliczeń
 - wykorzystanie makr na potrzeby automatyzacji pracy z arkuszem kalkulacyjnym
 - tabele przestawne jako zaawansowany przykład analizy i drążenia danych
- Redagowanie dokumentów
 - dokumenty o objętości typowej pracy studenckiej, dyplomowej lub naukowej
 - narzędzia wspomagające przetwarzanie tekstów
- Tworzenie stron WWW
 - podstawy HTML
- Przygotowywanie prezentacji multimedialnych
 - podstawy prezentacji multimedialnych
- Zasoby sieciowe UW
 - Uniwersytecki System Obsługi Studiów (USOS)
 - System informatyczny Biblioteki Uniwersytetu Warszawskiego

1.3 Harmonogram zajęć

Zajęcia są prowadzone zgodnie z następującym harmonogramem.

Numer zajęć	Treść zajęć
1	Podstawy posługiwania się komputerem
2	Podstawowe aplikacje biurowe
3,4	Korzystanie z przeglądarek sieciowych
5,6,7	Arkusze kalkulacyjny
8,9,10	Redagowanie dokumentów elektronicznych
11,12,13	Tworzenie stron WWW
14	Przygotowywanie prezentacji
15	Zasoby sieciowe UW

Szczególny nacisk położony jest na następujące umiejętności.

1. Korzystanie z arkuszy kalkulacyjnych
2. Redagowanie dokumentów elektronicznych
3. Tworzenie własnych stron www

1.4 Materiały dydaktyczne

Do zajęć przewidziane zostały następujące materiały dydaktyczne.

- Podręcznik studenta
- Slajdy do prowadzenia zajęć
- Przykładowe zadania egzaminacyjne
- Dostępne do pobrania ze strony przedmiotu

<http://pi.mimuw.edu.pl>

Podręcznik jest dokumentem w formacie elektronicznym PDF zawierającym szczegółowy opis materiału omawianego na zajęciach. Materiał zawarty w podręczniku zawiera wszystkie informacje wymagane w ramach Przesposobienia Informatycznego, a w wielu miejscach wykracza poza ich zakres.

Prezentacje są dokumentem w formacie elektronicznym PDF zawierającym treść prezentacji wyświetlanych przez instruktorów w trakcie prowadzenia zajęć.

Egzaminy są dokumentem w formacie elektronicznym PDF zawierającym opis trybu przeprowadzania egzaminów oraz przykłady zadań egzaminacyjnych.

Wszystkie dokumenty elektroniczne są dostępne do pobrania ze strony przedmiotu.

<http://pi.mimuw.edu.pl>

1.5 Zasady zaliczania

- Obecność
- Zadania cząstkowe
- Egzamin końcowy
- Certyfikat

Uczestnictwo w zajęciach jest obowiązkowe. W trakcie zajęć studenci wykonują ćwiczenia pod opieką instruktorów. Niektóre z ćwiczeń kończą się zadaniem cząstkowym zdefiniowanym przez instruktora. Dostarczanie zadań cząstkowych jest warunkiem dopuszczenia do egzaminu. Egzamin z przedmiotu jest realizowany jako zadanie praktyczne wykonywane w warunkach kontrolowanych w laboratoriach przeznaczonych do prowadzenia przedmiotu. Egzamin jest obowiązkowy.

Po uruchomieniu przedmiotu w skali uniwersyteckiej pozytywne zaliczenie przedmiotu będzie skutkowało wydaniem certyfikatu potwierdzającego poziom wiedzy informatycznej.

Przykład 1.5.1.



Certyfikat potwierdzający poziom wiedzy informatycznej

Rozdział 2

Zasady działania komputera

2.1 Wprowadzenie

Do podstawowych funkcji systemu operacyjnego należy autoryzowanie użytkowników, organizacja zasobów w komputerze w oparciu o pliki, katalogi oraz udostępnianie podstawowych operacji na nich.

Ściśle związane z pracą systemu operacyjnego w sieci komputerowej jest pojęcie przeglądarki internetowej, która staje się nieodzownym narzędziem współczesnej pracy biurowej.

W chwili obecnej rynek komputerów osobistych jest zdominowany przez trzy rodziny systemów operacyjnych: Windows, Linux i Mac. Zajęcia "Przygotowania Informatycznego" są prowadzone przede wszystkim w oparciu o system Microsoft Windows.

Niniejszy moduł pozwoli *aktywnie studiującej* osobie osiąść umiejętności pozwalające na:

- rozumienie pojęcia i roli systemu operacyjnego
- posługiwanie się
 - pulpitem
 - paskiem zadań
 - wierszem poleceń
 - eksploratorem systemu
- zmianę ustawień w panelu sterującym systemem operacyjnym
- obsługę drukarek i kolejek wydruku

2.1.1 Popularne systemy operacyjne

W chwili obecnej użytkownicy komputerów osobistych mogą wykorzystywać do pracy szeroką gamę platform sprzętowych i systemów operacyjnych.

- Microsoft
 - Dos, Windows 3.11
 - Windows 95, 98
 - Windows NT
 - Windows 2000
 - Windows XP, 2003
- Apple
 - Macintosh
 - System 6, 7
 - MacOS 8, 9
 - MacOS X 10
- Linux
 - RedHat
 - SuSE
 - Debian

Na początku lat osiemdziesiątych XX wieku świat systemów operacyjnych wyglądał dość ubogo. Urządzenia domowe były zdominowane przez mikrokomputery klasy Spectrum, Atari lub Commodore, które służyły przede wszystkim celom rozrywkowym. Wśród zastosowań komercyjnych największą część rynku zajmował trudny w obsłudze system IBM PC.

Początek 1984 roku przyniósł rewolucję w postaci komputera o nazwie Macintosh produkowanego przez firmę Apple. Najważniejszym założeniem nowego komputera było umożliwienie korzystania z niego szerokiemu gronu użytkowników. Była to strategia, która pociągnęła za sobą wiele innowacji, zarówno sprzętowych jak też programowych.

Jednym z wyróżników Macintosha była obudowa zintegrowana z monitorem. Do ważącego około 8 kilogramów komputera dodano uchwyt do przenoszenia, dzięki czemu był nie tylko lżejszy, ale i poręczniejszy niż wiele z ówczesnych komputerów przenośnych.

Komputer po włączeniu był natychmiast gotowy do użycia: użytkownik Macintosha nigdy nie zaglądał do środka, nie przestawiał przełączników, nie wkładał karty rozszerzeń, nie edytował parametrów konfiguracyjnych.

W konstrukcji monitora, zamiast obowiązującej wówczas bursztynowej lub zielonej poświaty użyto czerni i bieli, dzięki czemu napisy na monitorze wyglądały tak samo, jak na kartce papieru.

Dołączono też do klawiatury nowe urządzenie wejściowe w postaci myszy komputerowej. Mysz była podstawą obsługi graficznego systemu operacyjnego, opartego o okienka i menu. Była to najważniejsza z innowacji firmy Apple. W porównaniu do innych systemów operacyjnych elementy graficzne

nie były jedynie dodatkiem, lecz integralną częścią systemu. Ekran monitora wyglądał jak pulpit biurka, katalogi jak teczki, dokumenty jak kartki papieru, kosz na śmieci służył do usuwania zbędnych danych. Dzięki takiemu podejściu komputerem mogły się posługiwać osoby nie mającej wiedzy specjalistycznej, gdyż zamiast wprowadzania komend i długich ciągów parametrów mogły po prostu wskazać myszą wybrany element na pulpicie.

Mimo błyskotliwego startu firmy Apple, palmę pierwszeństwa przejął wkrótce Microsoft z system operacyjnym Windows. W pierwszych wersjach system Windows działał jako nakładka na system operacyjny DOS (oficjalnie wyłącznie na system operacyjny MS-DOS), następnie zaczął działać jako samodzielny system operacyjny. Pierwsza wydanie systemu nastąpiło w 1985 jako odpowiedź na graficzny system operacyjny firmy Apple Macintosh. Obecnie system Windows zdominował rynek komputerów osobistych, jest używany w ponad 90% komputerów nie będących serwerami.

Najmłodszym w rodzinie systemów operacyjnych popularnych w zastosowaniach domowych jest system Linux, który jako otwarty system operacyjny nieprzerwanie zyskuje popularność (w różnych dystrybucjach).

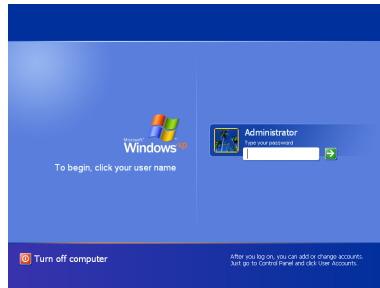
Linux zaczął powstawać w 1991 roku, kiedy to fiński programista Linus Torvalds stworzył jądro nowego systemu operacyjnego przeznaczonego do pracy z procesorami rodziny 80386 firmy Intel. Torvalds opublikował informacje o systemie na internetowej liście dyskusyjnej, co spotkało się z dużym zainteresowaniem i przyspieszyło rozwój systemu. Im bardziej system stawał się popularny, tym więcej ludzi wspierało jego rozwój. Proces ten trwa do dziś, a liczbę użytkowników różnych dystrybucji Linuksa szacuje się obecnie na wiele milionów na całym świecie.

Linux jest systemem wolnodostępnym rozpowszechnianym na licencji GNU General Public License, dzięki czemu kod źródłowy jest dostępny dla każdego użytkownika systemu i każdy użytkownik może dowolnie go modyfikować. System Linux funkcjonuje obecnie w wielu odmianach i wydaniach, choć sama nazwa Linux jest obecnie zastrzeżonym znakiem towarowym należącym do Linusa Torvaldsa.

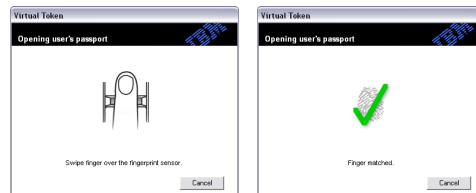
2.1.2 Autoryzacja użytkowników

Jedną z głównych ról systemu operacyjnego jest udostępnianie, ale również ochrona zasobów komputera. Dlatego pierwszym krokiem w pracy z komputerem jest autoryzacja użytkownika. We współczesnych systemach operacyjnych funkcjonują różne sposoby autoryzacji użytkowników.

- Login, hasło



- Autoryzacja biometryczna



Klasyczny sposób autoryzacji użytkowników bazuje na identyfikatorze i hasle. Użytkownik wprowadza unikalny identyfikator złożony z ciągu znaków oraz przypisane do niego hasło (również w postaci tekstowej).

W wielu nowoczesnych komputerach wprowadzono możliwość autoryzacji użytkowników w oparciu o dane biometryczne. Najbardziej powszechnym sposobem jest autoryzacja użytkowników w oparciu o odcisk palca. W takim przypadku użytkownik przeciąga palec nad czytnikiem linii papilarnych. Nie jest wymagane wprowadzanie żadnych dodatkowych informacji w postaci tekstowej.

2.1.3 Podstawowe narzędzia

System operacyjny komputera stanowi warstwę oprogramowania pośredniczącą pomiędzy użytkownikami i aplikacjami narzędziowymi, którymi się posługują a platformą sprzętową zawierającą gamę dedykowanych urządzeń. Współczesne systemy operacyjny są obsługiwane w pierwszym rzędzie w oparciu o graficzny interfejs użytkownika.

- Interfejs graficzny
 - Urządzenie wskazujące
 - Pulpit użytkownika
 - Pasek zadań
- Interfejs tekstowy
 - Klawiatura
 - Uruchom
 - Wiersz poleceń

W graficznym interfejsie użytkownika do sterowania komputerem wykorzystywane jest przede wszystkim urządzenie wskazujące. W przypadku komputerów osobistych najczęściej spotykanym urządzeniem wskazującym jest mysz komputerowa. Komputery wykorzystywane w zastosowaniach graficznych są często wyposażone w trackball lub pióro elektroniczne. W przypadku laptopów standardem jest obecnie trackpoint lub touchpad (lub oba urządzenia).



Trackball



Touchpad

Urządzenie sterujące pozwala przesuwać po ekranie wskaźnik (lub kursor), za pomocą którego można zaznaczać i przemieszczać na ekranie elementy graficznego interfejsu użytkownika, w szczególności otwierać okna aplikacji, zmieniać ich położenie i wielkość oraz je zamykać.

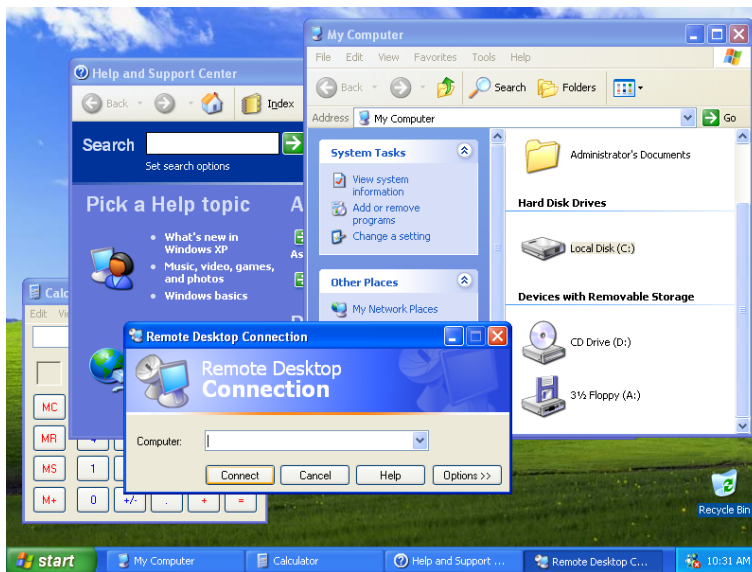
Za pomocą najpopularniejszego urządzenia wskazującego, jakim jest mysz komputerowa, można wykonać pięć podstawowych czynności.

- **Kliknięcie**
Należy nacisnąć i puścić podstawowy (domyślnie lewy) przycisk myszy. Zazwyczaj wykorzystywane do zaznaczanie aktywnych ikon, okien oraz opcji.
- **Kliknięcie drugim przyciskiem**
Należy nacisnąć i puścić dodatkowy (domyślnie prawy) przycisk myszy. Zazwyczaj wykorzystywane do wyświetlania menu podręcznego w kontekście klikniętego obiektu.
- **Podwójne kliknięcie**
Należy szybko dwukrotnie kliknąć podstawowy (domyślnie lewy) przycisk myszy. Zazwyczaj wykorzystywane do uruchamiania aplikacji i otwierania folderów.
- **Przeciągnięcie**
Należy przycisnąć i przytrzymać podstawowy (domyślnie lewy) przycisk myszy jednocześnie przesuwać mysz. Zwolnienie przycisku myszy kończy akcję. Zazwyczaj wykorzystywane do przesuwania lub zmiany rozmiaru okien, zaznaczania tekstu.

- Przeciągnięcie drugim przyciskiem
Należy przycisnąć i przytrzymać dodatkowy (domyślnie prawy) przycisk myszy jednocześnie przesuwając mysz. Zwolnienie przycisku myszy kończy akcję. Zazwyczaj skutkuje wyświetleniem menu kontekstowego pozwalającego wybrać rodzaj akcji właściwej przeciąganemu obiektowi.

Pulpit użytkownika

Elementem podstawowym interfejsu graficznego jest *pulpit użytkownika*. Część centralną pulpitu zajmują okna aplikacji otwartych przez użytkownika.



Poza przestrzenią pozwalającą na przechowywanie katalogów i plików, pulpit użytkownika składają się: pasek zadań, obszar powiadomień oraz obszar szybkiego uruchamiania.

Pasek zadań

Pasek zadań zazwyczaj umieszczany jest w dolnej części pulpitu. Na pasku zadań pojawiają się przyciski odpowiadające oknom otwartych aplikacji. Klikając przycisk myszy na pasku zadań można wygodnie przejść do wybranej aplikacji. Po kliknięciu drugim przyciskiem myszy na przycisku paska zadań odpowiadającym oknu wybranej aplikacji wyświetlane jest menu podręczne pozwalające między innymi na zamknięcie wybranej aplikacji.

- Pasek zadań



- Obszar powiadomień



- Obszar szybkiego uruchamiania

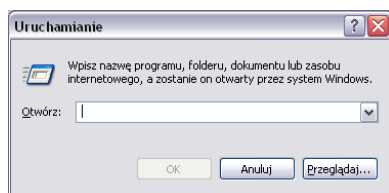


Obszar powiadomień to skrajnie prawa część paska zadań, w której wyświetlane są informacje o stanie działania aplikacji komputera. Większość informacji jest wyświetlana w postaci ikon lub piktogramów, na które można klikać. Kliknięcie, kliknięcie drugim przyciskiem myszy lub kliknięcie podwójne zazwyczaj skutkują inną czynnością, zależną od typu klikanego obiektu. Jedną ze standardowych informacji wyświetlanych w obszarze powiadomień jest zegar systemowy. Podwójne kliknięcie na zegarze pozwala na zmianę aktualnej daty i godziny (o ile użytkownik posiada odpowiednie uprawnienia).

Obszar szybkiego uruchamiania to lewa część paska zadań (obok przycisku Start), w której wyświetlane są ikony umożliwiające uruchamianie wybranych aplikacji. Ze względu na ograniczoną ilość miejsca w obszarze szybkiego uruchamiania, zalecane jest umieszczanie tam jedynie najczęściej używanych aplikacji. Zarówno obszar szybkiego uruchamiania, jak też obszar powiadomień, zmniejszają ilość miejsca dostępnego na pasku zadań na wyświetlanie przycisków odpowiadających otwieranym oknom aplikacji.

Uruchom

Okno *Uruchom* pozwala w szybki sposób uruchamiać programy w oparciu o ich nazwę. Aby wyświetlić okno *Uruchom* należy wybrać *Menu Start* ► *Uruchom*.

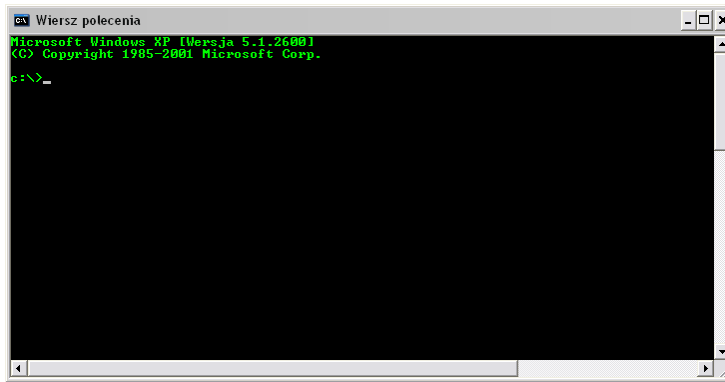


Aby przejść do szybkiego uruchamiania programów wybierz kolejno:

Menu Start ► Uruchom

Wiersz polecenia

Bardziej zaawansowani użytkownicy mogą potrzebować sposobu na wydawanie większej ilości poleceń w trybie tekstowym. W takim przypadku odpowiednim narzędziem jest aplikacja *Wiersz polecenia*, dostępna jako jedno z poleceń w menu Start. Innym sposobem uruchomienia *Wiersza poleceń* jest wpisanie komendy *cmd* w oknie *Uruchom*.



Aby wyświetlić okno wiersza polecenia wybierz kolejno:

Menu Start ► Programy ► Akcesoria ► Wiersz polecenia

Wiersz polecenia pozwala na przesyłanie poleceń do systemu operacyjnego za pomocą klawiatury. Typowe polecenia wydawane z klawiatury są następujące.

- `cls`
Czyści okno wiersza poleceń.
- `dir`
Wyświetla strukturę katalogową.
- `cd`
Zmienia katalog bieżący.
- `ipconfig`
Pokazuje bieżącą konfigurację sieciową komputera.
- `ping`
Wysyła pakiet kontrolny do komputera zdalnego.
- `tracert`
Wyświetla ścieżkę routingu do komputera zdalnego.
- `subst`
Mapuje katalog jako dysk lokalny.

2.2 Zasoby komputera

Współczesne komputery zarządzają licznym zestawem zasobów komputerowych. Historycznie zasobem fundamentalnym w koncepcji każdego współczesnego systemu operacyjnego są lokalne (to jest przechowywane na danym komputerze) pliki i organizujące je katalogi. Katalogi i pliki są zasobami

logicznymi, umieszczonymi na fizycznych urządzeniach różnego typu. Urządzeniem podstawowym przechowującym katalogi i pliki jest dysk twardy komputera.

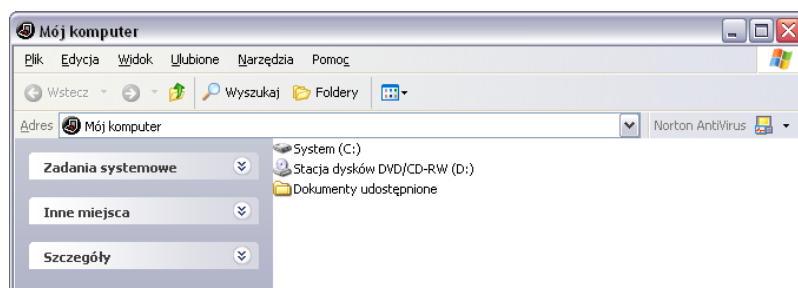
Kolejnym ważnym zasobem komputera są drukarki. W przypadku komputerów domowych drukarka jest zazwyczaj zasobem lokalnym, podłączonym bezpośrednio do komputera. Niekiedy do jednego komputera może być podłączone kilka drukarek.

Większość zasobów lokalnych może posiadać swoje odpowiedniki sieciowe. Tak więc można przechowywać pliki w katalogach zdalnych lub drukować na drukarkach sieciowych.

Użytkownik ma możliwość konfigurowania większości lokalnych fizycznych zasobów komputera oraz zarządzania konfiguracją zasobów logicznych komputera. Poza zarządzaniem zasobami, użytkownik komputera ma również możliwość sprawowania kontroli nad zadaniami realizowanymi przez system operacyjny komputera.

2.2.1 Eksplorator systemu

Podstawową aplikacją użytkownika pozwalającą na przeglądanie zasobów udostępnianych przez system operacyjny jest *Eksplorator systemu*. Najprostszym sposobem uruchomienia narzędzia jest kliknięcie dwukrotnie na ikonie *Mój komputer* znajdującej się na pulpicie lub kliknięcie na ikonie *Mój komputer* prawym przyciskiem myszy i wybranie opcji *Eksploruj* z menu podręcznego. Można też w oknie *Uruchom* wpisać polecenie *explorer*.



Aby wyświetlić okno eksploratora systemu wybierz kolejno:

Menu Start ► Programy ► Akcesoria ► Eksplorator Windows

□ **Ćwiczenie 2.2.1.** Należy znaleźć w strukturze katalogów miejsce, w którym umieszczony jest katalog odpowiadający pulpitowi. □

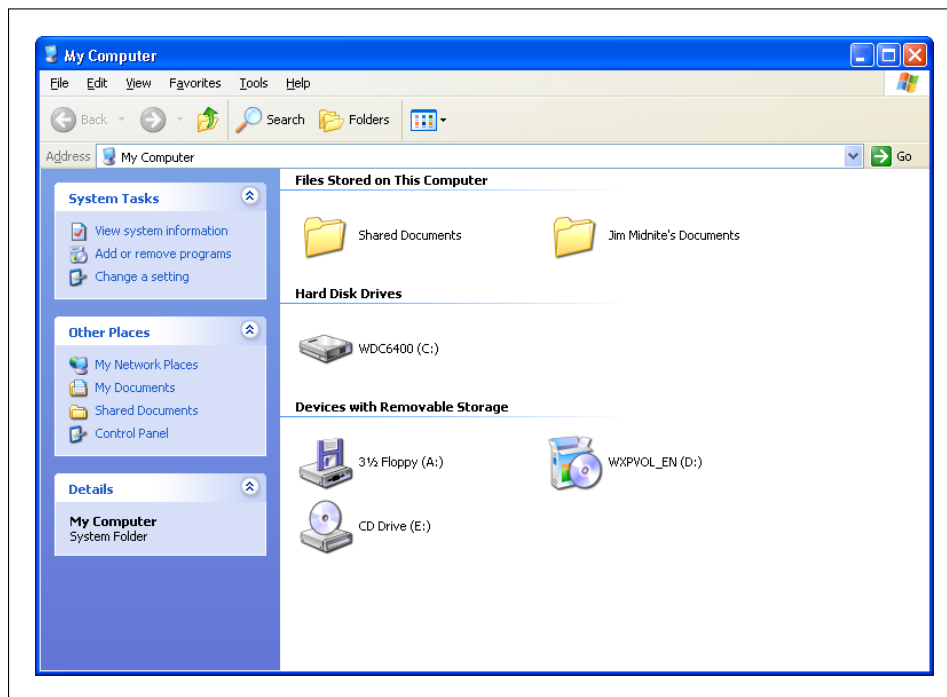
2.2.2 Katalogi i pliki

Podstawowym sposobem przechowywania danych w komputerach jest ich organizacja na nośnikach danych (dyskach, płytach) w pliki i katalogi.

- Dyski twarde
- Dyski wymienne (CD/DVD)
- Urządzenia USB

- Katalogi, pliki
- Ścieżka, nazwa, rozszerzenie
- Właściwości
- Rozróżnianie wielkości liter

Głównym miejscem przechowywania danych w komputerze osobistym jest lokalny system plików. Dysk główny komputera jest zwyczajowo oznaczany literą *C:*.



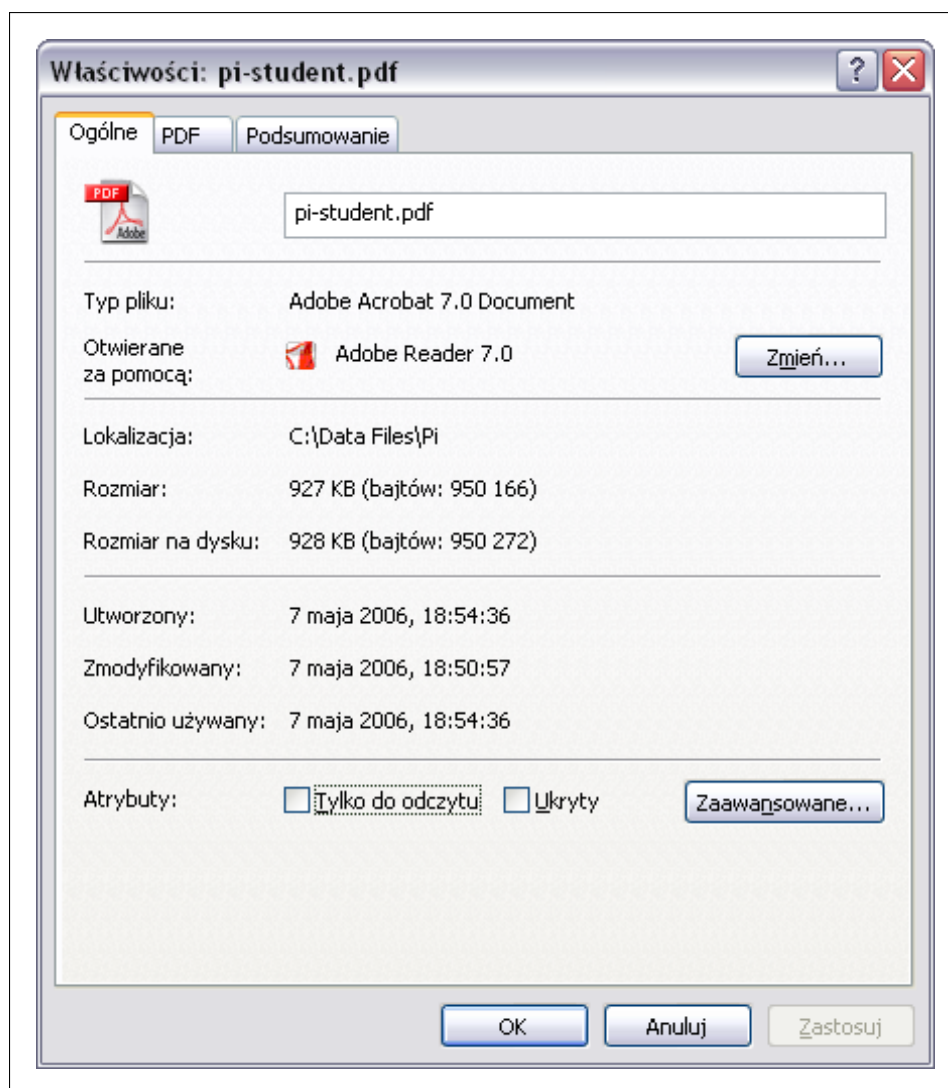
Eksplorator systemu

Litery *A:* oraz *B:* są zwyczajowo rezerwowane dla dyskietek 3.5" oraz 5.25". Dyskietki tego typu są już wycofywanym z użycia sposobem przechowywania i przekazywania informacji ze względu na małą pojemność tych nośników oraz dużą awaryjność stosowanego nośnika magnetycznego. W prezentowanym przykładzie kolejne listery *D:* oraz *E:* reprezentują urządzenia obsługujące płyty kompaktowe (CD, DVD).

W chwili obecnej popularność zdobywa przechowywanie danych na urządzeniach przenośnych podłączanych przez złącze USB. Są to dyski przenośne,

urządzenia typu *pendrive*, palmtopy, telefony komórkowe. Po wykryciu takiego urządzenia system operacyjny wyświetli je w eksploratorze systemu jako kolejny lokalny dysk twardy, nadając zazwyczaj pierwszą wolną literę.

Każdy zasób dyskowy komputera posiada zestaw właściwości opisujących zarówno położenie jak też charakterystykę danego zasobu.



Właściwości zasobu komputera

Do najważniejszych właściwości zasobu dyskowego należą nazwa pliku, rozszerzenie pliku oraz związana z nim domyślnie otwierająca plik aplikacja, położenie pliku na dysku, rozmiar pliku, data utworzenia oraz modyfikacji pliku, oraz atrybuty określające możliwość modyfikacji oraz widoczność pliku.

W systemach operacyjnych MacOs oraz Linux małe i wielkie litery w nazwach plików i katalogów są rozróżniane. W systemie operacyjnym Windows wielkość liter nie ma znaczenia. W większości instalacji systemów operacyj-

nych możliwe jest korzystanie w nazwach plików i katalogów z polskich znaków diakrytycznych, choć warto podkreślić, że jest to praktyka *niezalecana*, gdyż może skutkować problemami z przenoszeniem plików pomiędzy różnymi systemami operacyjnymi, zwłaszcza jeśli występują w różnych wersjach językowych.

2.2.3 Skróty

Plik skrótu wskazujący na inny źródłowy plik lub program jest wygodnym sposobem na uruchamianie danego zasobu źródłowego. Często praktyką jest utrzymywanie pliku źródłowego w strukturze katalogowej w jednym (niezmiennym) miejscu, a jednocześnie umieszczanie na pulpicie skrótu do pliku, nad którym trawają prace. Dwukrotne kliknięcie na skrócie skutkuje otwarciem pliku dokładnie tak, jak dwukrotne kliknięcie na samym pliku. Po zakończeniu prac można swobodnie usunąć skrót z pulpitu pozostawiając nienaruszony plik źródłowy w oryginalnym miejscu.



Aby utworzyć skrót do pliku lub programu wybierz kolejno:

Menu podręczne ► prawy przycisk myszy ► Nowy ► Skrót

Aby utworzyć skrót do pliku lub programu, można w eksploratorze systemu Windows wybrać z menu *Nowy* polecenie *Skrót*. Spowoduje to otwarcie okna dialogowego, w którym należy wskazać lokalizację pliku lub programu źródłowego. Po utworzeniu skrótu do pliku lub programu pojawi się ikona analogiczna do ikony reprezentującej zasób źródłowy. Jediną różnicą jest niewielka strzałka w lewym dolnym rogu ikony.

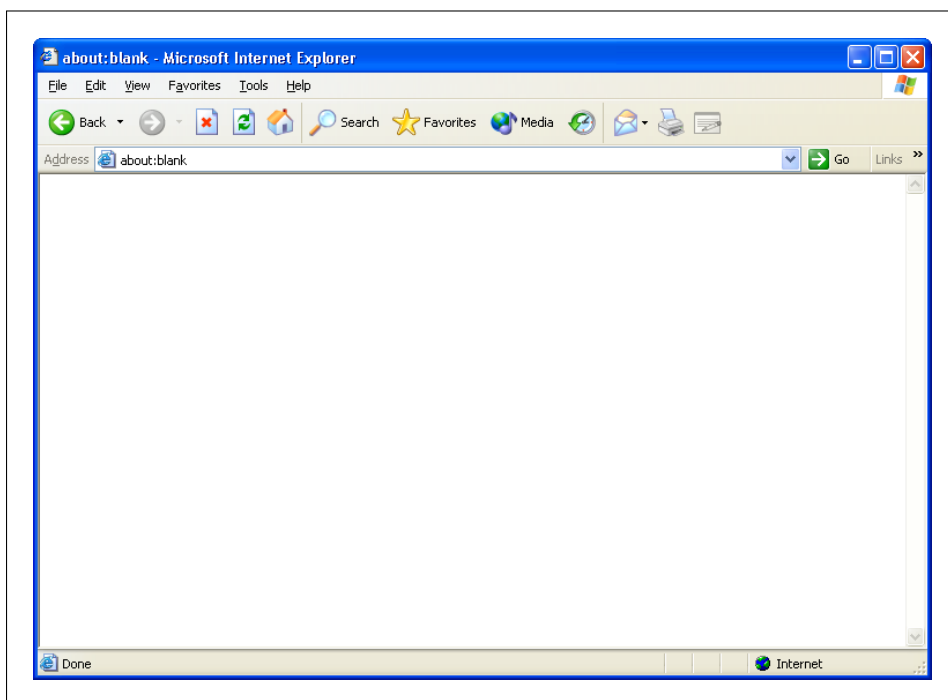


Skróty do pliku lub programu

Warto natomiast pamiętać, że skrót zawiera jedynie położenie pliku źródłowego – nie zawiera treści pliku źródłowego. W szczególności zapisanie pliku skrótu na dysku przenośnym lub wysłanie pliku skrótu pocztą elektroniczną nie spowoduje przesłania treści pliku źródłowego, a jedynie informację o położeniu pliku źródłowego na komputerze nadawcy.

2.2.4 Zasoby sieciowe

Na potrzeby przeglądania zasobów sieciowych przeglądarka internetowa pełni te same funkcje, co eksplorator systemu. Wpisanie adresu sieciowego w przeglądarce powoduje wyświetlenie odpowiednich zasobów sieciowych.



Przeglądarka internetowa

- Typowe zasoby sieci lokalnej
 - Dyski sieciowe
 - Drukarki sieciowe
- Typowe zasoby sieci Internet
 - Strony WWW
 - Zasoby FTP

Dla większości zastosowań komputerów osobistych można zasoby sieciowe podzielić na dwie kategorie: zasoby sieci lokalnej oraz zasoby internetowe.

Sieć lokalna (ang. Local Area Network lub w skrócie LAN) to najmniej rozległa postać sieci komputerowej, która zazwyczaj ogranicza się do jednego biura, jednego budynku, lub kilku pobliskich budynków (często bloków na osiedlu). Technologie stosowane w sieciach lokalnych można podzielić na rozwiązanie oparte na przewodach (kable miedziane, światłowody) lub komunikacji radiowej (sieci bezprzewodowe). Sieci lokalne służą do współdzielenia zasobów pomiędzy komputerami połączonymi w sieć. Typowymi współdzielonymi zasobami są dyski oraz drukarki sieciowe.

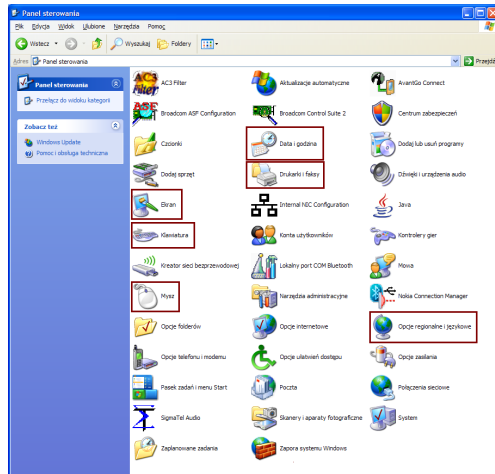
Początki Internetu wiążą się z siecią ARPANET i sięgają końca lat sześćdziesiątych XX wieku. Obecnie internet to sieć komputerowa o zasięgu światowym, która łączy ze sobą sieci lokalne, sieci rozległe i wszystkie komputery do nich podłączone. Standardy internetu regulowane są przez agencję Internet Engineering Task Force (IETF) i publikowane w dokumentach RFC.

Internet jest siecią opartą na protokołach IP (Internet Protocol) oraz TCP/IP (Transmission Control Protocol/Internet Protocol). W sieci Internet funkcjonują publiczne usługi wyższego poziomu oparte na komunikacji i związanej z nią infrastrukturze. Główne z nich to: FTP (ang. File Transfer Protocol), WWW (ang. World-wide Web) oraz usługi poczty elektronicznej. Szczegółowe omówienie usług sieciowych jest zawarte w dalszej części książki.

2.3 Panel sterowania

Panel sterowania to część systemu operacyjnego służąca do ustawiania parametrów sprzętu komputerowego oraz systemu operacyjnego. Za pomocą *Panelu Sterowania* można konfigurować urządzenia peryferyjne współpracujące z komputerem, zarządzać ustawieniami regionalnymi i językowymi systemu operacyjnego, uruchamiać lub wyłączać sprzęt będący częścią wykorzystywanego komputera, zarządzać ustawieniami sieciowymi komputera oraz przeprowadzać zadania administracyjne.

W niektórych konfiguracjach systemu operacyjnego prawa do modyfikowania ustawień w Panelu sterowania posiadają jedynie administratorzy systemu.



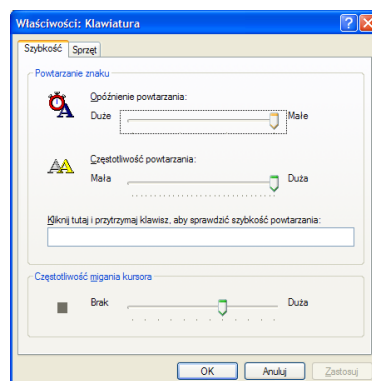
Aby wyświetlić panel sterowania wybierz kolejno:

Menu Start ► Ustawienia ► Panel Sterowania

Aby uruchomić *Panel sterowania* należy z menu *Start* wybrać polecenie *Ustawienia* a następnie polecenie *Panel sterowania*. Można też uruchomić *Panel sterowania* za pomocą komendy *control* wpisanej w oknie *Uruchom*.

2.3.1 Klawiatura

Panel sterowania umożliwia definiowanie parametrów działania podłączonej do komputera klawiatury. Do najważniejszych parametrów działania klawiatury zaliczają się *Opóźnienie powtarzania* klawiszy oraz *Częstotliwość powtarzania* klawiszy. *Opóźnienie powtarzania* pozwala określić po jakim czasie od naciśnięcia i przytrzymania klawisza odpowiadający klawiszowi znak zacznie być powtarzany na ekranie. *Częstotliwość powtarzania* określa jak szybko będzie powtarzany znak po naciśnięciu i przytrzymaniu klawisza.

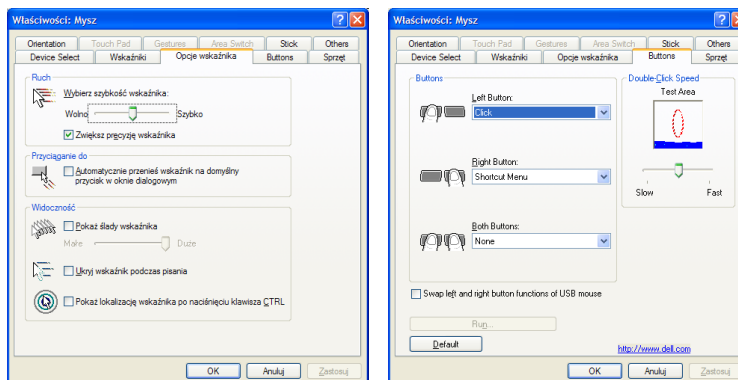


Aby konfigurować parametry klawiatury wybierz kolejno:

2.3.2 Mysz

Panel sterowania umożliwia definiowanie parametrów działania myszy podłączonej do danego komputera. Do najważniejszych parametrów działania myszy zaliczają się *Szybkość wskaźnika* oraz *Szybkość podwójnego kliknięcia*. Szybkość wskaźnika pozwala określić proporcję przesunięcia myszy na ekranie komputera po przesunięciu myszy na pulpicie biurka. Szybkość podwójnego kliknięcia określa przedział czasu w którym należy wykonać dwa kliknięcia myszą aby zostały rozpoznane jako jedno kliknięcie podwójne.

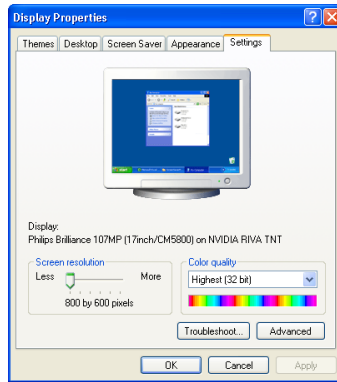
Dla użytkowników początkujących zalecane jest ustawianie podwójnego kliknięcia o niedużej szybkości.



Aby konfigurować parametry myszy wybierz kolejno:

2.3.3 Ekran

Panel sterowania umożliwia definiowanie parametrów działania ekranu podłączonego do danego komputera. Do najważniejszych parametrów działania ekranu zaliczają się *Rozdzielczość ekranu* oraz *Jakość kolorów*. Rozdzielczość ekranu pozwala określić ilość pikseli wyświetlanych na ekranie. Zazwyczaj monitor komputera potrafi obsługiwać pewien zestaw trybów działania charakteryzowanych ilością pikseli w pionie i w poziomie. Jakość kolorów określa wielkość palety kolorów, w jakich wyświetlane są informacje na ekranie. Zazwyczaj większa ilość kolorów podnosi atrakcyjność prezentowanych informacji. Często ilość kolorów jest skorelowana z rozdzielczością ekranu (niekiedy tryby wysokiej rozdzielczości są w stanie wyświetlać jedynie uboższą paletę kolorów).

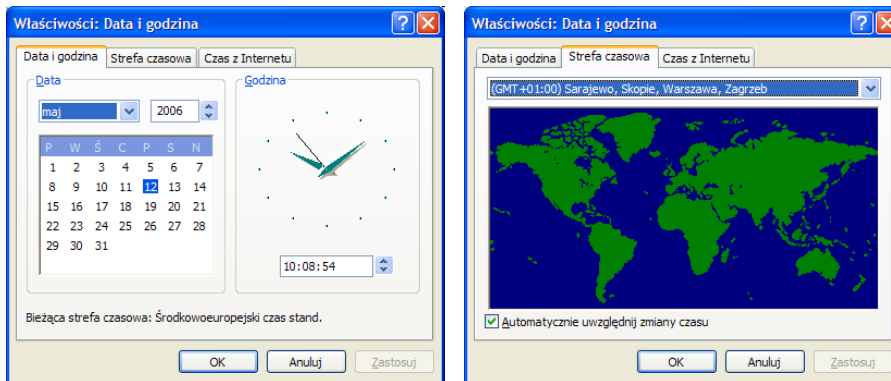


Aby konfigurować parametry ekranu wybierz kolejno:

Menu Start ► Ustawienia ► Panel sterowania ► Ekran

2.3.4 Data i godzina

Panel sterowania umożliwia definiowanie bieżącej *daty* i *godziny*. Dodatkowo możliwe jest zdefiniowanie *strefy czasowej*, w której znajduje się dany komputer. Dzięki wskazaniu właściwej strefy czasowej system operacyjny potrafi automatycznie przeprowadzić uaktualnienie zegara w momencie zmiany czasu pomiędzy letnim a zimowym.



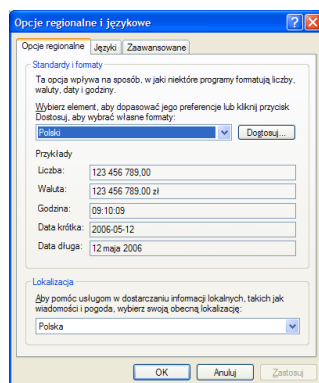
Aby konfigurować parametry daty i godziny wybierz kolejno:

Menu Start ► Ustawienia ► Panel sterowania ► Data i godzina

2.3.5 Opcje regionalne i językowe

Panel sterowania umożliwia definiowanie parametrów wyświetlania dla informacji zależnych od lokalizacji geograficznej danego komputera. Przykładem informacji dla których sposób wyświetlania jest zależny od miejsca

geograficznego jest sposób wyświetlania daty i godziny (np. format 24-godzinny lub 12-godzinny), domyślny symbol waluty (np. zł w przypadku Polski), sposób prezentowania daty (np. pierwszy czerwca 2006 wyświetlane w postaci 2006/01/06 w USA lub 2006-06-01 w Polsce). Możliwe jest wybranie predefiniowanego dla danego kraju zestawu parametrów wyświetlania lub dostosowanie poszczególnych ustawień do własnych potrzeb.



Aby konfigurować parametry ustawień regionalnych i językowych wybierz kolejno:

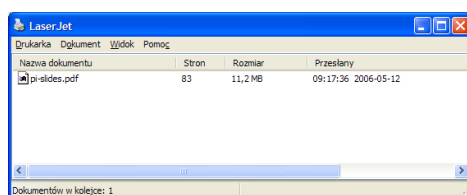
Menu Start ► Ustawienia ►

Panel sterowania ► Opcje regionalne i językowe

2.4 Drukarki i faksy

2.4.1 Kolejka wydruku

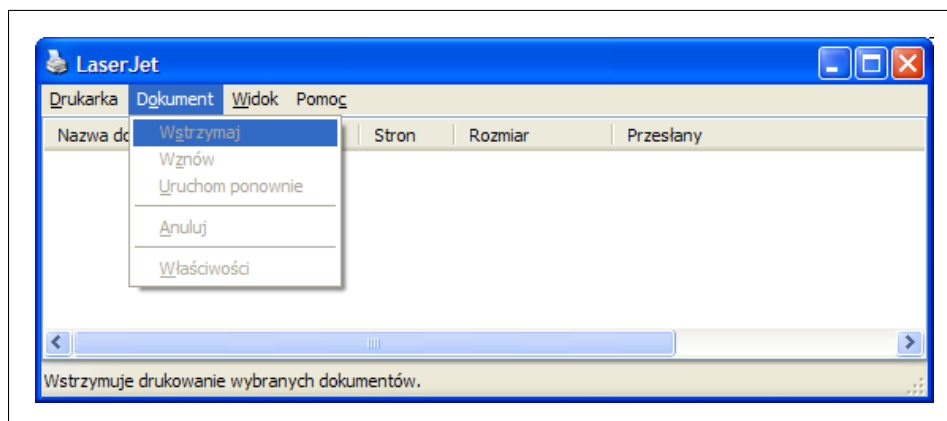
Panel sterowania umożliwia zarządzanie *kolejką wydruków*, która jest podstawowym narzędziem pracy z drukarką. Kolejka wydruków pozwala obejrzeć zakres bieżących zadań drukarki, czyli listę plików oczekujących obecnie na wydruk. W kolejce wydruku dla każdego zadania wyświetlane są podstawowe informacje takie jak nazwa drukowanego dokumentu (pliku), ilość stron wydruku, wielkość pliku (w bajtach) oraz data zgłoszenia do wydruku.



Aby zarządzać kolejką wydruku wybierz kolejno:

Menu Start ► Ustawienia ► Panel sterowania ► Drukarki i faksy

Użytkownik ma w szczególności możliwość wstrzymania i wznowienia wydruku dokumentu lub całkowitego usunięcia zadania wydruku.

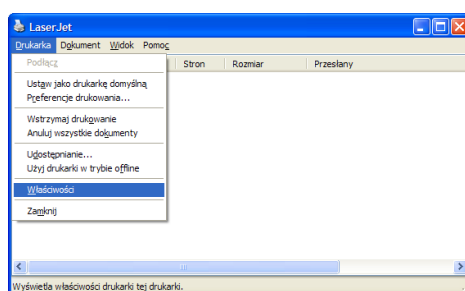


Kolejka wydruku

Realizacja tych zadań jest możliwa po zaznaczeniu dokumentu w kolejce wydruku i wybraniu odpowiedniej opcji z menu *Dokument*.

2.4.2 Parametry wydruku

Do typowych ustawień wydruku, które użytkownik ma prawo modyfikować, należą rozmiar papieru (standardem w Polsce jest papier rozmiaru A4, czyli papier o wymiarach 210 na 297 milimetrów), orientacja papieru (pionowa lub pozioma) oraz układ wydruku (ilość stron drukowanych na jednym arkuszu). Niektóre drukarki posiadają możliwość drukowania dwustronnego. Najczęściej drukowanie dwustronne jest realizowane w oparciu o wewnętrzny podajnik papieru a zwyczajowa nazwa takiej funkcjonalności drukarki to *dupleks*. Wybierając opcję drukowania dwustronnego możemy wskazać, czy wynikowy wydruk powinien być przygotowany dla przekładania wydrukowanych kartek w lewo w trakcie czytania, czy do góry (układ książkowy lub skoroszytowy).



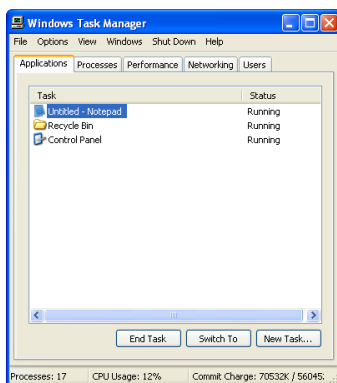
Aby zdefiniować parametry wydruku wskaż właściwą drukarkę i wybierz kolejno:

Właściwości ►Ogólne ►Preferencje drukowania

Aby określić preferencje drukowania należy w kolejce wydruku wybrać poleceni *Drukarka* ►*Właściwości* a następnie w wyświetlonym oknie dialogowym w zakładce *Ogólne* nacisnąć przycisk *Preferencje drukowania*.

2.5 Menedżer zadań

Współczesny system operacyjny obsługuje wiele aplikacji działających równoległe. Większość aplikacji uruchamianych przez użytkowników posiada interfejs graficzny i jest widoczna na pasku zadań. Niekiedy aplikacje po uruchomieniu nie wyświetlają interfejsu graficznego mimo, że działają. Do wygodnego zarządzania zestawem aktualnie działających na komputerze aplikacji służy *Menedżer zadań*. Oprócz wyświetlania listy działających aplikacji oraz związanych z nimi procesów systemu operacyjnego, menedżer zadań umożliwia awaryjne zamknięcie wybranej aplikacji. Taki tryb zamykania aplikacji jest szczególnie przydatny w momencie, gdy aplikacja przestała komunikować się z użytkownikiem (aplikacja "zawiesiła się"). W takim przypadku po uruchomieniu Menedżera zadań wystarczy wskazać na liście aplikacji daną aplikację i nacisnąć przycisk *Zakończ zadanie*.



Aby zarządzać realizowanymi przez system operacyjny zadaniami wybierz kolejno:

Control + Alt + Delete ►Menedżer zadań

Aby uruchomić *Menedżer zadań* należy w dowolnym momencie pracy z komputerem nacisnąć kombinację klawiszy *Control + Alt + Delete*. Spowoduje to wyświetlenie okna w którym należy nacisnąć przycisk *Menedżer zadań*. W oknie obok menedżera zadań możliwe jest wybranie polecenia

Zmień hasło, które wyświetla okno dialogowe pozwalające wskazać użytkownikowi nowe hasło chroniące identyfikator użytkownika. W większości systemów operacyjnych procedura zmiany hasła wymaga podania obowiązującego hasła i dwukrotnego wpisania (w celu uniknięcia pomyłek) hasła nowego. Możliwe jest również zablokowanie dostępu do komputera poprzez wskazanie opcji *Zablokuj komputer*. Wówczas ekran komputera przełączany jest w taki sam tryb jak przy starcie komputera. Wszystkie bieżące aplikacje wyświetlane na pulpicie są ukrywane, zazwyczaj odblokowanie komputera wymaga podania loginu i hasła. Warto podkreślić, że zablokowanie komputera oznacza, że użytkownik jest nadal zalogowany a wszystkie jego aplikacje działają, choć wynik działania tych aplikacji jest ukrywany przez system operacyjny przed podglądem niepowołanych osób.

Rozdział 3

Podstawowe aplikacje

3.1 Wprowadzenie

Z każdym systemem operacyjnym dostarczany jest zestaw aplikacji, które umożliwiają realizację podstawowych zadań edytorsko-biurowych, takich jak tworzenie i edycja plików tekstowych i graficznych czy organizowanie struktur katalogowych, zmiana nazw plików i katalogów, zmiana położenia plików i katalogów w strukturach katalogowych.

Po zakończeniu zajęć oczekujemy, że studenci będą potrafili realizować podstawowe zadania edytorsko-biurowe, przeszukiwać zasoby komputera oraz posługiwać się plikami skompresowanymi.

Niniejszy moduł pozwoli *aktywnie studiującej* osobie osiąść umiejętności pozwalające na:

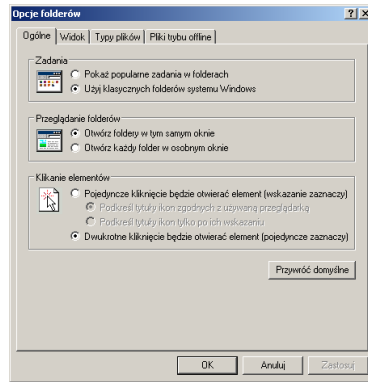
- korzystanie z podstawowy aplikacji dostarczanych wraz z systemem operacyjnym
- przeszukiwanie zasobów komputera
- posługiwanie się plikami skompresowanymi

3.2 Eksplorator systemu

Zaawansowane opcje eksploratora systemu.

3.2.1 Opcje folderów

Użytkownik systemu operacyjnego może zdefiniować w jaki sposób mają być wyświetlane foldery oraz pliki w nich zawarte. Można dowolnie eksperymentować z tymi ustawieniami, gdyż zawsze jest możliwość powrotu do ustawień domyślnych.



Aby zmienić opcje folderów wybierz kolejno:

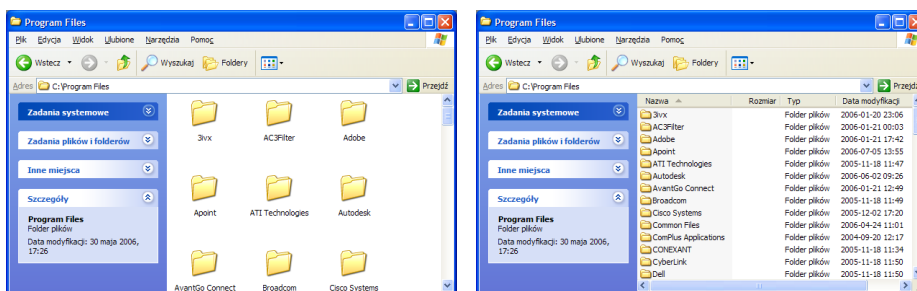
Eksplorator systemu ► Narzędzia ► Opcje folderów

Klasyczna konfiguracja zakłada korzystanie z jednego okna. W takim przypadku otwarcie podfolderu wyświetlanego w danym oknie powoduje zmianę zawartości bieżącego okna. Alternatywne rozwiązanie powoduje wyświetlenie zawartości podfolderu w nowo otwartym oknie.

Użytkownik ma też możliwość wymuszenia otwarcia zawartości podfoldera w nowym oknie. W tym celu należy w momencie otwierania podfoldera przytrzymać wciśnięty klawisz *Shift*. Przytrzymanie klawisza *Shift* w momencie zamykania okna podfoldera spowoduje zamknięcie kaskady okien, które zostały otwarte podczas zagłębiania się w strukturze folderów.

3.2.2 Widoki

Możliwe jest wyświetlanie zawartości folderów w oparciu o różne widoki. Każdy z widoków kładzie nacisk na inne aspekty wyświetlanych danych.



Aby zmienić sposób wyświetlania zawartości foldera wybierz kolejno:

Eksplorator systemu ► Widok ►

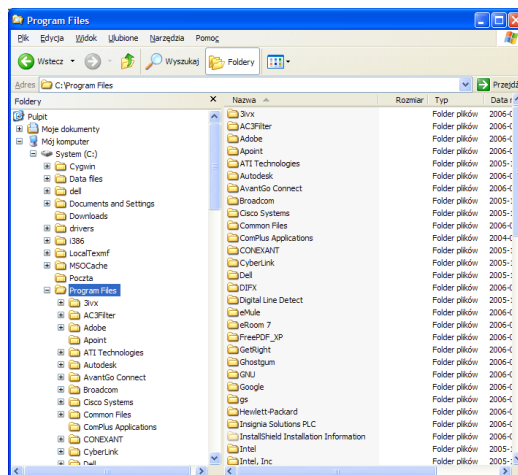
Miniatury, Kafelki, Ikony, Lista lub Szczegóły

Na uwagę zasługuje widok *Miniatury*, w którym wyświetlane są duże ikony, które w przypadku przechowywania w folderze danych multimedialnych zawierają podgląd próbki przechowywanych tam danych.

Widoku *Szczegóły* wyświetla podstawowe informacje statystyczne o folderach i plikach, takie jak rozmiar, rozszerzenie, data utworzenia. Pozwala to w szczególności wygodnie sortować zawartość folderu w oparciu o wyświetlane atrybuty. Sortowanie jest możliwe zarówno w porządku rosnącym jak też malejącym, zmiana sortowania następuje po kliknięciu na nagłówek kolumny, względem której sortowanie ma miejsce. Po kliknięciu zmianie ulega strzałka wyświetlana obok nazwy kolumny symbolizująca kierunek sortowania.

3.2.3 Drzewo katalogowe

Drzewo katalogowe można przeglądać w pełnej postaci za pomocą *Eksploatora systemu*. Podstawowymi operacjami w trakcie przeglądania drzewa katalogowego są rozwijanie i zwijanie gałęzi drzewa.

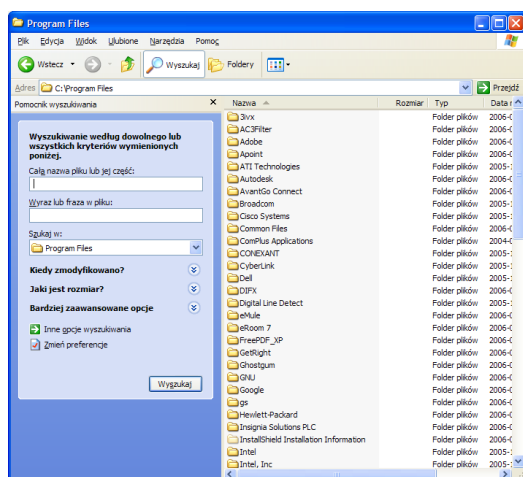


Aby wyświetlić drzewo katalogowe wybierz kolejno:

Eksploator systemu ► Widok ► Pasek eksploratora ► Foldery

3.2.4 Wyszukiwanie

Użytkownik systemu operacyjnego ma możliwość przeszukiwania zasobów komputera. Można wyszukiwać pliki znając jedynie część ich nazwy lub znając frazę znajdującą się w (treści) pliku. Obszar wyszukiwania można ograniczyć do określonego folderu.



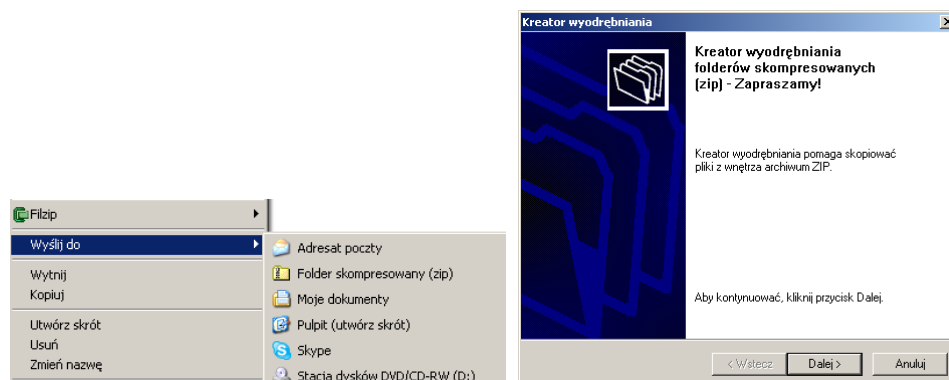
Aby wyświetlić opcje wyszukiwania wybierz kolejno:

Eksplorator systemu ► Widok ► Pasek eksploratora ► Wyszukaj

3.3 Kompresja

3.3.1 Tworzenie folderów skompresowanych

Kompresja polega na takiej zmianie sposobu zapisu zbioru danych, by objętość danych uległa zmniejszeniu, a jednocześnie by (przynajmniej w typowym przypadku) nie utracić informacji zawartych w zbiorze danych. Takie zmniejszenie rozmiaru danych pozwala na lepsze wykorzystanie przestrzeni dyskowej, łatwiej też przesłać dane w postaci skompresowanej przez sieć komputerową (np. jako załączniki do poczty elektronicznej).



Aby skompresować zasób wybierz kolejno:

Menu podręczne ► Wyślij do ► Folder skompresowany

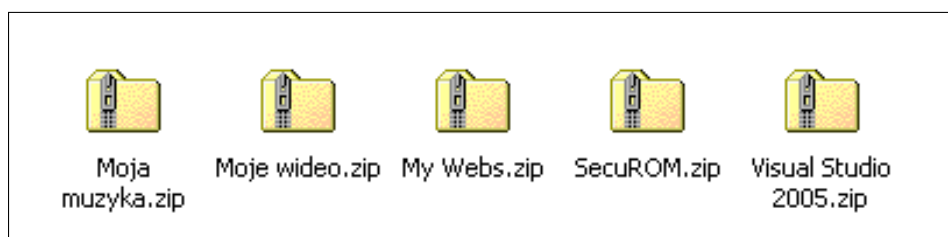
Aby zdekompresować zasób wybierz kolejno:

Menu podręczne ►Wyodrębnij wszystkie

Zazwyczaj użytkownicy zainteresowani są kompresją bezstratną, w której dekompresja postaci skompresowanej pozwala odzyskać postać pierwotną. Praktykuje się również kompresję stratną, w której pełne odzyskanie informacji nie jest możliwe. W kompresji stratnej zachowywane są jedynie główne właściwości kompresowanego zbioru, które interesują użytkownika. Kompresja stratna ma zastosowanie przede wszystkim w kompresji dźwięku lub obrazu, natomiast nie nadaje się do kompresji tekstu.

Najczęściej spotykanym formatem kompresji (bezstratnej) jest obecnie ZIP. Narzędzie do kompresji w formacie ZIP jest wbudowane w system operacyjny Microsoft Windows. Aby skompresować dany zasób należy wskazać go myszą i z menu podręcznego wyświetlanego po kliknięciu zasobu prawym przyciskiem myszy wybrać opcję *Wyślij do ►Folder skompresowany*.

Po skompresowaniu folder wygląda podobnie do folderu oryginalnego i w podobny sposób można się nim posługiwać, w szczególności można go otwierać, przeglądać i zmieniać jego zawartość.



Foldery skompresowane

Aby przeprowadzić dekompresję należy z menu podręcznego wyświetlanego po kliknięciu zasobu prawym przyciskiem myszy wybrać opcję *Wyodrębnij wszystkie*, co spowoduje uruchomienie prostego kreatora dekompresji.

3.4 Akcesoria

3.4.1 Notatnik

Aplikacją pozwalającą na prostą edycję tekstu jest *Notatnik*.



Aby uruchomić notatnik wybierz kolejno:

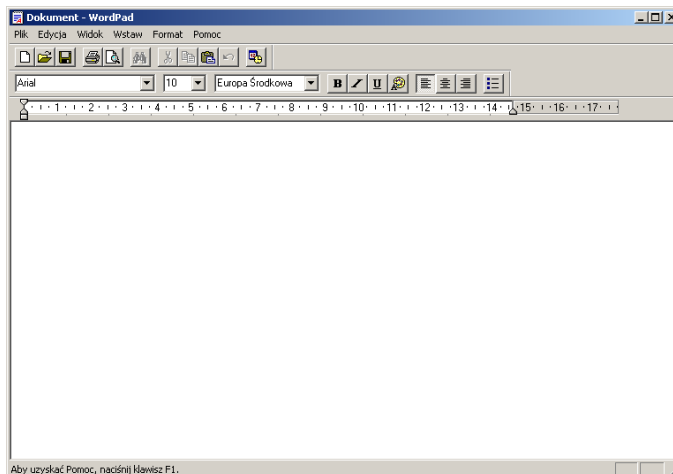
Menu Start ► Programy ► Akcesoria ► Notatnik

Notatnik można uruchomić poleceniem z menu *Start* lub wpisując polecenie *notepad* w oknie szybkiego uruchamiania.

Ćwiczenie 3.4.1. Należy stworzyć nowy plik tekstowy zawierający imię i nazwisko studenta i zachować go na pulpicie pod nazwą *osoba.txt*.

3.4.2 Wordpad

Wordpad jest edytorem tekstu posiadającym dużo więcej opcji niż *Notatnik*. W szczególności *Wordpad* umożliwia składanie czcionek, tworzenie akapitów i wcięć, wstawianie aktualizowanej daty i godziny oraz wstawianie obiektów zagnieżdżonych dostępnych w systemie.



Aby uruchomić edytor Wordpad wybierz kolejno:

Menu Start ► Programy ► Akcesoria ► WordPad

3.4.3 Kalkulator

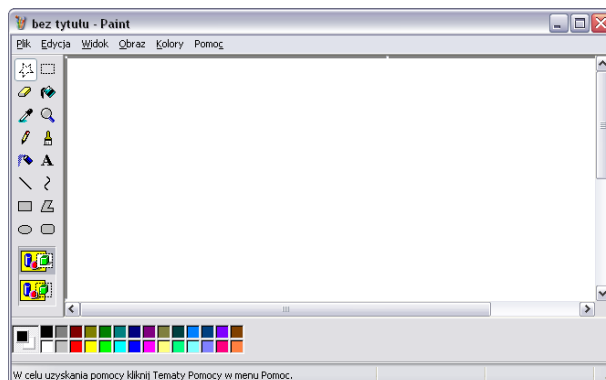


Aby uruchomić kalkulator wybierz kolejno:

Menu Start ►Programy ►Akcesoria ►Kalkulator

Domyślnie kalkulator jest uruchamiany w trybie standardowym, ale menu *Widok* kalkulatora umożliwia przełączenie do trybu naukowego.

3.4.4 Paint

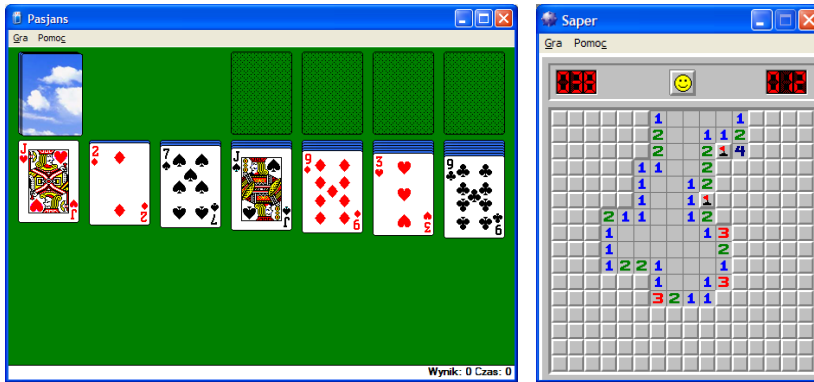


Aby uruchomić edytor rysunków wybierz kolejno:

Menu Start ►Programy ►Akcesoria ►Paint

3.4.5 Gry

Najczęściej dostępnymi wraz z systemem operacyjnym programami rozrywkowymi są różnego rodzaju gry. Najpopularniejsze z nich to *Pasjans* (w różnych odmianach) oraz *Saper*.



Aby uruchomić grę Pasjans lub Saper wybierz kolejno:

Menu Start ► Programy ► Gry ► Pasjans lub Saper

Rozdział 4

Przeglądarki internetowe

4.1 Wprowadzenie

Przeglądarki internetowe należą obecnie do najczęściej wykorzystywanych aplikacji komputerowych i są używane do pobierania z sieci i wyświetlania dokumentów utworzonych w językach HTML, XHTML, XML. Ogólnie mówiąc, przeglądarki służą do oglądania stron internetowych (nazywanych też stronami sieci Web) publikowanych w sieci.

Pod pojęciem strony internetowej należy rozumieć dokument (plik) lub zbiór dokumentów (plików) umieszczonych na serwerze lub nawet na dysku lokalnym komputera, utworzony w jednym z ww. języków. Cechą charakterystyczną stron internetowych jest tzw. hipertekstowość, która oznacza sposób organizacji treści. Treść ta może być dzielona na niezależne części powiązane hiperłączami (odnośnikami, linkami).

Niniejszy moduł pozwoli *aktywnie studiującej* osobie osiąść umiejętności pozwalające na:

- korzystanie z przeglądarek internetowych
- zarządzanie zakładkami
- konfigurację serwerów proxy
- zapamiętywanie danych formularzy i haseł
- pobieranie dokumentów
- korzystanie z wyszukiwarek internetowych

Kliknięcie odnośnika powoduje przejście do innej, określonej przez twórcę, części strony. Język HTML umożliwia tworzenie dokumentów w ww. sposób. Mogą one zawierać odnośniki do dowolnych dokumentów umieszczonych na tym samym bądź innym serwerze. Bardzo często strony internetowe zawierają też elementy graficzne (obrazy, animacje) i dźwiękowe, m.in. w formatach gif, jpeg, png, flash. Zbiór stron internetowych powiązanych tematycznie lub funkcjonalnie, nazywany bywa serwisem WWW, lub witryną

Słowa te można traktować jako synonimy.

internetową. Skrót WWW (World Wide Web – ang. światowa pajęczyna) można pojmować jako ogólnoświatowy system informacyjny, który składa się ze stron internetowych (stron WWW) połączonych za pomocą hiper-tekstowych odnośników i znajdujących się na serwerach mających dostęp do Internetu.

- Pobieranie dokumentów (plików) z serwerów sieciowych
- Przetwarzanie pobranych dokumentów i wyświetlenie rezultatu
- Zapewnienie funkcji zwiększających wygodę przeglądania stron sieci Web

4.1.1 Krótka historia przeglądarek internetowych

- 1991. WorldWideWeb
 - 1993. Mosaic
 - 1994. Opera
 - 1994. Netscape Navigator
 - 1995. Microsoft Internet Explorer
 - 1998. Mozilla
-
- 1991. WorldWideWeb. Pierwsza przeglądarka w ogóle, autorstwa Tima Berners-Lee z CERN, który stworzył także WWW
 - 1993. Mosaic. Pierwsza przeglądarka, która była pod względem wyglądu i funkcjonalności podobna do obecnie używanych, powstała dla uniksowych systemów operacyjnych i środowiska X Window System. Mosaic obsługiwał grafikę, dźwięk, wideo, formularze, posiadał możliwość stosowania zakładek i przechowywał historię odwiedzanych stron
 - 1994. Opera. Powstała w Norwegii, co ciekawe, wcześniej niż np. Internet Explorer
 - 1994. Netscape Navigator. Powstał w firmie Netscape Communications Corporation
 - 1995. Microsoft Internet Explorer.
 - 1998. Mozilla. W 1998 roku firma Netscape Communications Corporation udostępniła kod źródłowy Netscape Navigatora, co doprowadziło do powstania przeglądarki Mozilla, później powstał bazujący na Mozilli – Firefox

4.1.2 Protokoły komunikacyjne

Aby uzyskać dostęp do strony WWW, trzeba posłużyć się programem komputerowym, którym jest właśnie przeglądarka internetowa. Przeglądarka łączy się z serwerem internetowym, skąd pobiera pewien zbiór informacji nazywany stroną internetową. Strona taka może zostać wyświetlona, zapisana na dysku lokalnym komputera lub np. wydrukowana. Strona internetowa jest hipertekstem, co oznacza, że użytkownik oglądając ją, może używać odnośników, które przenoszą go do innych stron, umieszczonych na tym samym lub innym serwerze.

Odnośniki mogą nas kierować do dowolnego serwera dostępnego w Internecie.

4.1.3 Standardowe protokoły sieciowe

Aby strona mogła być wyświetlona przez przeglądarkę, musi zostać przez nią pobrana z serwera. W tym celu przeglądarki korzystają z komunikacyjnych protokołów sieciowych, określających reguły transmisji danych - grup plików, tworzących stronę internetową - pomiędzy serwerem a przeglądarką.

- HTTP (Hypertext Transfer Protocol)
 - protokół przesyłania hipertekstu
- HTTPS (HTTP Secure)
 - protokół HTTP z szyfrowaniem
- FTP (File Transfer Protocol)
 - protokół przesyłania plików

Protokół HTTP został specjalnie stworzony do przesyłania przez sieć stron internetowych. Obsługuje transfer plików oraz dodatkowych informacji (np. o kliknięciu odnośnika). HTTPS pozwala dodatkowo szyfrować przesyłane dane. Jest powszechnie używany np. do obsługi naszych kont bankowych za pomocą przeglądarki internetowej. FTP służy do przesyłania jedynie plików. Możemy się połączyć za pomocą naszej przeglądarki z serwerem FTP, ale nie zobaczymy wtedy strony WWW, tylko listę plików i katalogów. Usługi ftp dzielimy na:

- Publiczne – dostępne dla wszystkich, którzy znają jedynie nazwę lub adres serwera
- Niepubliczne – wymagające autoryzacji

4.1.4 URL (Uniform Resource Locator)

Aby obejrzeć stronę internetową, należy podać jej adres w okienku adresu przeglądarki. Adres ten musi zostać wpisany w formacie zgodnym z obowiązującym standardem URL.

- Adresy zgodne z URL:
 - typ-zasobu://nazwa-serwera/sciezka-dostepu/
 - typ-zasobu://nazwa-uzytownika@nazwa-serwera/sciezka-dostepu/
 - typ-zasobu://nazwa-uzytownika:haslo@nazwa-serwera/sciezka-dostepu/
- Przykład adresu w postaci zgodnej z URL
 - <http://www.mimuw.edu.pl/wydzial>
- Przykład adresu w postaci zgodnej z URL z użyciem protokołu z szyfrowaniem:
 - <https://gmail.com>
- URL (Uniform Resource Locator) jest uniwersalnym formatem odnośników do zasobów (np. serwerów oraz plików na serwerach)
- URL to po kolei: nazwa typu zasobu, dwukropek oraz część która zależy od typu zasobu, może to być np. nazwa serwera
- Typ zasobu określa najczęściej nazwę protokołu sieciowego wykorzystywanego dla udostępnienia zasobu. W przypadku przesyłania stron WWW jest to protokół HTTP lub HTTPS
- Jeśli zasób został udostępniony za pomocą protokołu FTP, może być konieczna autoryzacja. Należy wtedy podać nazwę użytkownika i hasło w sposób zgodny z URL. Najlepiej jednak podać tylko nazwę użytkownika. Wtedy przeglądarka wyświetli okienko dialogowe, a w odpowiednim polu tego okienka będzie można podać hasło. Unikniemy wtedy bezpośredniej „widoczności” hasła w zapamiętanej historii (liście) odwiedzanych stron WWW lub serwerów FTP
- Przeglądarki internetowe często pozwalają na pomijanie rodzaju zasobu w adresie URL (czyli umożliwiają stosowanie adresów niezgodnych z URL), wtedy jednak przeglądarka najczęściej zakłada, że chcemy skorzystać z zasobu udostępnianego przez HTTP. Jeśli więc chcemy rzeczywiście użyć FTP, np. gdy wiemy, że serwer obsługuje jedynie FTP, trzeba podać pełny URL: <ftp://ftp.icm.edu.pl/>, natomiast aby obejrzeć witrynę Wydziału Matematyki UW wystarczy wpisać w pasku adresu: www.mimuw.edu.pl.

Jak już wiemy, adres URL zawiera nazwę serwera, ale trzeba pamiętać, aby zawsze stosować jej pełną postać, tzn. taką, która zawiera również nazwę domeny. Tylko pełna nazwa umożliwi dotarcie do maszyny w Internecie. Np. nazwę www.mimuw.edu.pl, można podzielić na część domenową (mimuw.edu.pl) i właściwą nazwę serwera, w tym przypadku www. W internecie serwer ten jest jednoznacznie identyfikowany po podaniu pełnej nazwy (wraz z domeną) www.mimuw.edu.pl. W jednej domenie może istnieć wiele serwerów, np. duch.mimuw.edu.pl, students.mimuw.edu.pl.

Zauważmy, że w adresach stron internetowych rzeczywiście często stosuje się frazę *www*. Wspomniano już, że oznacza to dokładnie tyle, że serwer na którym jest umieszczona strona, ma nazwę *www*. Dla maszyn utrzymujących strony WWW przyjęło się stosowanie takiej właśnie nazwy. Nie jest to jednak konieczne. Dodatkowo istnieje możliwość pomijania nazwy serwera w adresie i używania jedynie nazwy domeny, ale zależy to od konfiguracji serwera nazw (DNS, ang. Domain Name Server) obsługującego domenę.

Do zamiany pełnej nazwy domenowej na tzw. na adres IP służy DNS, czyli *serwer nazw*. Komputery w komunikacji przez sieć posługują się właśnie tymi adresami – IP określa jednoznacznie urządzenie w sieci Internet. Człowiek jednak dużo łatwiej zapamięta nazwę np.: *www.mimuw.edu.pl* niż adres IP, który zawsze składa się z liczb, np. 193.0.96.15. DNS pozwala na posługiwanie się nazwami w adresach URL stron internetowych i zwalnia nas z konieczności pamiętania adresów IP. Podsumowując: po podaniu adresu URL strony internetowej, np. *www.mimuw.edu.pl* w okienku adresu przeglądarki, musi nastąpić zamiana nazwy serwera znajdującej się w tym URL-u na jego adres IP. Wtedy dopiero może nastąpić połączenie za pośrednictwem sieci i w efekcie wyświetlenie strony. Proces zamiany nazwy na adres IP odbywa się w sposób całkowicie automatyczny – stacja robocza z której korzystamy odpytuje serwer DNS, który zwraca wynik w postaci adresu IP, odpowiadającego podanej w pytaniu nazwie.

W szczególności należy pamiętać iż adres *http://www.mimuw.edu.pl*, nie musi (choć może) kierować do tej samej strony WWW co *http://mimuw.edu.pl*. Podsumowując, można stwierdzić, że czasami jest możliwe odwoływanie się do stron WWW przez podanie samej nazwy domeny, np. *http://wp.pl* (adres *http://wp.pl* nie zawiera nazwy serwera!). Związana jest z tym pewna zaleta – adres URL ulega skróceniu. Łatwiej i szybciej napiszemy w okienku adresu przeglądarki *wp.pl*, niż *www.wp.pl*.

4.1.5 Technologie obsługiwane przez współczesne przeglądarki

- Protokoły
 - HTTP, FTP, HTTPS
- Języki
 - HTML
 - JavaScript
- Kaskadowe arkusze stylów
 - CSS (ang. Cascading Style Sheets)
- Formaty graficzne
 - GIF, PNG, JPEG
- Ciasteczka (ang. cookies)

- Pluginy
 - Flash
 - Aplety Java

Obsługa protokołów HTTP, FTP, HTTPS oraz języka opisu strony HTML jest standardem we współczesnych przeglądarkach internetowych.

Przeglądarki potrafią też wyświetlać grafikę w różnych formatach.

Nowoczesne przeglądarki obsługują również tzw. kaskadowe arkusze stylów (CSS). CSS jest językiem służącym do opisu wyglądu wyświetlanej treści stron WWW. Arkusz CSS to zbiór reguł określających np. w jaki sposób ma być pokazywany przez przeglądarkę tekst lub inny element, znajdujący się w dokumencie utworzonym w języku HTML, XHTML lub XML. W ten sposób można definiować wszystkie parametry odpowiedzialne za wygląd treści umieszczonych w pliku HTML, np. wielkość i styl czcionki, kolory, marginesy, odstępy itp. oraz położenie elementów względem siebie oraz względem strony. Wykorzystanie arkuszy stylów daje ogromne możliwości kontroli nad organizacją zawartości strony. Taką funkcjonalność oferuje też sam język HTML, ale nie zaleca się używania mechanizmów HTML służących do formatowania wyglądu strony. Do tego służy właśnie CSS.

Inne obsługiwane technologie to JavaScript, flash, ciasteczka i aplety.

JavaScript jest językiem programowania stworzonym przez firmę Netscape, i jest technologią w zasadzie pomocniczą, chętnie jednak wykorzystywaną przez twórców stron WWW. Skrypty pisane w JavaScript służą najczęściej do tworzenia stron interaktywnych, zawierających formularze HTML oraz elementy nawigacyjne (np. rozwijane menu). JavaScriptem można np. sprawdzić poprawność danych wpisywanych do formularza. Obsługa JavaScriptu w przeglądarkach internetowych pozwala też na modyfikowanie zawartości dokumentu oraz tworzenie i odczyt tzw. ciasteczek, zarządzanie oknami przeglądarki i wyświetlanie okien dialogowych.

Oraz np. zmianę rozmiaru okna.

Flash jest technologią tworzenia animacji graficznych, przeznaczonych do osadzenia na stronie internetowej i odtwarzania za pomocą przeglądarki. Pliki utworzone w technologii Flash często zawierają reklamy internetowe, prezentacje, czasem też np. dokumentacje z animowanymi elementami graficznymi i są częścią wielu stron WWW. Flash został stworzony przez firmę Macromedia. Posiada też sporo wad, m.in. zużywa dużo zasobów komputera klienta (tego który za pomocą przeglądarki, wyświetla zawartość pliku flash, np. animację).

Ciasteczka (cookies) to krótkie informacje tekstowe, wysyłane przez serwer WWW i zapisywane na komputerze użytkownika korzystającego z witryny obsługującej ten mechanizm. Ciasteczka mogą zawierać różne informacje o użytkowniku strony WWW i czynnościach, które wykonał, łącząc się z tą stroną. Zazwyczaj wykorzystywane są do automatycznego rozpoznawania danego użytkownika (a właściwie tylko jego przeglądarki) przez witrynę. Dzięki temu, może on oglądać stronę spersonalizowaną, np. o dostosowanym wyglądzie lub innych elementach. Umożliwia to utworzenie dedykowa-

nego serwisu WWW, który będzie mógł wyświetlać treść, uzależnioną od odwiedzającej serwis osoby. Gdy użytkownik ponownie skorzysta ze strony obsługującej cookies, na żądanie tej strony przeglądarka odsyła poprzednio zapisane na dysku ciasteczka. Użytkownik ogląda serwis np. w kolorze tła, który wcześniej wybrał, lub też z informacjami, które podał. Przyjęcie takiego założenia w procesie tworzenia stron WWW wydaje się wygodne dla użytkownika, gdyż zwalnia go z konieczności każdorazowego powtarzania pewnych czynności. Mechanizm ciasteczek przesyła witrynie jedynie informacje podane przez użytkownika w czasie, gdy ją odwiedzał. Mogą one jednak zostać wykorzystane niezgodnie z jego wolą i bez jego wiedzy, do utworzenia np. sondy o preferencjach dotyczących np. wyboru towarów w sklepie internetowym. Witryna nie pobiera żadnych dodatkowych informacji z komputera klienta. Mechanizm ten nie stanowi więc bardzo istotnego problemu związanego z bezpieczeństwem danych, należy jednak zaznaczyć, iż działa w zasadzie bez nadzoru użytkownika. Poważną wadą mechanizmu ciasteczek jest to, iż serwer może rozpoznać tylko przeglądarkę, a nie właściwą osobę. Cookies nie sprawdzają się więc, jeśli korzystamy z kilku przeglądarek, albo tej samej przeglądarki używa wiele osób.

Aplet Java jest to program napisany w języku Java. Aplety mogą być osadzone na stronach WWW i uruchomione przez przeglądarkę internetową. Mechanizm ten nie jest obecnie często stosowany, choć kiedyś był dość popularny.

4.2 Używanie przeglądarki internetowej

Wszystkie przykłady w niniejszym rozdziale zostały przygotowane w oparciu o przeglądarkę Mozilla Firefox.

Określenie *open source* oznacza, że kod źródłowy oprogramowania jest publicznie dostępny. W Konsekwencji zwykle, oprogramowanie takie jest rozwijane przez wiele osób, także firm lub ochotników i można go używać bez ponoszenia jakichkolwiek kosztów. Dostępność kodu ułatwia również wykrywanie błędów, oprogramowanie open source często bywa znakomitej jakości.

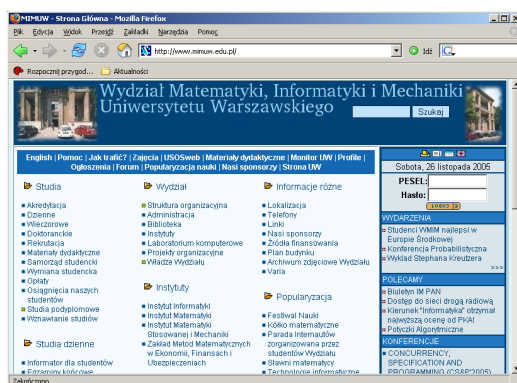
Przykładem świetnej aplikacji open source jest właśnie przeglądarka Firefox. Jest to program wywodzący się z przeglądarki Mozilla, można go pobrać ze strony:

<http://www.firefox.pl>

i używać zupełnie za darmo!

□ **Ćwiczenie 4.2.1.** Należy wykonać ćwiczenie z posługiwania się i konfiguracji przeglądarki według opisów przedstawionych na slajdach. □

4.2.1 Główne okno programu Firefox



- Wystarczy wpisać adres strony w polu adresu
- Ikony ułatwiają nawigację
 - np. wyświetlenie poprzednio oglądanej strony

4.2.2 Wyszukiwanie, panele, zakładki

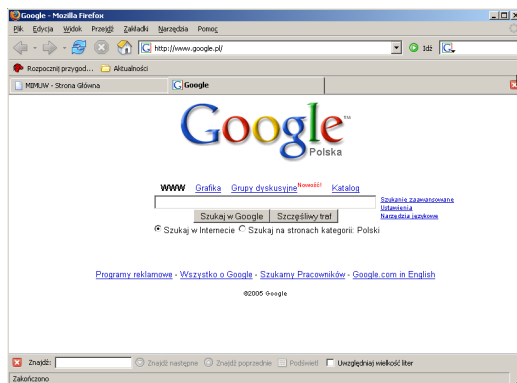
Przeglądarka Firefox umożliwia wyświetlanie stron w tzw. panelach

menu Plik ► Nowy panel

Możemy też wyświetlić stronę w nowym oknie

menu Plik ► Nowe okno

Można też łatwo wyszukiwać ciągi znaków na stronie oraz zapamiętywać adresy stron w folderze zakładek.



- Aby wyświetlić stronę w panelu lub w nowym oknie należy wybrać:

menu Plik ► Nowy panel lub Nowe okno

- Aby wyszukać ciąg znaków na stronie należy wybrać:

menu Edycja ► Znajdź na tej stronie

Menu *Przejdź* pozwala na dostęp do poprzednio oglądanych stron. Aby zobaczyć pełną listę odwiedzonych stron internetowych, należy wybrać z menu *Przejdź* polecenie *Historia*.

Warto korzystać z tzw. paneli (kart). Jeśli chcemy rozpocząć oglądanie nowej strony WWW, nie zamykając dotychczas wyświetlanej, możemy ją otworzyć w nowym oknie

Plik ► Nowe okno

lub w nowym panelu

Plik ► Nowy panel

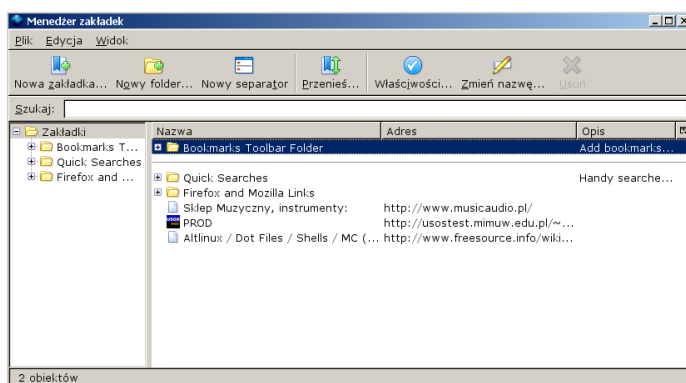
Użycie paneli jest dużo wygodniejsze. Nie musimy nawigować między wieloma otwartymi oknami. Do wszystkich wyświetlonych stron mamy łatwy dostęp z głównego okna przeglądarki.

Większość współczesnych przeglądarek obsługuje panele (np. Firefox, SeaMonkey, Opera). Wyjątkiem jest Internet Explorer, który niestety nie posiada tej funkcji.

4.2.3 Zakładki

Zakładki są bardzo użyteczne. Stosuje się je w celu zapisania w przeglądarce adresów URL stron, które będziemy chcieli w przyszłości ponownie odwiedzić. Zarządzanie zakładkami jest zbliżone do zarządzania plikami i katalogami w systemie plików

Używajmy narzędzia *Zakładki!* Oszczędzi nam to ponownego i być może pracochłonnego wyszukiwania interesujących nas informacji.



- Aby dodać bieżącą stronę do zakładek należy wybrać kolejno:

Menu Zakładki ►Dodaj do zakładek

- Aby zarządzać zakładkami należy wybrać kolejno:

Menu Zakładki ►Menedżer zakładek

Menedżer zakładek przypomina narzędzia do zarządzania plikami i katalogami w systemie plików. Możemy dodawać zakładki (klikając *Nowa zakładka*), w tym przypadku należy podać nazwę zakładki oraz adres strony z którą zakładka ma być powiązana. Stronę też dodać stroną bieżącą w momencie gdy jest one wyświetlona w przeglądarce. Nie ma wtedy potrzeby używania menadżera zakładek – wystarczy z menu *Zakładki* wybrać opcję *Dodaj do zakładek*.

Przy tworzeniu zakładki należy także podać słowo kluczowe (związane z tematyką strony), ułatwi to późniejsze jej odnalezienie (należy skorzystać z paska *Szukaj*). Aby zmienić, lub dodać słowo kluczowe dla już istniejącej zakładki – np. dodanej za pomocą

Zakładki ►Dodaj do zakładek

należy kliknąć prawym klawiszem myszy na danej zakładce i wybrać *Właściwości*. Ukáže się okienko, do którego będzie można wprowadzić słowo kluczowe.

Za pomocą menedżera zakładek można tworzyć katalogi (klikając *Nowy folder*), oraz przenosić zakładki do katalogów (używając *Przenieś*). Należy dążyć do grupowania w katalogach zakładek do stron o podobnej tematyce, ułatwi nam to ich późniejsze odnalezienie. Możemy np. utworzyć katalog *studia* i dodawać tam zakładki do stron związanych z nauką na uczelni.

Nazwy *katalog* i *folder* są synonimami.

4.3 Konfiguracja przeglądarki internetowej

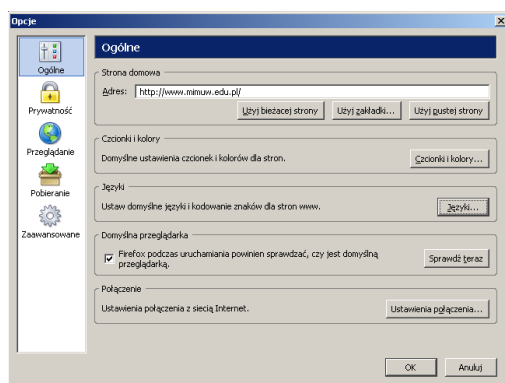
Przykłady konfiguracji oparto o przeglądarkę Mozilla Firefox w wersji 1.0.5. W innych wersjach układ interfejsu użytkownika może się różnić. Słowo *zakładka* będzie stosowane również w odniesieniu do elementów interfejsu przeglądarki, a nie tylko do narzędzia opisywanego wcześniej, służącego do przechowywania adresów stron WWW.

Firefox jest bardzo dobrą, niewątpliwie wartą polecenia przeglądarką, obsługującą wszystkie współczesne technologie wykorzystywane przy tworzeniu stron WWW. Dlatego zagadnienia konfiguracyjne zostaną omówione na jego przykładzie.

4.3.1 Ustawienia ogólne

Opcje konfiguracji są dostępne z menu głównego

Narzędzia ► Opcje



Aby wyświetlić okno konfiguracji podstawowych parametrów programu:

Narzędzia ► Opcje ► Zakładka Ogólne

Można ustawić adres strony, która będzie się wyświetlać po uruchomieniu programu. Można zmieniać czcionki i kolory oraz ustawienia językowe. Ma to wpływ na wygląd wyświetlanych stron (możemy np. zwiększyć czcionkę). Można też skonfigurować tzw. serwer pośredniczący do połączenia z Internetem (przycisk *Ustawienia połączenia*).

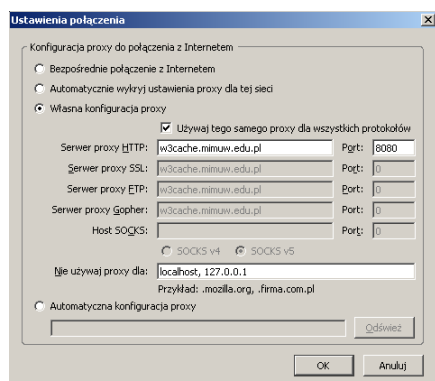
Funkcja, którą spełnia serwer pośredniczący (ang. proxy) jest następująca:

- Serwer proxy łączy się ze stronami internetowymi, nie robi tego bezpośrednio przeglądarka.
- Przeglądarka łączy się natomiast tylko z serwerem proxy

Serwerów proxy używa się z następujących powodów:

- Pozwalają na dostęp do stron, które są widoczne jedynie z określonych adresów IP (często serwer proxy wymaga autoryzacji). Przykładem może być następująca sytuacja: Biblioteka Uniwersytetu Warszawskiego umożliwia darmowy dostęp do czasopism on-line, jednak tylko z sieci Uniwersytetu Warszawskiego. Aby pracownicy mieli dostęp do ww. czasopism z domu, muszą skonfigurować w przeglądarce serwer pośredniczący, który jest uruchomiony na komputerze podłączonym do sieci UW.
- Oprócz ww. funkcji serwery pośredniczące pozwalają na szybszy dostęp do stron WWW. Serwer pośredniczący przechowuje na dysku twardym serwera elementy oglądanych wcześniej stron, co oznacza, że aby dany element pojawił się w przeglądarce użytkownika, nie musi on być pobrany z często odległego miejsca. Zostanie jedynie przesłany z serwera pośredniczącego do przeglądarki. Ma to sens przy założeniu, że serwer proxy jest podłączony szybkim łączem do komputera, na którym jest uruchomiona przeglądarka.

Konfiguracja serwera proxy

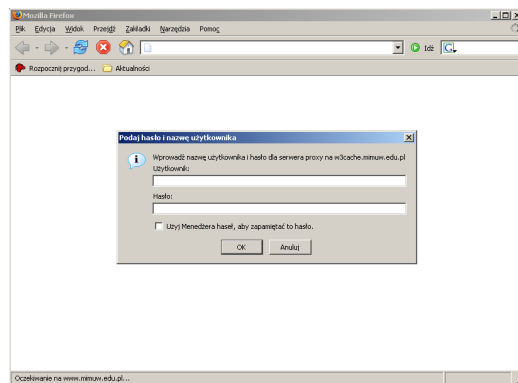


Aby skonfigurować serwer proxy należy wybrać kolejno:

okno konfiguracji ► Ustawienia połączenia ► Własna konfiguracja proxy

- W niektórych sieciach proxy może zostać skonfigurowane automatycznie przez przeglądarkę
- Można skonfigurować serwer proxy osobno dla każdego z protokołów obsługiwanych przez przeglądarkę. W tym przypadku wybieramy tę samą nazwę serwera proxy dla wszystkich protokołów
- Można nie używać proxy dla wybranych domen internetowych (pole *Nie używaj proxy dla*)

Autoryzacja do serwera proxy



Proxy może wymagać autoryzacji

- w3cache.mimuw.edu.pl

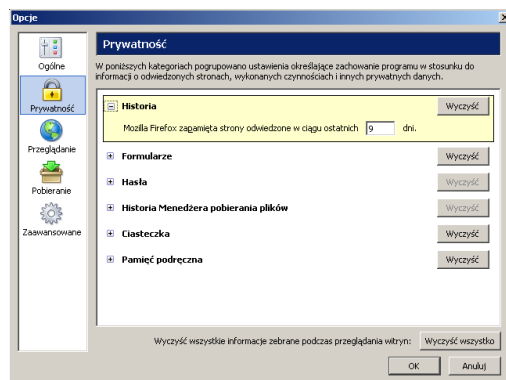
Aby zapamiętać hasło należy zaznaczyć odpowiednią opcję

- Parametry konfiguracji można otrzymać od administratora sieci

W niektórych sieciach administratorzy konfiguruje tzw. przezroczyste proxy. Oznacza to, że użytkownik nie musi używać ręcznej (ani jakiegokolwiek) konfiguracji w przeglądarce, a mimo to serwer proxy jest używany.

4.3.2 Prywatność

Historia odwiedzonych stron

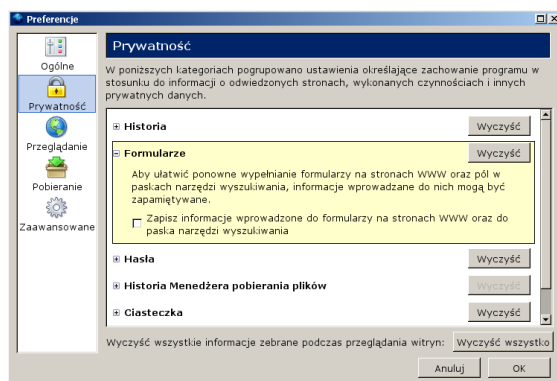


Aby skonfigurować czas przez który będą przechowywane adresy oglądanych stron należy wybrać:

zakładka Prywatność ► pole Historia

Jeśli ktoś inny korzysta z naszej przeglądarki i nie chcemy, aby poznał adresy odwiedzanych przez nas stron, możemy w każdej chwili wyczyścić zapamiętane adresy używając przycisku *Wyczyść*. W takim przypadku należy rozważyć także skrócenie czasu ich przechowywania.

Zapamiętywanie danych formularzy

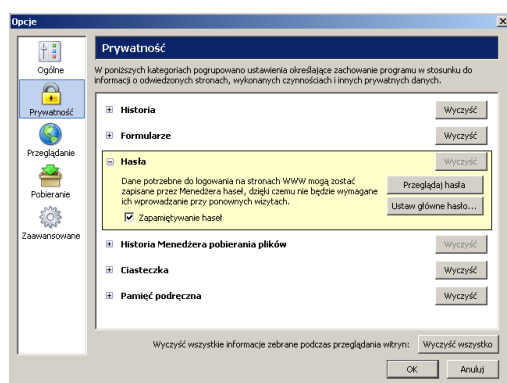


Aby uruchomić zapamiętywanie danych wprowadzanych do formularzy na stronach WWW należy kolejno wybrać:

zakładka Prywatność ► Zapisz informacje...

Przechowywanie przez przeglądarkę danych wprowadzanych do formularzy na stronach WWW czasem nie jest pożądane ze względów bezpieczeństwa, gdyż informacje te są przechowywane na dysku lokalnym naszego komputera. Zapamiętane przez przeglądarkę dane użyte w formularzach można wykasować za pomocą przycisku *Wyczyść* (tak samo jak w przypadku opisanej wcześniej historii odwiedzanych stron).

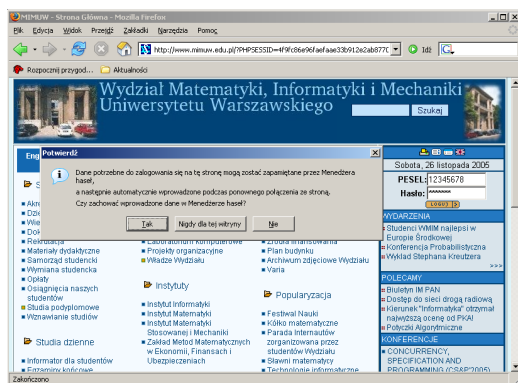
Hasła



Aby uruchomić zapamiętywanie haseł należy kolejno wybrać:

Jeśli zaznaczymy opcję *Zapamiętywanie haseł*, Firefox będzie oferował zapamiętywanie haseł używanych w formularzach na stronie WWW. Można też ustawić hasło główne, które będzie zabezpieczało dostęp do zapamiętanych haseł. Zapamiętanych haseł można pozbyć się przyciskiem *Wyczyść*.

Zapamiętywanie haseł



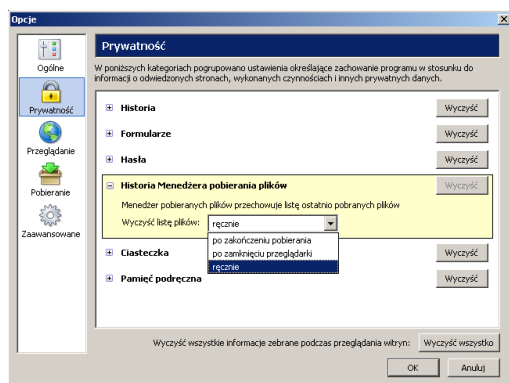
Gdy włączymy zapamiętywanie haseł:

- przy wprowadzaniu hasła do formularza program oferuje jego zapamiętanie
- hasła są przechowywane na dysku lokalnym naszego komputera

Jeśli z naszego komputera, korzystają osoby którym nie ufamy, nie należy używać zapamiętywania haseł w przeglądarce, lub przynajmniej wprowadzić hasło główne!

Dodatkowo, zapisywanie haseł w przeglądarce zwalnia nas z konieczności ich pamiętania. Oznacza to oczywiście wygodę, ale i potencjalny kłopot. Jeśli będziemy chcieli uzyskać dostęp do witryny chronionej hasłem z innego komputera, może się okazać że nie możemy go sobie przypomnieć, gdyż powierzyliśmy pamiętanie hasła przeglądarce na komputerze domowym. Lepiej więc nie zapisywać ważnych haseł w przeglądarce, albo wręcz wyłączyć funkcję przechowywania haseł.

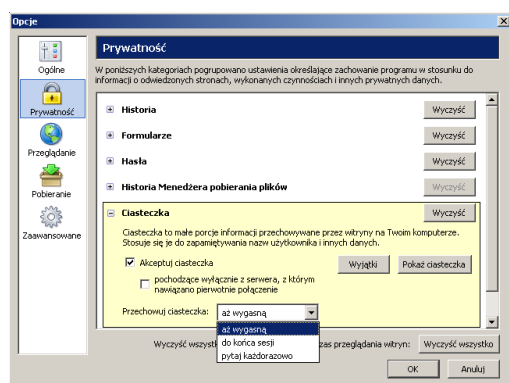
Historia pobierania plików



Wygodnie jest przechowywać listę pobranych plików

- możemy wybrać najdogodniejszy dla nas sposób czyszczenia listy
- warto ustawić ręczne czyszczenie listy

Ciasteczka



Aby zmienić opcje związane z mechanizmem ciasteczek należy kolejno wybrać:

zakładka Prywatność ► Akceptuj ciasteczka

Mechanizm ciasteczek jest bardzo często wykorzystywany przez witryny internetowe, do personalizowania treści wyświetlanych użytkownikowi. Dla wygody korzystania z wielu witryn trzeba więc zaznaczyć opcję *Akceptuj ciasteczka*.

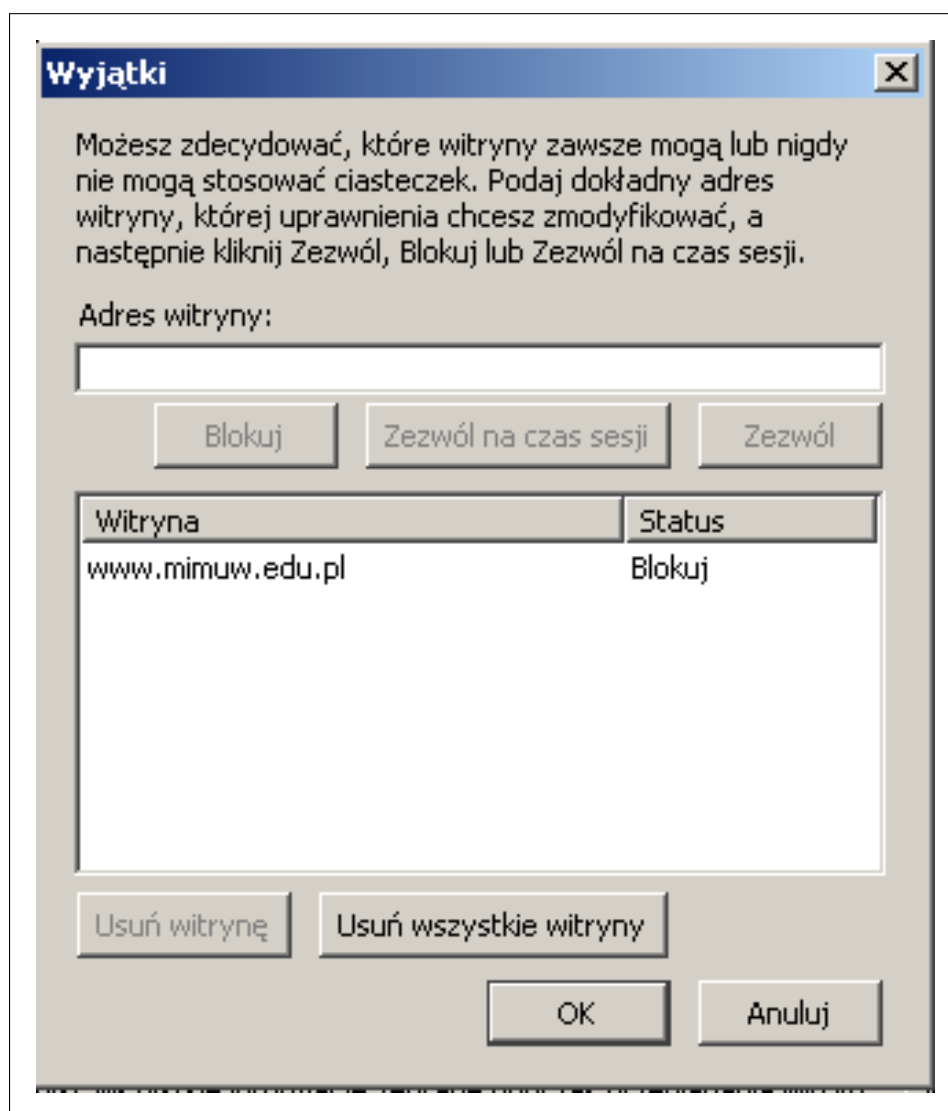
Mechanizm obsługi ciasteczek został skonstruowany w sposób pozwalający na dostęp do danych zawartych w ciasteczku jedynie witrynie, która je utworzyła. Istnieje jednak metoda pozwalająca na korzystanie z informacji w ciasteczkach szerszej grupie witryn. Oznacza to, że wiele stron internetowych może zbierać informacje o użytkowniku do celów np. marketingowych.

Część witryn może nie działać bez włączenia mechanizmu ciasteczek!

Gdy zaznaczymy opcję *pochodzące wyłącznie z serwera, z którym nawiązano połączenie pierwotnie* opisana wyżej sytuacja nie będzie mogła mieć miejsca. Należy więc z tej opcji skorzystać.

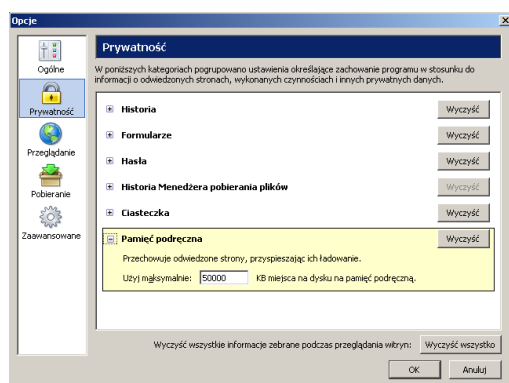
Każde ciasteczko posiada informację o czasie wygasania. Dla własnej wygody użyjmy opcji *Przechowuj ciasteczka: aż wygasną*. Jeśli wybierzemy *do końca sesji*, wprowadzone przez nas ustawienia w witrynach nie będą zapamiętywane i przy następnych odwiedzinach danej strony będzie je trzeba powtórzyć.

Politykę dotyczącą akceptowania ciasteczek można uzależnić od adresu witryny. Służy do tego przycisk *Wyjątki*. Możemy np. nie zezwolić na przyjmowanie ciasteczek przez naszą przeglądarkę od wybranych witryn.



Zezwalanie na przyjmowanie ciasteczek

Pamięć podręczna



Aby zmienić ilość pamięci dyskowej przeznaczonej na przechowywanie elementów stron internetowych należy kolejno wybrać:

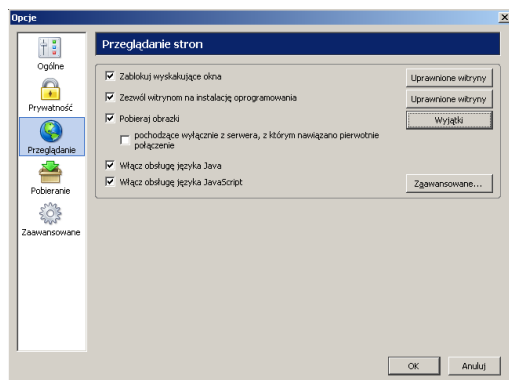
zakładka Prywatność ► Użyj maksymalnie KB...

Przeglądarka internetowa ma możliwość przechowywania elementów stron internetowych na dysku twardym komputera w celu szybszego wyświetlania stron. Mechanizm ten jest nieco podobny do stosowanego w opisywanych wcześniej serwerach proxy

Gdy przypuszczamy, że wyświetlana strona jest nieaktualna (wyświetlona z pamięci dyskowej a nie z serwera), należy kliknąć przycisk *Odśwież bieżącą stronę*. Przeglądarka sprawdzi aktualność strony i w miarę potrzeby odświeży jej elementy

Zalecana wielkość pamięci na przechowywanie stron to 50 do 70MB. Przy zamykaniu przeglądarki, dokonuje ona porządkowania przechowywanych danych. Jeśli kończenie pracy Firefoxa trwa zbyt długo, należy zmniejszyć ilość pamięci podręcznej

4.3.3 Parametry przeglądania



Aby zmieniać parametry przeglądania należy kolejno wybrać:

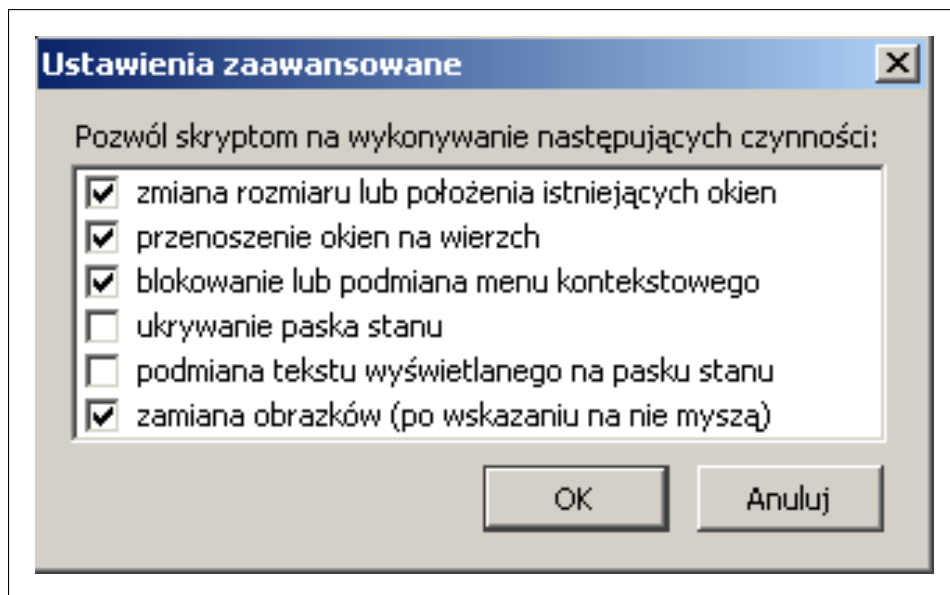
Opcja *Zablokuj wyskakujące okna* jest bardzo istotna, gdyż może ogromnie zwiększyć komfort korzystania z WWW. Zaznaczenie jej powoduje blokowanie mechanizmu, który może być stosowany przez niektóre strony zawierające kod w JavaScript. Kod ten może posłużyć do automatycznego otwarcia wielu okien przeglądarki (tak jak gdybyśmy użyli polecenia *Nowe Okno*) z niekoniecznie interesującą nas zawartością np. reklamową. Warto więc ten mechanizm wyłączyć (czyli zablokować wyskakujące okna)

Jeśli opcja *Zezwól witrynom na instalację oprogramowania* jest włączona pozwala na automatyczną instalację tzw. rozszerzeń do Firefoxa. Nie chcemy, aby na naszym komputerze było instalowane automatycznie jakiegokolwiek oprogramowanie. Stanowi to potencjalne zagrożenie. Instalowane oprogramowanie może np. zawierać wirusy. Tę opcję trzeba więc wyłączyć

Opcja *Pobieraj obrazki* w zasadzie powinna być włączona, gdyż w przeciwnym razie przeglądarka nie będzie wyświetlać grafiki. Dodatkowo włączenie opcji *Pochodzące wyłącznie z serwera z którym nawiązano pierwotnie połączenie* może zmniejszyć ilość wyświetlanych obrazków reklamowych, chociaż z drugiej strony może też zapobiec wyświetlaniu obrazów z użyteczną treścią. Lepiej zamiast tego użyć przycisku *Wyjątki* i dodać do listy nazwy witryn z których nie chcemy wyświetlać obrazków

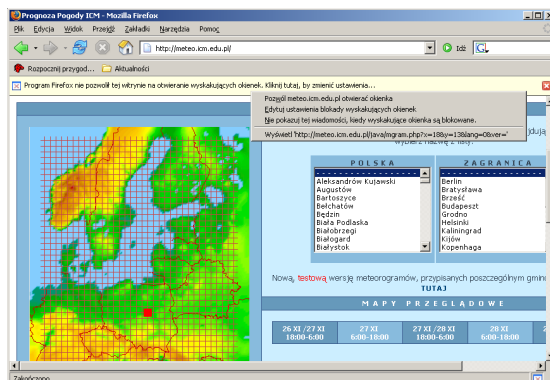
Opcja *Włącz obsługę języka Java*, gdy włączona, umożliwia uruchamianie w przeglądarce omawianych wcześniej apletów Java. Dodatkowo, aby aplety działały, musimy mieć zainstalowane na komputerze środowisko uruchomieniowe Java

Ostatnia opcja *Włącz obsługę języka JavaScript* umożliwia wykonywanie kodu napisanego w języku JavaScript będącego bardzo często składową wielu witryn. Opcja ta musi być włączona, gdyż bez niej duża liczba stron nie będzie wyświetlana poprawnie lub nie będzie posiadać pełnej funkcjonalności. Można natomiast ograniczać pewne mechanizmy JavaScriptu. Wcześniej była mowa o blokowaniu wyskakujących okien, dodatkowo naciśnięcie przycisku *Zaawansowane* wyświetla okno, które umożliwia zablokowanie też innych funkcji JavaScript.



Blokowanie funkcji JavaScript

Informacja o zablokowanych wyskakujących oknach



W przypadku włączonego blokowania wyskakujących okienek, przeglądarka:

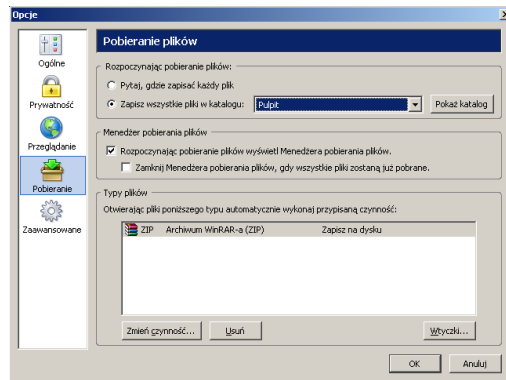
- wyświetli informację o zablokowaniu takiego okienka
- umożliwi dodanie witryny do listy witryn mogących otwierać wyskakujące okienka

W przypadku strony z prognozą pogody, wyskakujące okno zawierałoby prognozę dla Warszawy, czyli treść jak najbardziej pożądaną. Należy więc kliknąć *Pozwól meteo.icm.edu.pl otwierać okienka*. Można oczywiście dodać ręcznie witrynę do listy witryn uprawnionych do wyświetlania nowych okien, korzystając z polecenia

zakładka Przeglądanie ► Opcje ► Uprawnione

□ **Ćwiczenie 4.3.1.** Na przykładzie witryny <http://meteo.icm.edu.pl>, która korzysta z mechanizmu wyskakujących okien, przećwiczyć blokowanie i odblokowywanie wyskakujących okien. □

4.3.4 Pobieranie plików



Aby wskazać katalog do którego będą zapisywane pobierane z sieci pliki należy kolejno wybrać:

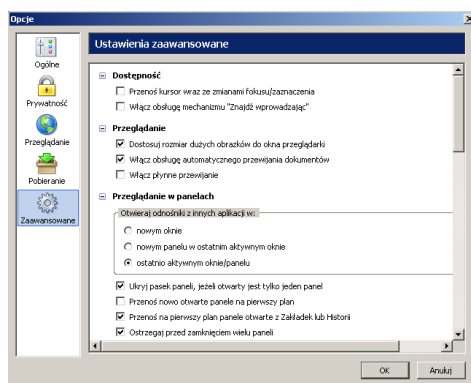
zakładka Pobieranie ► Zapisz wszystkie pliki w katalogu

- użycie do tego celu Pulpiteu nie jest dobrym pomysłem

Aby pobrać plik z sieci, należy kliknąć na odnośnik do pliku na stronie. Często taki link nazywa się np. *pobierz*. Zbiór może wtedy zostać zapisany na dysku w określonym przez nas w konfiguracji katalogu, przeglądarka proponuje również uruchomienie lub otwarcie pliku skojarzoną z danym typem plików aplikacją. Można też zmieniać czynności związane z danymi typami zbiorów. Np. wybrać odpowiednią aplikację, która ma być użyta do otwierania zbioru. Dla pliku *archiwum.zip* jest przypisana czynność *Zapisz na dysku*.

Jeśli nie chcemy, aby pliki były zachowywane zawsze w tym samym, określonym katalogu, należy zaznaczyć opcję *Pytaj, gdzie zapisać każdy plik*. Jest to rozwiązanie lepsze, gdyż można porządkować zbiory na bieżąco – zapisywać je we wcześniej założonej strukturze katalogów.

4.3.5 Ustawienia zaawansowane



Szereg dodatkowych opcji konfiguracyjnych podzielonych na kategorie

- zazwyczaj można bazować na ustawieniach domyślnych (nic nie zmieniać)

Niektóre opcje w zakładce *Zaawansowane* zostaną opisane w części rozdziału poświęconej szyfrowanej wersji protokołu HTTP (HTTPS).

4.3.6 Protokół FTP w przeglądarce internetowej

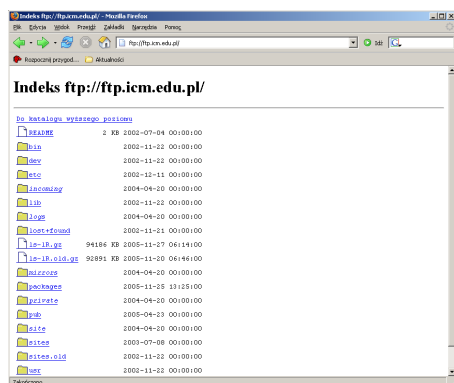
Przeglądarka internetowa służy głównie do pobierania i wyświetlania stron internetowych. Jak już wiemy, do przesyłania stron i związanych z nimi elementów (np. plików) służy protokół HTTP. Istnieje jednak jeszcze protokół FTP, który jest używany wyłącznie do transferu zbiorów. Po połączeniu z serwerem FTP w oknie przeglądarki zobaczymy jedynie listę plików i katalogów.

Serwer HTTP może zostać tak skonfigurowany, aby funkcjonował w podobny jak FTP sposób (przeglądarka używając HTTP również może wyświetlić jedynie listę plików i katalogów). Protokół FTP jest jednak chętnie stosowany także z tego powodu, że umożliwia stosowanie autoryzacji oraz łatwe przenoszenie plików w obie strony (pobieranie na stację roboczą z serwera oraz wysyłanie na serwer).

FTP jest często stosowany w przypadku, gdy chcemy udostępnić w sieci np. oprogramowanie, lub inne dane w postaci plików dyskowych. Zaleta jest oczywista – nie trzeba tworzyć strony w języku HTML. Wystarczy umieścić zbiory na serwerze na którym skonfigurowano protokół FTP. Nie trzeba wykonywać czynności związanych z projektowaniem strony WWW. Serwery FTP są często stosowane do umieszczania w sieci oprogramowania open source. Dla przykładu: ogromny zbiór bezpłatnego oprogramowania jest przechowywany na publicznym serwerze FTP Interdyscyplinarnego Centrum Modelowania Matematycznego i Komputerowego Uniwersytetu Warszawskiego.

<ftp://ftp.icm.edu.pl>

Połączenie z serwerem FTP

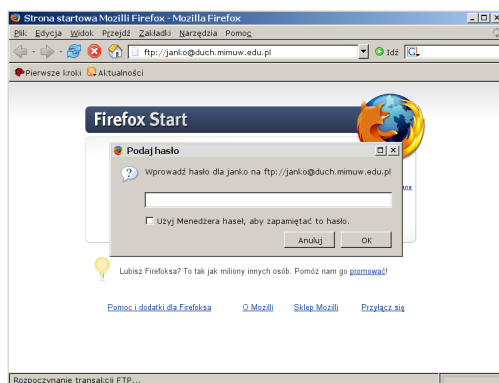


Udostępnione oprogramowanie znajduje się w katalogu *pub*

- jest to standardem na publicznych serwerach FTP

Pobieranie udostępnionych plików za pomocą przeglądarki i protokołu FTP jest analogiczne jak w przypadku HTTP, tylko zamiast kliknięcia odnośnika do zbioru na stronie, w przypadku FTP, należy kliknąć bezpośrednio na nazwę pliku.

Niepubliczne FTP



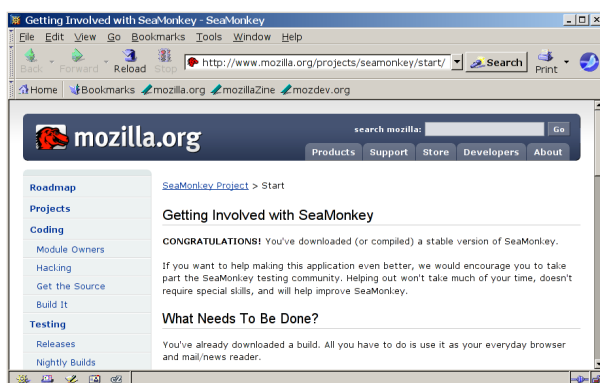
Serwer FTP może wymagać podania nazwy użytkownika i hasła:

- tak ogranicza się dostęp do serwera
- np. tylko dla osób, które mają na nim założone konta

4.4 Inne popularne przeglądarki internetowe

Oprócz opisywanego w niniejszym rozdziale Firefoxa, istnieje jeszcze wiele innych przeglądarek internetowych. Wśród nich niewątpliwie należy wspomnieć o przeglądarce SeaMonkey.

4.4.1 SeaMonkey

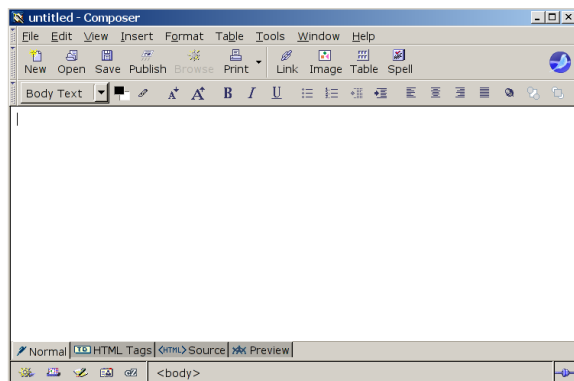


Przeglądarka SeaMonkey zawiera dodatkowe narzędzia

- klient poczty, grup dyskusyjnych oraz edytor HTML

Przeglądarka SeaMonkey jest kontynuacją nierozwijanej już Mozilli. Podobnie jak Mozilla, w porównaniu z Firefoxem, SeaMonkey zawiera szereg dodatkowych narzędzi. SeaMonkey, tak jak Firefox, jest oprogramowaniem open source. Możemy używać SeaMonkey całkowicie za darmo.

4.4.2 SeaMonkey Composer



Aby uruchomić edytor HTML należy kolejno wybrać:

menu Window ►Composer

4.4.3 Microsoft Internet Explorer

W tym miejscu warto krótko wspomnieć o przeglądarce Microsoft Internet Explorer. Jest ona dostarczana razem z systemem operacyjnym Windows, nie oznacza to jednak, że musimy z niej korzystać. Możemy zainstalować

pod Windows dowolną inną przeglądarkę, np. Firefoxa, lub SeaMonkey. Poważną wadą Internet Explorera jest brak jego dostępności dla innych oprócz Windows systemów operacyjnych. Internet Explorer jest oprogramowaniem komercyjnym, ale nie jest sprzedawany jako osobny produkt. Jedynym sposobem, aby go legalnie używać, jest zakup licencji na system operacyjny MS Windows.

Inne omawiane przeglądarki mogą pracować pod wieloma systemami operacyjnymi - np. Windows, Linux, FreeBSD.

4.5 Bezpieczeństwo danych przesyłanych protokołem HTTP

Standardowa implementacja protokołu HTTP za pomocą którego przeglądarki pobierają dokumenty z serwerów WWW, nie posiada żadnych mechanizmów kryptograficznych. Oznacza to, że dane (w tym hasła) przesyłane są przez sieć otwartym tekstem. Dlatego np. witryny internetowe banków zawsze wymuszają stosowanie szyfrowanej wersji HTTP – HTTPS. Jeśli spróbujemy się zalogować na stronie naszego banku, ujrzemy w polu adresu przeglądarki adres URL zaczynający się polem *https://*, a nie *http://*. Jeśli, łącząc się z bankiem nie wpisujemy w polu *adres* adresu w postaci *https://mojbank*, ale *http://mojbank*, to witryna banku zadba o to, aby transmisja odbyła się protokołem HTTPS, a nie HTTP. Adres zostanie zamieniony automatycznie na *https://mojbank*.

4.5.1 Algorytmy szyfrujące stosowane w protokole HTTPS

W podrozdziale tym zostaną omówione podstawy kryptografii pod kątem zastosowania ich w bezpiecznym przesyłaniu zawartości stron WWW. Informacje związane z szyfrowaniem można też znaleźć w rozdziale poświęconym poczcie elektronicznej. Część z tych wiadomości zostanie tu powtórzona, aby umożliwić natychmiastowe zrozumienie opisywanych mechanizmów.

Szyfrowanie symetryczne

Szyfrowanie z użyciem algorytmu symetrycznego ma poważną wadę – musi istnieć niepodważalnie bezpieczny kanał dostarczenia klucza korespondentowi. Można to łatwo zrobić np. spotykając się i przekazując klucz osobiście. Problem występuje, jeśli korespondent jest zbyt oddalony, aby możliwy był osobisty kontakt.

- Polega na użyciu tylko jednego klucza
- Ten sam klucz musi być w posiadaniu nadawcy i odbiorcy
 - nim się szyfruje i deszyfruje wiadomości
- Istnieje poważny problem z przekazywaniem klucza

Szyfrowanie niesymetryczne

Wiadomość zaszyfrowana kluczem prywatnym może być odszyfrowana jedynie odpowiadającym mu kluczem publicznym. I analogicznie, wiadomość zaszyfrowana kluczem publicznym, może być odszyfrowana tylko odpowiadającym mu kluczem prywatnym. Jeśli chcemy zaszyfrować wiadomość do naszego korespondenta, musi on przysłać nam swój klucz publiczny. Klucz nie musi być jednak przekazany w bezpieczny sposób. Jeśli ktoś przechwyci klucz publiczny nie będzie mógł odszyfrować wiadomości nim zaszyfrowanych.

- Polega na użyciu pary kluczy
 - klucz publiczny
 - klucz prywatny
- Należy chronić jedynie klucz prywatny
- Można upublicznić klucz publiczny
- Nie posiada wady algorytmu symetrycznego

Podpisywanie wiadomości

Stosowane tu określenie *podpis elektroniczny* nie jest podpisem elektronicznym zgodnym z obowiązującym prawem. Odpowiednia ustawa mówi, że do składania podpisu, należy używać m.in. karty mikroprocesorowej, na której znajdują się klucze. Więcej informacji można znaleźć np. na <http://www.certum.pl>.

- Korespondenci wymieniają się kluczami publicznymi
- Nadawca podpisuje wiadomość dla odbiorcy swoim kluczem prywatnym
- Odbiorca weryfikuje podpis za pomocą klucza publicznego nadawcy

Podpisywanie wiadomości przebiega w następujący sposób:

- Tworzony jest tzw. skrót wiadomości, która ma być podpisana. Należy wyjaśnić, iż skrót jest generowany w taki sposób, aby był unikalny dla danej wiadomości. Dba o to oprogramowanie szyfrujące
- Skrót jest następnie szyfrowany kluczem prywatnym twórcy wiadomości i załączany do oryginalnej wiadomości
- Wiadomość oraz jej zaszyfrowany skrót (który jest właśnie podpisem elektronicznym) są przesyłane do odbiorcy

Weryfikowanie podpisu odbywa się w następujący sposób:

- Tworzony jest skrót odebranej wiadomości
- Skrót otrzymany od nadawcy jest odszyfrowywany kluczem publicznym nadawcy (odbiorca musi ten klucz posiadać)
- Jeśli dało się odszyfrować skrót przysłany przez nadawcę jego kluczem publicznym, to oznacza, że nadawca zaszyfrował skrót swoim kluczem prywatnym, więc można mieć pewność, że to nadawca jest faktycznym autorem wiadomości. Tylko nadawca ma dostęp do swojego klucza prywatnego, czyli to nadawca złożył podpis
- Jeśli odszyfrowany skrót, jest taki sam jak utworzony skrót, to dodatkowo istnieje pewność, że wiadomość nie została po drodze do odbiorcy zmodyfikowana. Pamiętajmy przecież, że skrót jest unikalny dla danej wiadomości

Szyfrowanie wiadomości

Podpis elektroniczny oprócz tego, że pozwala stwierdzić autentyczność przekazu, dodatkowo zapewnia mechanizm wykrywania potencjalnych modyfikacji wiadomości. Szyfrowanie dodatkowo zapobiega nieautoryzowanemu odczytaniu treści wiadomości.

- Korespondenci wymieniają się kluczami publicznymi
- Nadawca szyfruje wiadomość dla odbiorcy kluczem publicznym odbiorcy
- Tylko odbiorca może ją odczytać, gdyż tylko on ma swój klucz prywatny

Przebieg procesu szyfrowania w protokole HTTPS

- Klient zgłasza chęć użycia szyfrowania serwerowi
- Serwer wysyła klientowi tzw. certyfikat, który zawiera klucz publiczny serwera
- Klient generuje klucz sesji i szyfruje go kluczem publicznym serwera, następnie przesyła do serwera
- Dalszy ciąg komunikacji odbywa się przez kanał szyfrowany kluczem sesji (ale już z użyciem algorytmu symetrycznego, jest on szybszy od asymetrycznego)

Problem ustalenia autentyczności serwera

- Serwer A przesyła klientowi (przeglądarcie) certyfikat zawierający klucz publiczny
- Skąd wiemy, że to jest certyfikat *prawdziwego* serwera ?

- Przeglądarka potrafi sprawdzić autentyczność certyfikatu serwera
- Używa do tego mechanizmu podpisu elektronicznego

Można sobie wyobrazić następującą sytuację: Ktoś podszył się pod serwer banku. Użytkownicy nie łączą się już z prawdziwym serwerem banku, tylko z serwerem imitującym. Serwer imitujący zbiera wszelkie treści (np. identyfikatory i hasła) przekazywane przez użytkowników, którzy myślą, że wyświetlili stronę WWW banku. Informacje zebrane w ten sposób mogą posłużyć osobie, która uruchomiła „podstawiony” serwer do użycia na prawdziwej stronie WWW banku.

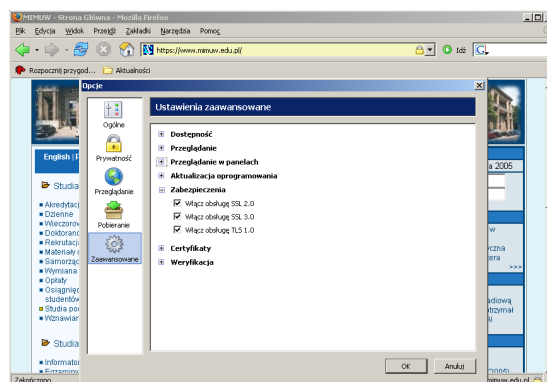
Istnieje jednak mechanizm, który pozwala na sprawdzenie autentyczności certyfikatu. Certyfikaty są podpisywane, a przeglądarka jest w stanie zweryfikować podpis na certyfikacie serwera.

Certyfikat serwera zostaje podpisany kluczem prywatnym organizacji CA (ang. Certification Authority). CA jest to jednostka, która, zanim podpisze certyfikat serwera sprawdza tożsamość osób za niego odpowiadających.

Przeglądarka może zweryfikować podpis na certyfikacie serwera za pomocą klucza publicznego organizacji CA. Cały ten proces przebiega w sposób zgodny z opisanym wcześniej mechanizmem podpisu elektronicznego. Przeglądarka posiada klucze publiczne wielu zaufanych CA (a dokładnie certyfikaty CA zawierające klucze publiczne CA). Jeśli przeglądarka otrzyma certyfikat i nie może zweryfikować podpisu na nim, wyświetla stosowne ostrzeżenie

4.5.2 Używanie protokołu HTTPS w przeglądarce Firefox

Włączanie protokołu HTTPS w przeglądarce

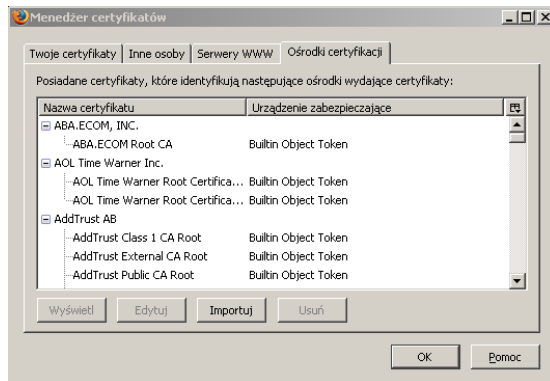


Aby korzystać z protokołu HTTPS należy włączyć w przeglądarce obsługę SSL:

menu Narzędzia ► Opcje ► Zaawansowane ► Zabezpieczenia

HTTPS korzysta z mechanizmu SSL/TLS. SSL (ang. Secure Socket Layer) implementuje algorytmy kryptografii opisane wcześniej. TLS (ang. Transport Layer Security) jest następcą SSL, jednak nie wszystkie witryny obsługują TLS. W przeglądarce Firefox SSL/TLS jest włączony standardowo, więc nie musimy nic modyfikować. Gdy uzyskaliśmy połączenie szyfrowane z jakąś witryną, to pasek adresu ma kolor żółty i jest wyświetlany symbol kłódki.

Certyfikaty znanych CA

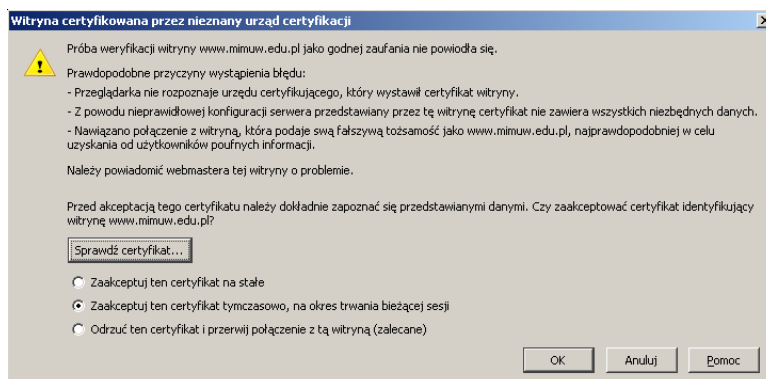


Dostęp do certyfikatów znanych CA jest możliwy za pomocą menadżera certyfikatów:

menu Narzędzia ► Opcje ► Zaawansowane ►

Menadżer certyfikatów ► zakładka Ośrodki certyfikacji

Problemy ze sprawdzeniem podpisu



Przeglądarka otrzymała od serwera certyfikat

- certyfikat jest podpisany kluczem prywatnym nieznanego przeglądarce CA
- przeglądarka wyświetla ostrzeżenie

W przypadku strony <https://www.mimuw.edu.pl> na serwerze jest używany certyfikat podpisany kluczem prywatnym utworzonym na Wydziale MIMUW. Tym samym Wydział MIMUW stworzył własne CA do podpisywania certyfikatów. Klucz publiczny tego CA (zawarty w certyfikacie CA MIMUW) można pobrać ze strony

<http://www.mimuw.edu.pl/pomoc/CAcert.pem>

i dodać do certyfikatów CA przeglądarki, która będzie wtedy mogła weryfikować podpis na certyfikacie serwera WWW MIMUW i ostrzeżenie się nie pojawi.

Takie CA jednak nie cieszy się powszechnym zaufaniem, ponieważ przeglądarki standardowo nie posiadają klucza publicznego CA MIMUW. Dlatego, jeśli certyfikat CA MIMUW nie został dodany, w przeglądarce zawsze pojawi się ostrzeżenie.

W tym przypadku nie została podjęta decyzja o podpisie certyfikatu serwera <https://www.mimuw.edu.pl> za pomocą klucza prywatnego zaufanego CA (np. www.certum.pl) ze względów ekonomicznych. W przypadku instytucji np. finansowych certyfikaty *zawsze* są podpisywane przez znane CA i przedstawione ostrzeżenie nigdy nie może się pojawić. Gdyby jednak tak się stało, to mamy powody przypuszczać, że ktoś podszywa się pod serwer np. banku. Należy wtedy natychmiast przerwać połączenie z witryną (opcja *Odrzuć ten certyfikat i przerwij połączenie z tą witryną*) i skontaktować się z bankiem podając treść ostrzeżenia.

W przypadku strony <https://www.mimuw.edu.pl> moglibyśmy po prostu zaakceptować certyfikat, ponieważ nie ponosimy tu wielkiego ryzyka nawet w przypadku, gdyby serwer nie okazał się autentyczny. Trzeba by więc zaznaczyć opcję *Zaakceptuj ten certyfikat na stałe*. Jeśli certyfikat zostanie zaakceptowany, będzie on zapisany na dysku naszego komputera i uznawany przez przeglądarkę za zaufany. Mimo tego, że nie jest odpowiednio podpisany, okienko z ostrzeżeniem nie będzie się pojawiać. Pamiętajmy, że podczas połączenia z bankiem, ostrzeżenie i pytanie o chęć zaakceptowania certyfikatu w ogóle nie może się ukazać. Dla pewności, najlepiej nigdy nie korzystać z opcji *Zaakceptuj ten certyfikat na stałe*. Jeśli już musimy połączyć się ze stroną, w przypadku której przeglądarka sygnalizuje problem z certyfikatem, ale stronie tej ufamy, lepiej użyć opcji *Zaakceptuj ten certyfikat tymczasowo, na okres trwania bieżącej sesji*. Wtedy przy ponownym połączeniu, znów zostaniemy poinformowani o kłopotach z certyfikatem.

W przypadku witryny <https://www.mimuw.edu.pl>, lepszym sposobem od akceptowania certyfikatu jest wspomniane wcześniej pobranie certyfikatu CA MIMUW ze strony i dodanie go do znanych przeglądarce certyfikatów CA. Przeglądarka uzna certyfikat MIMUW za poprawny, bo będzie miała

dostęp do klucza publicznego, który jest zawarty w certyfikacie CA MIMUW i będzie mogła sprawdzić podpis na certyfikacie MIMUW za pomocą tego klucza. Jest to dużo bezpieczniejsze, niż kliknięcie *Zaakceptuj ten certyfikat na stałe*, bo opiera się na bardziej świadomym działaniu. Oczywiście w przypadku połączeń z bankiem nigdy nie możemy i nie powinniśmy być proszeni o wykonanie opisanych kroków. Jeśli jest inaczej – niech wzbudzi to nasze podejrzenia!

Zawsze przerywajmy połączenie i kontaktujmy się z bankiem, gdy przeglądarka wyświetli nam *jakikolwiek* komunikat związany z certyfikatami. Pod żadnym pozorem nie używajmy wtedy opcji *Zaakceptuj ten certyfikat na stałe*, ani też *Zaakceptuj ten certyfikat tymczasowo, na okres trwania bieżącej sesji*.

W tym miejscu należy jeszcze w dokładniej wyjaśnić znaczenie słowa certyfikat, jak już wiemy, najważniejszą informacją jaką zawiera certyfikat, jest klucz publiczny serwera, bądź też klucz publiczny organizacji CA, oraz podpis. Oprócz ww. danych, certyfikat zawiera jeszcze sporo dodatkowych informacji.

- Nazwę certyfikatu (musi być taka jak nazwa serwera)
- Nazwę organizacji do której należy certyfikat
- Nazwę jednostki organizacyjnej
- Okres ważności certyfikatu
- Dane o wystawcy certyfikatu (czyli o CA, które złożyło podpis na certyfikacie)
- Klucz publiczny serwera
- Podpis wszystkich ww. informacji dokonany kluczem prywatnym CA

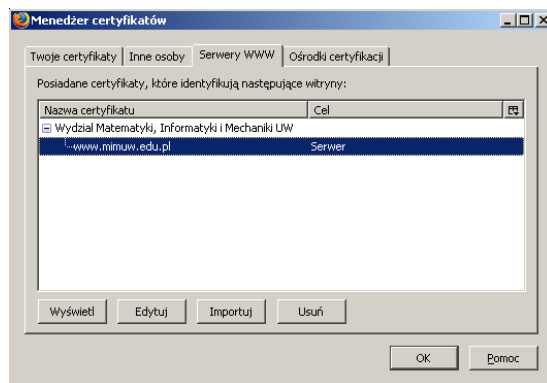
W przypadku organizacji CA, certyfikat CA jest podpisany kluczem prywatnym tego samego CA. Jest to uzasadnione, ponieważ z założenia CA powinno być godne zaufania. Nie oznacza to, że faktycznie musimy ufać każdemu CA. Jeśli jednak ufamy jakiemuś CA, a jego certyfikatu nie ma na liście przeglądarki, możemy go dodać używając menedżera certyfikatów, zakładki *Ośrodki Certyfikacji* i klikając przycisk *importuj*. Wcześniej trzeba oczywiście posiadać dany certyfikat CA, w przypadku opisywanego CA MIMUW można go pobrać ze strony.

Wyświetlanie szczegółów certyfikatu

- Szczegóły dt. certyfikatu można wyświetlić bezpośrednio z okienka ostrzeżenia klikając na *Sprawdź certyfikat*
- Bardziej dokładne informacje kryją się pod zakładką *Szczegóły*
- Certyfikatami, które zostały zaakceptowane można zarządzać za pomocą menadżera certyfikatów

W przypadku użycia opcji *Zaakceptuj ten certyfikat na stałe* korzystalibyśmy z fałszywego serwera nawet wielokrotnie, nic o tym nie wiedząc, bo przeglądarka nie wyświetlałaby już ostrzeżeń przy kolejnych połączeniach!

Zarządzanie certyfikatami



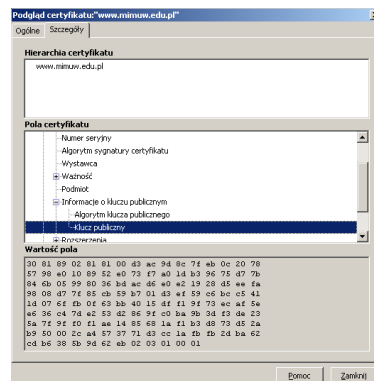
Aby edytować zapamiętane wcześniej certyfikaty należy uruchomić menadżera certyfikatów:

menu Narzędzia ► Opcje ► Zaawansowane ►

Menadżer certyfikatów ► zakładka Serwery WWW

Wyświetlenie klucza publicznego zawartego w certyfikacie

W opisany sposób można przeglądać pola związane z danym certyfikatem. Jedno z tych pól zawiera informacje o kluczu publicznym.



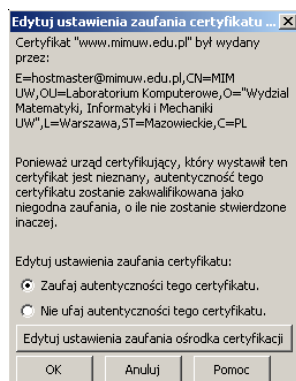
Aby wyświetlić wartość klucza publicznego certyfikatu, należy wybrać kolejno:

zakładka Serwery WWW ► dany certyfikat ► Wyświetl ►

zakładka Szczegóły ► Informacje o kluczu publicznym ► Klucz Publiczny

Edycja certyfikatu

Za pomocą mechanizmu edycji certyfikatów można zmienić poziom zaufania związany z danym certyfikatem.



Menedżer certyfikatów umożliwia edycję certyfikatów:

menu Narzędzia ► Opcje ► Zaawansowane ►

Menadżer certyfikatów ► zakładka Serwery WWW ► Edytuj

4.6 Wyszukiwarki internetowe

Wyszukiwarka internetowa jest aplikacją, która posiada interfejs dostępny przez przeglądarkę internetową i umożliwia wyszukiwanie dokumentów umieszczonych w sieci Internet – głównie (choć nie tylko) na stronach WWW. W praktyce, z wyszukiwarek internetowych korzysta się bardzo często, dlatego też w niniejszym rozdziale zostaną omówione zasady ich działania. Będą też podane informacje dotyczące korzystania z wyszukiwarek.

4.6.1 Zasada działania wyszukiwarki internetowej

Ze względu na dynamiczny rozwój Internetu, okazało się, że istnieje potrzeba stworzenia oprogramowania, które będzie służyło do automatycznego tworzenia zbioru informacji, opartego o treść analizowanych stron, tak aby zbiór ten mógł być później efektywnie przeszukiwany. Okazało się, że inne – nie w pełni automatyczne metody katalogowania Internetu, nie mają szans powodzenia.

- Działanie wyszukiwarki związane jest z modułami z których składa się jej oprogramowanie:

- Robot (ang. crawler) pobiera strony z sieci
- Analizator treści dokonuje analizy pobranych stron, następnie tworzony jest zapis do bazy danych związany z treścią strony
- Interfejs do wyszukiwania – dostępny za pomocą przeglądarki internetowej służy do wprowadzania zapytań oraz wyświetlania wyników

Zasada działania wyszukiwarki jest następująca.

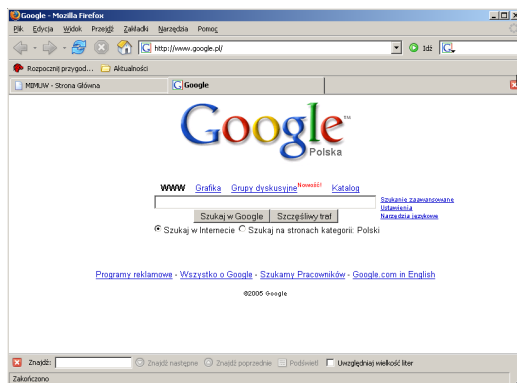
Robot pobiera dokumenty z serwerów dostępnych w sieci (stosuje się techniki umożliwiające dotarcie do jak największej liczby stron). Robot jest swoistym rodzajem automatycznej przeglądarki internetowej, która sama potrafi podążać za odnośnikami.

W praktyce robotom udaje się to całkiem dobrze!

Analizator dokonuje analizy treści pobranej strony, następnie pewne informacje dotyczące strony zostają zapisane do bazy danych (np. występowanie wyrazów i innych elementów – zależy to od algorytmu użytego w konkretnej wyszukiwarce). Wykonywanie tego procesu bywa nazywane indeksowaniem stron WWW. Indeksowanie umożliwia późniejsze szybkie wyszukiwanie słów kluczowych i fraz bez konieczności ponownego analizowania i przeszukiwania źródłowych dokumentów. Jeśli zapytamy o jakąś informację i informacja ta znajduje się na danej stronie, to wynikiem wyszukiwania będzie odnośnik to tej właśnie strony.

Aby wyszukać interesującą nas treść, należy podać związany z nią wyraz bądź frazę.

4.6.2 Podstawy obsługi wyszukiwarki internetowej



Korzystanie z wyszukiwarki jest proste:

- W przeglądarce należy wpisać adres wyszukiwarki: www.google.com
- Można teraz podać słowo lub frazę i kliknąć *Wyszukaj w Google*

Wynikiem wyszukiwania będą odnośniki do stron WWW, które według wyszukiwarki najlepiej odpowiadają wyszukiwanej treści i w tej też kolejności

będą posortowane. Przy każdej z pozycji listy pojawia się również odnośnik *Kopia*, wiąże się z tym ważna informacja: wyszukiwarka Google oprócz samego tylko indeksowania, przechowuje także kopie stron internetowych!

Pamiętajmy więc, że jeśli publikujemy jakąś treść na stronie, a następnie tą treść usuniemy, to może się okazać że jest ona nadal dostępna dla osób korzystających z wyszukiwarki Google. Klikając na odnośnik *Kopia* możemy się przekonać, że strona rzeczywiście jest dostępna na serwerach wyszukiwarki, a nie tylko na właściwym serwerze, do którego kieruje główny odnośnik. Niewątpliwie może nas także interesować takie zagadnienie: jak to jest możliwe, że wyszukiwarka może przechowywać treść tak ogromnej liczby stron internetowych?

Wyjaśniając to należy zauważyć, że Google korzysta z architektury rozproszonej, a nie jakiegoś np. jednego centralnego serwera. W związku z tym jest możliwe utrzymywanie ogromnych ilości danych i realizowanie wydajnego dostępu do nich. W skład rozproszonej architektury wchodzi wiele serwerów umieszczonych w różnych krajach, działających pod kontrolą systemu operacyjnego Linux. Dokładna ilość maszyn jest utrzymywana przez firmę Google w tajemnicy.

Google indeksuje ponad 8 miliardów stron!

Zamiast przycisku *Wyszukaj w Google* możemy kliknąć *Szczęśliwy traf*. Zostanie od razu wyświetlona strona, do której odnośnik pojawiłby się na pierwszym miejscu po kliknięciu *Wyszukaj w Google*. Nie musi to być dokładnie ta strona, którą chcemy obejrzeć.

Należy wspomnieć, iż wyszukiwarki są oprogramowaniem komercyjnym i przynoszą ich twórcom (przynajmniej niektóre) duże zyski. Wyszukiwarka jest przecież świetnym miejscem do umieszczania reklam, bowiem możemy uzależniać wyświetlane treści od zapytania, które sformułuje użytkownik. Oznacza to, że można mieć pewność, że pokazywana reklama jest właściwie adresowana. Wyniki wyszukiwania będące płatnymi ogłoszeniami są jednak zazwyczaj oddzielone od pozostałych odnośników. Np. w przypadku Google są to tzw. *Linki sponsorowane*.

4.6.3 Zaawansowane wyszukiwanie

Aby wyszukać treść, która nas rzeczywiście interesuje, należy w odpowiedni sposób określić frazę pytającą, która zostanie użyta. Poniższe wiadomości pozwolą dowiedzieć się w jaki sposób budować właściwe zapytania. Informacje dotyczące wyszukiwania, stosowania operatorów itd. będą dotyczyły wyszukiwarki Google, w innych wyszukiwarkach obsługa opisanych funkcji może być inna, lub może nie występować.

Dobór właściwych zapytań

- Dobór właściwych haseł ma zasadnicze znaczenie dla uzyskania pozytywnych wyników wyszukiwania
- Często lepiej zastosować frazę niż pojedyncze wyrazy

- np. szukamy informacji o wakacjach w górach
- można użyć frazy *wczasy Tatry*
- albo *tani hotel Tatry*
- Google wyszuka strony, które zawierają wszystkie wyrazy z których składa się zapytanie
 - domyślnie jest stosowany operator AND
- Wielkość liter nie ma znaczenia
- Można wyszukać strony zawierające którykolwiek z wyrazów podanych w zapytaniu
 - należy użyć operatora OR
 - np. *tani hotel Tatry OR Mazury*

Uwzględnienie często używanych słów

- Jeśli wpiszemy do okienka wyszukiwania jakąś frazę
 - Google przeprowadzi wyszukiwanie
 - poda odnośniki do stron
 - słowa z których składa się fraza mogą wystąpić w dowolnej kolejności
- Jeśli chcemy wyszukać strony zawierające dokładną frazę
 - układamy słowa w zadanej kolejności
 - należy umieścić ją w cudzysłowach
- Standardowo wyszukiwarka pomija w zapytaniach
 - często używane słowa
 - niektóre litery i cyfry
 - użytkownik jest o tym informowany stosownym komunikatem
- Gdy użycie takiego słowa lub cyfry jest konieczne
 - możemy użyć operatora „+”
 - wtedy nie zostanie ono pominięte przy wyszukiwaniu
- Przykład:
 - *Wojna światowa +I*
 - jeśli nie użylibyśmy +I, to otrzymalibyśmy również odnośniki dotyczące II Wojny.

Wyszukiwanie dokładnych fraz przydaje się szczególnie przy szukaniu np. tytułów książek, utworów, nazw własnych.

Pomijanie przez wyszukiwarkę często stosowanych słów wynika z tego, iż najczęściej ich obecność w zapytaniach nie poprawia wyników, natomiast spowalnia proces wyszukiwania.

Zawężanie zakresu wyszukiwania

- Użycie operatora „-”
 - pozwala wykluczyć w wynikach wyszukiwania pewne strony
- Przykład
 - chcemy dowiedzieć się czegoś o telefonach komórkowych
 - nie chcemy kupić telefonu
 - można użyć frazy *telefon komórkowy -sklep*
- Wynik wyszukiwania
 - zostaną pominięte odnośniki do stron które być może są sklepami internetowymi

Operator „-” jest również użyteczny w przypadku, gdy poszukujemy słowa, które ma więcej niż jedno znaczenie. Używając znaku „-” można pominąć znaczenie na którym nam nie zależy.

Wyszukiwanie typów plików w określonych witrynach

- Operator *site:*
 - pozwala zawęzić wyszukiwanie do konkretnej witryny
- Przykład
 - chcemy wyszukać informacje dotyczące geometrii różniczkowej
 - ale tylko na stronie Wydziału Matematyki UW
 - możemy użyć frazy: *geometria różniczkowa site:mimuw.edu.pl*
- Operator *filetype:rozszerzenie*
 - możliwość szukania określonych typów plików
 - rozszerzenie określa faktyczny typ pliku
 - * np. PDF
- Przykład
 - używamy frazy: *umowa kupna sprzedaży filetype:pdf*
 - wynikiem są odnośniki do plików pdf

Oprócz wymienionych, istnieją jeszcze inne operatory umożliwiające precyzowanie zapytań:

- *link:* - powoduje zwrócenie w wyniku stron zawierających linki do danej strony np. można użyć w zapytaniu *link:www.mimuw.edu.pl*

- *intitle:* - ogranicza wynik wyszukiwania do stron, które zawierają słowo umieszczone po dwukropku tylko w tytule strony
- *inurl:* - ogranicza wynik wyszukiwania do stron, które zawierają słowo umieszczone po dwukropku tylko w adresie strony
- *intext:* - ogranicza wynik wyszukiwania do stron, które zawierają słowo umieszczone po dwukropku tylko w tekście strony

Można też stosować operatory *allintitle:*, *allinurl:*, *allintext:*, różnica jest taka, iż po dwukropku możemy podać listę słów – frazę, operator będzie się odnosił do całej frazy. W poprzednim przypadku operator np. *intitle:* dotyczył tylko słowa występującego bezpośrednio po dwukropku. Jeśli dalej pojawiłyby się jakieś wyrazy to wobec nich, wyszukiwanie nie byłoby ograniczane np. do występowania w tytule.

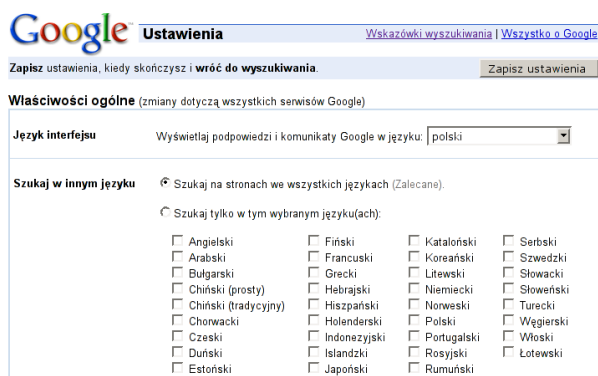
4.6.4 Szukanie zaawansowane

Do precyzyjnego wyszukiwania można również użyć:

- główne okno wyszukiwarki Google, odnośnik *Szukanie zaawansowane*
- zastępuje korzystanie z opisanych wyżej operatorów

Używanie operatorów bezpośrednio w zapytaniach jest dużo szybsze, niż korzystanie z odnośnika *Szukanie zaawansowane*. Warto się więc ich nauczyć.

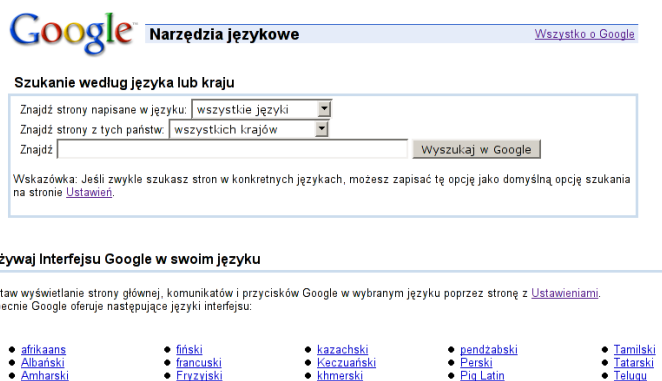
4.6.5 Ustawienia wyszukiwania



Istnieje możliwość zapamiętania niektórych ustawień wyszukiwarki w przeglądarce:

- główne okno wyszukiwarki Google, odnośnik *Ustawienia*
- wykorzystanie mechanizmu ciasteczek
 - można ustawić, aby były wyszukiwane strony w określonym języku
 - domyślnie Google szuka stron utworzonych we wszystkich językach

4.6.6 Narzędzia językowe



Można wybrać język wyszukiwanych stron

- główne okno wyszukiwarki Google, odnośnik *Narzędzia językowe*

Google, oprócz wyszukiwania odnośników do stron WWW, umożliwia też wyszukiwanie grafiki, wyszukiwanie w grupach dyskusyjnych, a także udostępnia katalog w którym znajdują się odnośniki podzielone na kategorie. Aby wyszukiwać np. pliki graficzne, należy kliknąć odnośnik *Grafika* w głównym oknie wyszukiwarki.

Google jest wyszukiwarką, która daje bardzo dobre wyniki wyszukiwania, ale oczywiście nie jest jedyną. Inne o których warto wspomnieć to www.yahoo.com, www.altavista.com, www.netsprint.pl. Ta ostatnia umożliwia wyszukiwanie jedynie wśród polskich zasobów Internetu.

4.6.7 Problemy z używaniem wyszukiwarek internetowych

Podstawowy problem związany z korzystaniem z wyszukiwarki polega przede wszystkim na braku umiejętności właściwego określenia kryteriów wyszukiwania. Jeśli jednak znamy zasady związane z korzystaniem z wyszukiwarki, zwykle dość łatwo możemy osiągnąć zadowalające wyniki wyszukiwania. Warto w tym miejscu wspomnieć również o problemach innej natury – związanych z wiarygodnością źródeł z których czerpiemy za pomocą wyszukiwarki informacje. Należy zauważyć, iż sieć Web składa się z ogromnej ilości stron internetowych, autorstwa osób prywatnych, oraz osób reprezentujących firmy, instytucje państwowe, korporacje, szkoły, uniwersytety itp. W treści stron WWW często pojawiają się błędy i fałszywe informacje, niektóre są wynikiem braku wiedzy lub pomyłki autora, inne mogą być efektem zamierzonego działania. Pamiętajmy, aby stosownie weryfikować wszelkie dane czerpane z Internetu, w szczególności te, o których autorze nic nie wiemy, lub nie możemy określić jego wiedzy oraz rzetelności.

Pamiętajmy również o prawach autorskich, które mogą mieć zastosowanie w odniesieniu do dokumentów pobieranych przez nas z sieci. Na te zagadnienia trzeba szczególnie zwrócić uwagę, w momencie, gdy chcemy wykorzystać treści pobrane ze stron WWW we własnych pracach. W razie wątpliwości, próbujemy skontaktować się z autorem.

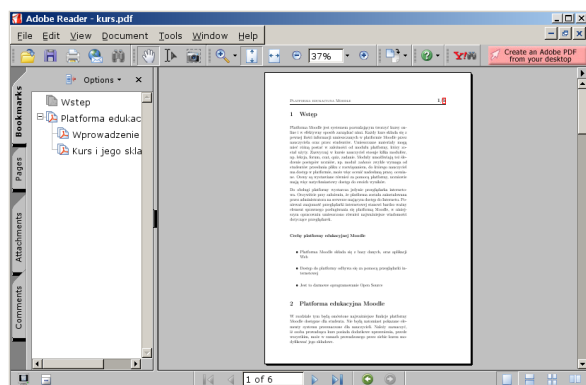
4.7 Pobieranie dokumentów i konwersja formatów

Jak już wiemy, przeglądarka internetowa wyświetla strony sieci Web oraz może służyć także do pobierania plików z sieci. Warto wspomnieć o bardzo popularnym formacie publikowania dokumentów – PDF (ang. Portable Document Format), który został opracowany przez firmę Adobe. Wiele zbiorów z którymi się spotkamy w Internecie będzie właśnie w tym formacie. PDF stał się standardem, jeśli chodzi o publikowanie dokumentów na stronach WWW. W tym rozdziale zostaną omówione narzędzia służące do odczytu i tworzenia plików PDF.

Format PDF jest chętnie stosowany, gdyż ma tę własność, iż wydruk wygląda dokładnie tak samo, jak dokument ukazany na ekranie monitora. Mamy również pewność, że osoba, której przekazaliśmy plik PDF, oglądając go na monitorze, będzie widziała dokument z zachowaniem krojów pisma i położenia elementów graficznych, czyli dokładnie tak, jak chcemy, aby dokument był odbierany. W przypadku plików np. DOC, wygląd może zależeć od ustawień oprogramowania. Ktoś, kto przegląda nasz dokument w formacie DOC, może widzieć efekt inny, niż oglądany przez nas.

W przeciwieństwie do formatów tworzonych oprogramowaniem MS Office (DOC, XLS, PPT itd.), specyfikacja PDF jest otwarta i powszechnie znana – to kolejna zaleta PDF. Otwartość standardu gwarantuje, iż mamy możliwość sprawdzenia, czy w utworzonym dokumencie nie znajdują się jakieś informacje, których tam być nie powinno. Nie jesteśmy też uzależnieni od oprogramowania konkretnej firmy. Jeszcze jedna zaleta warta wymienienia: PDF umożliwia stosowanie elementów hipertekstu – odnośników, ułatwia to przeglądanie dokumentu. Więcej informacji o hipertekście w PDF można znaleźć w rozdziale poświęconym językowi HTML. Stosujemy więc format PDF do publikacji naszych dokumentów!

4.7.1 Odczyt dokumentów formatu PDF

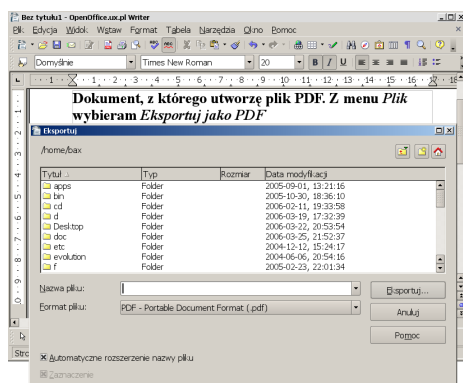


Aby odczytać dokumenty PDF

- możliwe za pomocą programu Adobe Acrobat Reader
- program do pobrania za darmo ze strony <http://www.adobe.com>

Obsługa programu Acrobat Reader jest bardzo łatwa i nie wymaga dodatkowych wyjaśnień. Program występuje w wersjach dla wielu systemów operacyjnych. Jeśli więc opublikujemy dokument w formacie PDF, mamy pewność, że będą go mogli przeczytać użytkownicy różnych systemów.

4.7.2 Tworzenie dokumentów formatu PDF



Aby utworzyć dokument PDF można:

- skorzystać z oprogramowania Adobe Acrobat; lub
- skorzystać z oprogramowania Open Office Writer

menu Plik ► opcja Eksportuj jako PDF

Oprogramowanie Adobe Acrobat (w odróżnieniu od Adobe Acrobat Readera) jest płatne, ale nie jest jedyną drogą do utworzenia dokumentu PDF. W tym celu możemy także użyć bezpłatnego pakietu oprogramowania biurowego Open Office. Open Office można pobrać z sieci:

<ftp://ftp.icm.edu.pl/packages/OpenOffice>

a każdy z modułów Open Office (edytor tekstu – Writer, arkusz kalkulacyjny – Calc, narzędzie do tworzenia prezentacji – Impress) umożliwi utworzenie pliku PDF. Wystarczy więc utworzyć dokument np. w edytorze tekstu, następnie eksportować go w sposób pokazany wyżej do formatu PDF.

OpenOffice radzi sobie również z obsługą formatów MS Office.

Możemy zatem utworzyć plik PDF z dokumentu w formacie DOC, otwierając go za pomocą OpenOffice i eksportując do PDF-a w sposób opisany wyżej. Jak widać OpenOffice jest uniwersalnym narzędziem, i niewątpliwie należy z niego korzystać.

Kolejna metoda tworzenia zbiorów PDF polega na użyciu oprogramowania (głównie dla systemu Windows), które instaluje się jako wirtualna drukarka. Możemy więc otrzymać dokument PDF, drukując np. tekst z edytora za pomocą tej drukarki. Przykładem takiego programu jest np. PDF995, można go pobrać ze strony <http://www.pdf995.com> i używać za darmo (darmowa wersja wyświetla reklamę, jeśli chcemy się jej pozbyć, trzeba dokonać opłaty i zarejestrować program). Inne oprogramowanie do tworzenia PDF to np. darmowe Free EasyPDF, PrimoPDF.

Czasem są z tym problemy, ale pamiętajmy, że Microsoft nie opublikował specyfikacji dokumentów DOC, XLS itd. Są to standardy zamknięte.

4.7.3 Konwersja formatów graficznych

Często istnieje potrzeba konwersji dokumentów graficznych z jednego formatu graficznego, do innego. Do tego celu świetnie nadaje się oprogramowanie GIMP. Wersję dla Windows można pobrać za darmo z: <http://gimp-win.sourceforge.net>. Aby dokonać konwersji formatów, należy otworzyć dany plik graficzny, a następnie zapisać, wybierając odpowiedni (np. inny) format. Program GIMP może służyć nie tylko do konwersji formatów graficznych – jest to bardzo rozbudowane środowisko do tworzenia i obróbki grafiki rastrowej, funkcjonalnie podobne do Adobe Photoshop.

Rozdział 5

Poczta elektroniczna

5.1 Wprowadzenie

Poczta elektroniczna (ang. e-mail, electronic mail) to jedna z najbardziej popularnych i najszerzej wykorzystywanych obecnie usług internetowych. Służy do przesyłania wiadomości głównie tekstowych tzw. listów elektronicznych.

Stały wzrost popularności poczty elektronicznej jest połączony z rozwojem usług przesyłu danych i związanego z nimi masowego dostępu do sieci Internet. Poczta elektroniczna posiada szereg zalet w porównaniu z pocztą tradycyjną. Niestety istnieją też wady tej metody komunikacji.

Niniejszy moduł pozwoli *aktywnie studiującej* osobie osiąść umiejętności pozwalające na:

- korzystanie z poczty elektronicznej
- konfigurację programów pocztowych
- rozumienie zasad działania poczty elektronicznej

5.1.1 Zalety poczty elektronicznej

- Duża szybkość działania
 - poczta dochodzi do adresata w ciągu sekund/minut
- Mały koszt
 - przy użyciu stałego łącza ponosi się jedynie opłatę abonamentową
- Prosta forma adresów
- Możliwość przesyłania danych dowolnego typu
 - tekstu, grafiki, dźwięku itp.

- Wygoda użytkowania
 - nie trzeba iść na pocztę

5.1.2 Wady poczty elektronicznej

- Brak mechanizmu potwierdzenia doręczenia
- Poczta elektroniczna ułatwia proces rozprzestrzeniania się wirusów komputerowych
 - niestety
- Poczta elektroniczna bywa źródłem również niechcianych wiadomości
 - spamu
- Z używaniem poczty elektronicznej często wiążą się konieczności wykorzystania oprogramowania antyspamowego i antywirusowego
 - potencjalne dodatkowe koszty

5.1.3 Bardzo krótka historia poczty elektronicznej

Poczta elektroniczna powstała w roku 1965. W tym czasie jednak służyła tylko do przesyłania wiadomości między użytkownikami tej samej maszyny, a adres e-mailowy w postaci znanej dzisiaj jeszcze nie był stosowany. Usługa polegająca na wysyłaniu wiadomości przez sieć (między komputerami) powstała w roku 1971, wtedy też użyto znaku @ do odseparowania nazwy użytkownika od nazwy maszyny lub nazwy domeny internetowej. Na początku, poczta elektroniczna bazowała na protokole CPYNET, później wykorzystywano np. FTP i kilka innych protokołów. Dopiero na początku lat 80 powstał specjalnie do tego celu skonstruowany protokół SMTP. Wszystkie definicje związane z pocztą elektroniczną (wymagania dotyczące formatów wiadomości, opisy protokołów itd.) są zawarte w tzw. dokumentach RFC. Dokumenty RFC (Request for Comments) opisują większość standardów i protokołów używanych w sieciach komputerowych. Więcej informacji można znaleźć na:

<http://www.rfc-editor.org>

5.1.4 Mechanizmy funkcjonowania poczty elektronicznej

Poczta elektroniczna korzysta z sieci Internet (protokołu TCP/IP) jako medium transportowego. Do transmisji wykorzystywane są obecnie trzy protokoły: SMTP do wysyłania i do przesyłania poczty między serwerami, oraz POPv3 i IMAP do tzw. „ściągnięcia” (pobierania) wiadomości, czyli przesyłania z serwera na stację klienta. Proces tworzenia i wysyłania listu elektronicznego przebiega następująco: nadawca posiada program pocztowy tzw. Mail User Agent (MUA) zainstalowany na swojej stacji roboczej i

za pomocą tego programu tworzy wiadomość. Program ten następnie łączy się z serwerem poczty za pomocą protokołu SMTP, przesyła wiadomość do serwera, w dalszej kolejności serwer wysyła wiadomość dalej. Poczta dochodzi na maszynę, która jest właściwym serwerem dla danego adresata. Adresat (czyli odbiorca) wiadomości pobiera pocztę na swój komputer za pomocą programu MUA i czyta treść również za pomocą MUA. Oczywiście musi on posiadać konto pocztowe (skrzynkę pocztową) na serwerze identyfikowane przez nazwę użytkownika oraz hasło. Zanim program kliencki (MUA) zacznie transfer poczty z serwera, musi dokonać autoryzacji przesyłając do maszyny serwerowej nazwę użytkownika i hasło, te parametry identyfikują konkretne konto.

Do odbierania poczty z serwera, autoryzacja jest zawsze wymagana, natomiast w przypadku wysyłania jest inaczej — w ramach np. sieci wydziałowej użytkownicy mogą wysyłać pocztę przez serwer bez procesu uwierzytelnienia.

Do odbierania poczty MUA używa protokołu POP3 lub IMAP. Przykładem oprogramowania MUA są np. Outlook Express lub Mozilla Thunderbird. Na serwerach pośredniczących w przesyłaniu poczty musi być zainstalowane oprogramowanie MTA (Mail Transport Agent).

Jak widać, funkcjonowanie mechanizmów poczty elektronicznej jest oparte o maszyny serwerowe. Przypatrzymy się teraz nieco dokładniej roli jaką pełnią w procesie wymiany wiadomości. Poczta jest przesyłana między serwerami z których korzystają nadawcy i odbiorcy. Nadawca wysyła wiadomość przez serwer na którym posiada konto pocztowe, następnie serwer przesyła pocztę do serwera docelowego — na którym adresat posiada konto. W dalszej kolejności adresat przesyła (pobiera) pocztę na swoją stację roboczą. Poczta nie jest przesyłana bezpośrednio między komputerami nadawcy i odbiorcy. Ma to pewne konsekwencje praktyczne - nie musimy mieć cały czas włączonego komputera!

W ogólności, listy elektroniczne mogą być przesyłane bezpośrednio między stacjami roboczymi użytkowników. Wymaga to jednak zainstalowania nie tylko oprogramowania klienckiego MUA, ale także MTA oraz odpowiedniej konfiguracji sieci. Komputer musi być stale włączony.

Pod pojęciem *serwera pocztowego* rozumiemy oprogramowanie MTA zainstalowane na komputerze (serwerze). Należy zwrócić uwagę na podwójne znaczenie słowa serwer, w jednym przypadku oznaczające maszynę, w innym oprogramowanie. Potocznie mianem serwera pocztowego określa się fizyczny serwer (komputer) z zainstalowanym oprogramowaniem MTA, czyli obie wspomniane rzeczy na raz.

MTA (serwer pocztowy) najczęściej funkcjonuje też jako tzw. przekaźnik poczty, (pełni wtedy rolę serwera, przez który klient wysyła pocztę) oraz jako serwer końcowego przeznaczenia dla domeny lub domen, przyjmujący pocztę wysyłaną (przez klientów) za pomocą różnych serwerów do danej domeny/domen.

Pod pojęciem serwera rozumiemy komputer, włączony przez cały czas, mający stały dostęp do Internetu oraz posiadający stały adres IP.

Z wymienionych powodów, taka sytuacja występuje niezwykle rzadko.

Oprogramowanie służące do wysyłania i odbierania poczty

- MUA
 - aplikacja użytkownika
 - służy do
 - * tworzenia poczty
 - * wysyłania jej przez serwer
 - * odbierania z serwera
- MTA
 - aplikacja serwera
 - służy do
 - * przesyłania poczty przekazywanej przez MUA dalej
 - * *w świat*, czyli do serwera z którego korzysta adresat

Protokoły służące do wysyłania i odbierania poczty

- SMTP (ang. Simple Mail Transfer Protocol)
 - wysyłanie wiadomości przez serwer
 - przesyłanie między serwerami
- POP3 (ang. Post Office Protocol)
 - odbieranie z serwera
- IMAP (ang. Internet Message Access Protocol)
 - odbieranie z serwera

5.1.5 Adresy poczty elektronicznej

Adres pocztowy składa się z: nazwy użytkownika, znaku @ i nazwy domeny internetowej. W adresie e-mail nie mogą występować spacje. Dla danego konta pocztowego można tworzyć dodatkowe nazwy tzw. aliasy. Np. użytkownik *janko* posiadający adres e-mail *janko@mimuw.edu.pl*, może posiadać alias *Jan.Kowalski@mimuw.edu.pl*. Poczta wysłana na adres *Jan.Kowalski@mimuw.edu.pl* trafi na konto *janko*, tak samo zresztą, jak wysłana na *janko@mimuw.edu.pl*.

Adresy pocztowe i aliasy

- Adres pocztowy składa się z
 - nazwy użytkownika, znaku @ (at), nazwy domeny internetowej
- Adres *janko@mimuw.edu.pl* jest poprawnym adresem email

- Aliasy to alternatywne nazwy konta pocztowego
 - Jan.Kowalski@mimuw.edu.pl
 - J.Kowalski@mimuw.edu.pl
- Użytkownik może posiadać dowolną ilość aliasów
 - w granicach rozsądku
- Nazwa użytkownika lub alias może się składać z
 - małych i dużych liter
 - cyfr
 - znaków: . _ -
- Małe i duże litery są traktowane jednakowo
 - jan.kowalski@mimuw.edu.pl
 - Jan.Kowalski@mimuw.edu.pl
 - to ten sam adres
- Inne znaki niż wymienione wyżej są niedopuszczalne
 - nie można też używać znaków narodowych!

5.2 Konfiguracja

W rozdziale tym zostaną omówione aspekty związane z konfiguracją oprogramowania klienckiego poczty elektronicznej (MUA). Nie będzie natomiast informacji dotyczących programów MTA, tematyka ta wykracza poza zakres niniejszego kursu. Należy założyć, że MTA jest uruchomiony na serwerze i został poprawnie skonfigurowany przez administratora. Aby móc dokonać konfiguracji klienta poczty, niezbędne jest posiadanie konta na serwerze pocztowym.

5.2.1 Parametry potrzebne do konfiguracji klienta poczty

Poniższe parametry można uzyskać od osoby lub firmy, która utrzymuje serwer pocztowy i dokonała założenia konta dla użytkownika.

Informacje niezbędne do konfiguracji klienta poczty

- Nazwa wyświetlana
 - często podaje się imię i nazwisko
- Adres e-mail
 - składa się z identyfikatora, znaku @ oraz nazwy domeny internetowej

- identyfikator skrzynki pocztowej (login), oraz hasło
- adres IP lub nazwa serwera SMTP (wysyłanie)
- adres IP lub nazwa serwera POP3 (odbieranie)
- opcjonalnie może być wykorzystany serwer IMAP
 - wtedy zamiast adresu lub nazwy serwera POP3 trzeba podać adres lub nazwę serwera IMAP

Przykładowe dane konfiguracyjne

- Nazwa wyświetlana (imię i nazwisko studenta)
- Adres e-mail złożony z
 - nazwy konta studenta używanej do logowania w laboratorium
 - znaku @
 - domeny *students.mimuw.edu.pl*
 - przykład *student1@students.mimuw.edu.pl*
- Identyfikator skrzynki pocztowej oraz hasło
 - login, czyli nazwa konta
 - hasło takie jak używane do logowania w laboratorium
- adres IP lub nazwa serwera SMTP
 - *students.mimuw.edu.pl*
- adres IP lub nazwa serwera POP3
 - *students.mimuw.edu.pl*

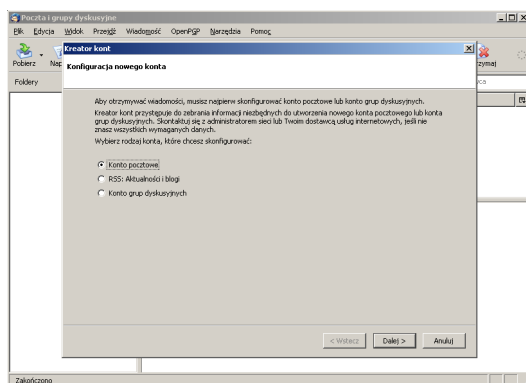
Podane parametry konfiguracyjne są prawdziwe jedynie dla zajęć odbywających się w laboratorium PI na Wydziale MIM UW. W innych przypadkach należy je traktować jedynie jako przykład.

5.2.2 Konfiguracja programu Mozilla Thunderbird

W niniejszej instrukcji konfiguracja klienta poczty zostanie przedstawiona na przykładzie oprogramowania open source Mozilla Thunderbird. Thunderbird jest zainstalowany w laboratorium MIM UW. Można go też pobrać z sieci:

<http://www.thunderbird.pl>

Początek konfiguracji Thunderbirda

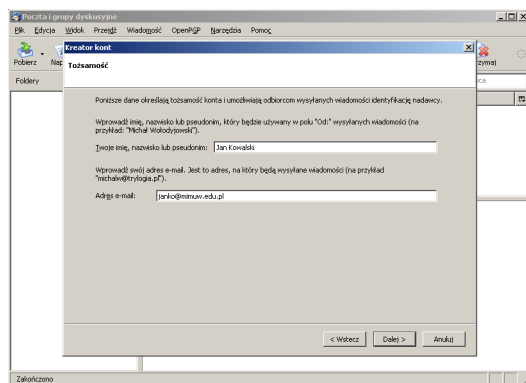


Po uruchomieniu programu pojawi się pierwsze okno konfiguracji

- Należy wybrać opcję *Konto pocztowe*
- Aby przejść do kolejnego etapu konfiguracji, należy użyć przycisku *Dalej*
- Thunderbird obsługuje też grupy dyskusyjne

Pierwsze okno konfiguracyjne pojawia się zaraz po uruchomieniu, jeśli program nie był jeszcze konfigurowany.

Konfiguracja adresu e-mail

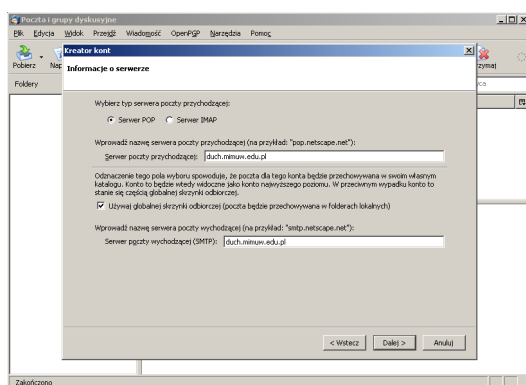


Wpisujemy imię i nazwisko oraz adres e-mail

- przykład *student1@students.mimuw.edu.pl*

To co wpisujemy w pierwszym polu tego okna konfiguracji, będzie widoczne przez odbiorców w polu *Od*, czyli będzie identyfikatorem naszej wiadomości. Oznacza to, iż te dane osobowe (imię i nazwisko) będą przekazywane przez sieć do adresata. Jeżeli istnieje potrzeba ochrony tożsamości, nie należy w tym miejscu podawać imienia i nazwiska.

Konfiguracja serwerów POP3 i SMTP

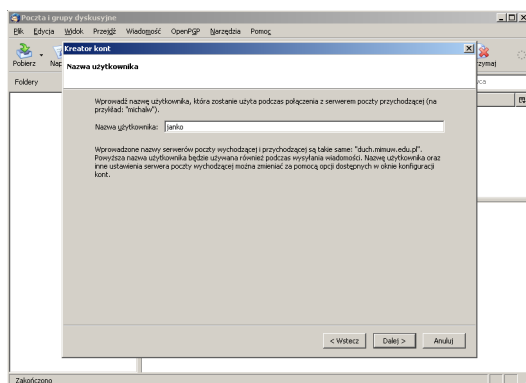


- Wybieramy typ serwera obsługujący pocztę przychodzącą
 - odbieraną (*ściąganą* z serwera do naszego komputera)
- Następnie podajemy nazwy serwerów do wysyłania i odbierania poczty
 - przykład: students.mimuw.edu.pl

Trzeba znać typ serwera poczty przychodzącej, oraz nazwy serwerów obsługujących pocztę wysyłaną i odbieraną (przychodzącą). Informacje te można uzyskać np. od firmy która obsługuje serwer pocztowy. W naszym przypadku nazwy serwera poczty przychodzącej i wychodzącej są takie same. Oznacza to, że jedna maszyna wykonuje zadania wysyłania oraz przesyłania poczty (SMTP), oraz transferowania poczty na stację roboczą odbiorcy (POP). Jest to bardzo powszechna konfiguracja. Istotne jest, aby zwrócić uwagę na używaną terminologię:

- Poczta wychodząca, to poczta wysyłana z naszego komputera za pośrednictwem serwera na którym mamy konto pocztowe z użyciem protokołu SMTP.
- Poczta przychodząca to poczta przesyłana z serwera, na którym mamy konto, na nasz komputer - protokół POP. Serwer ten jest docelową maszyną, na którą trafia poczta wysyłana na nasz adres.

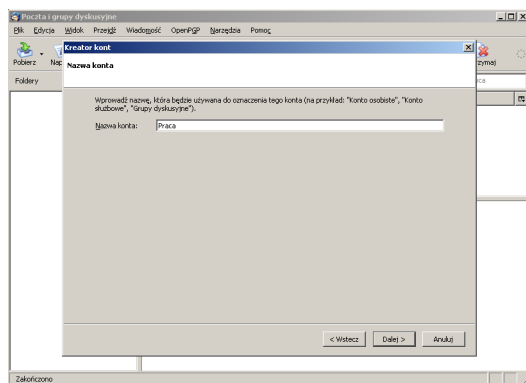
Konfiguracja nazwy użytkownika



- Wpisujemy nazwę użytkownika, która identyfikuje nasze konto pocztowe na serwerze
- W tym okienku konfiguracyjnym jest to konto na serwerze obsługującym pocztę przychodzącą

Nazwa użytkownika identyfikuje konto pocztowe danego użytkownika na serwerze, ale sama nazwa nie jest adresem e-mail. Tej nazwy (oraz hasła), klient poczty używa do autoryzacji użytkownika po połączeniu z serwerem pocztowym. Dopiero nazwauzytkownika@domena jest jego adresem e-mail.

Konfiguracja nazwy konta



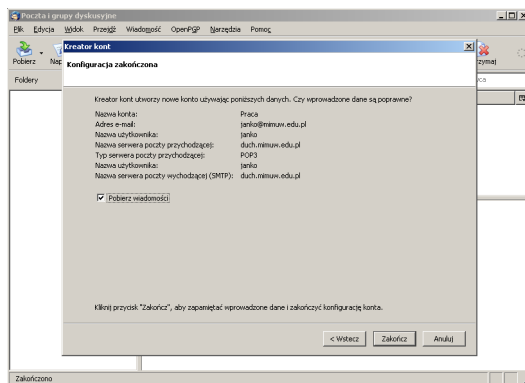
- Wpisujemy nazwę, która będzie identyfikować konto w Thunderbirdzie

Ponieważ Thunderbird oraz większość klientów poczty, umożliwiają obsługę więcej niż jednego konta pocztowego, w tym miejscu program prosi o nadanie nazwy identyfikującej dane konto. Nazwa ta nie jest związana z nazwą użytkownika identyfikującą konto na serwerze! W szczególności nie należy tu podawać nazwy będącej nazwą użytkownika, gdyż często jest tak,

iz posiadając konta na różnych serwerach, używa się tej samej nazwy użytkownika. Należy podać raczej jakąś inną (dowolną, wymyśloną nazwę — np. *konto-mimuw*).

- Można posiadać konto na serwerze *duch.mimuw.edu.pl* obsługującym domenę *mimuw.edu.pl* o nazwie użytkownika *janko* i adresie email *janko@mimuw.edu.pl*.
- Jednocześnie przecież można mieć konto *janko* na serwerze *mail.fuw.edu.pl* o nazwie użytkownika *janko* i adresie email *janko@fuw.edu.pl*.

Końcowy etap konfiguracji

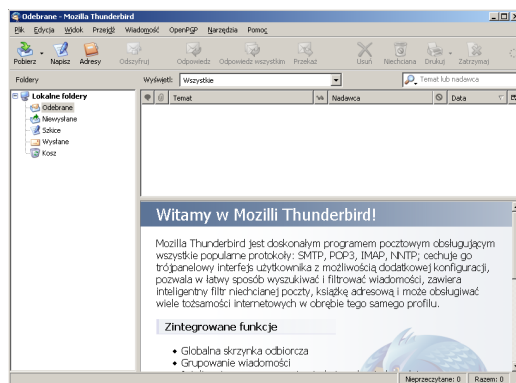


- Aby ukończyć konfigurację, należy użyć przycisku *Zakończ*

Można zostawić opcję *Pobierz wiadomości* zaznaczoną, wtedy po użyciu przycisku *Zakończ* program spróbuje pobrać wiadomości z serwera.

5.2.3 Używanie programu Thunderbird

Główne okno programu



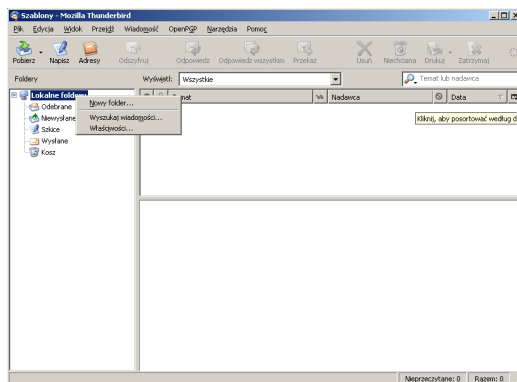
Widok *Foldery*

- przedstawia katalogi, w których będą trzymane wiadomości przychodzące, oraz utworzone i wysłane

Nie będą w tym miejscu opisywane wszystkie ikony znajdujące się w głównym oknie programu. Warto natomiast wspomnieć o przeznaczeniu folderów lokalnych.

- W folderze *Otrzymane* znajdują się wiadomości przychodzące. Wiadomość po pobraniu z serwera jest umieszczana właśnie w tym katalogu
- W katalogu *Niewysłane* są umieszczane wiadomości, których z jakiegoś powodu nie można wysłać
- Folder *Szkice* służy do przechowywania wiadomości, których pisanie nie zostało ukończone. W oknie tworzenia wiadomości należy wybrać *Zapisz jako ► szkic*
- W katalogu *Wysłane* są umieszczane wiadomości, które utworzyliśmy, po ich poprawnym wysłaniu

Zarządzanie katalogami

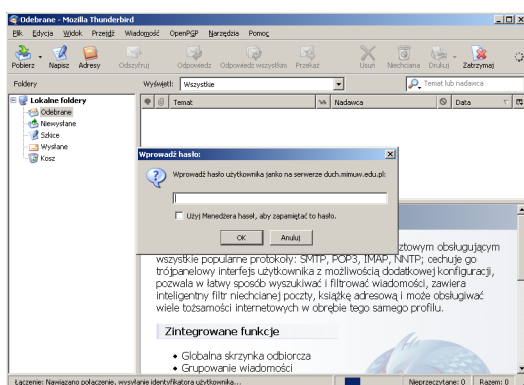


Aby utworzyć własny katalog należy kolejno wybrać:

- prawy klawisz myszy na katalogu *Foldery lokalne*, Utwórz katalog
- aby dodać podkatalog
 - prawy klawisz myszy na katalogu, w którym ma być umieszczony podkatalog

Jak widać, w programie Thunderbird można też korzystać z innych niż standardowe, własnoręcznie tworzonych katalogów. Katalogi takie przydają się np. do sortowania wiadomości za pomocą narzędzia *Filtrowanie wiadomości* o którym będzie mowa w dalszej części rozdziału.

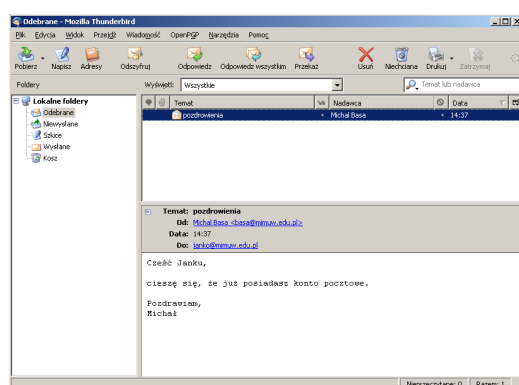
Odbieranie poczty



Abu odebrać pocztę, należy:

- nacisnąć przycisk *Odbierz*
- podać hasło do konta pocztowego na serwerze
- jeśli nie chcemy podawać hasła przy każdym pobieraniu poczty należy:
 - zaznaczyć opcję *Użyj menadżera haseł, aby zapamiętać to hasło*
- Hasło zazwyczaj dostajemy wraz z innymi parametrami konta pocztowego (nazwą serwera itd.)
- Należy pamiętać, iż jeśli zdecydujemy się na zapamiętanie hasła, będzie ono przechowywane lokalnie na naszym komputerze. W pewnych okolicznościach może to być niewskazane

Czytanie wiadomości



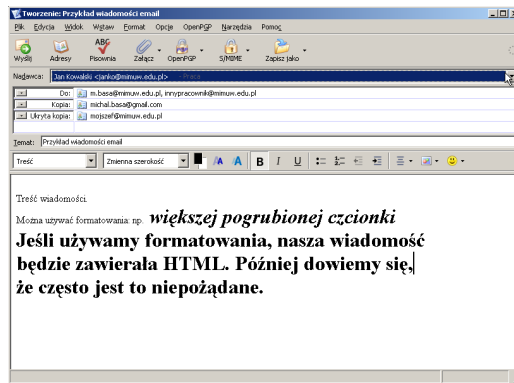
- Po odebraniu wiadomości, pojawi się ona w folderze *Odebrane*
- Aby przeczytać wiadomość, należy kliknąć na folderze *Odebrane*, a następnie na wiadomości, która pojawi się po prawej stronie

Jeśli do komputera mają dostęp osoby, którym nie ufamy, lepiej nie zapamiętywać hasła w programie Thunderbird

- Wiadomości można sortować, klikając na pasek *Tytuł, Nadawca, Data*

Symbol agrałki oznacza sortowanie wg załączników. Należy dodać, iż po ustawieniu kursora myszy np. na pasku *Tytuł* program wyświetla podpowiedź w której jest mowa o funkcji paska.

Tworzenie wiadomości



Aby utworzyć nową wiadomość, należy kolejno wybrać:

- ikona *Napisz*, okno tworzenia wiadomości
- w polu *Do:* należy wpisać adres korespondenta
 - w polu *Do:* może być więcej niż jeden adres
- w polu *Temat* krótko podajemy temat korespondencji

Oprócz pola *Do:* istnieją jeszcze inne typy pól:

- Pole *Kopia* służy do wpisywania adresów email korespondentów, którzy również otrzymają wiadomość. To samo można uzyskać wpisując więcej niż jeden adres w pole *Do:*
- Pole *Ukryta kopia* ma nieco inne zastosowanie. Adresaci umieszczeni w polach *Do:* oraz *Kopia:* nie będą widzieli, że korespondencja została wysłana jeszcze do kogoś, adres szef@mimuw.edu.pl nie pojawi się w e-mailu do m.basa@mimuw.edu.pl

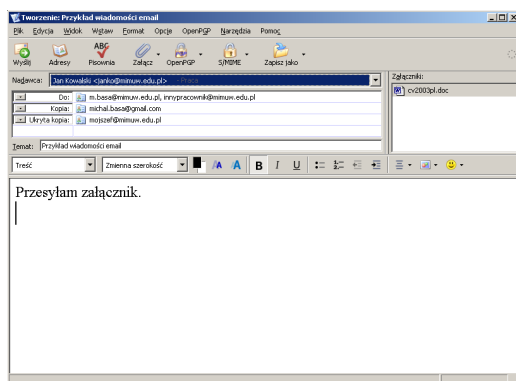
Pisząc wiadomość można używać formatowania, jednak wtedy będzie ona zawierać HTML. Nie wszystkie programy klienckie poczty potrafią obsługiwać HTML. W związku z tym lepiej nie używać HTML-a jeśli nie mamy pewności, czy odbiorca będzie mógł poprawnie wyświetlić naszą wiadomość.

Odpowiadanie na wiadomość

- Aby odpowiedzieć na wiadomość, należy użyć ikony *Odpowiedz*
- Nie trzeba już wpisywać adresu odbiorcy

Pole *Kopia:* bywa w niektórych programach nazywane *Dw:* (Do wiadomości), pole *Ukryta kopia:* - *Udw:*. Czasem stosuje się też angielskie określenia: *Cc:* (ang. Carbon copy) oraz odpowiednio *Bcc:* (ang. Blind carbon copy).

Wysyłanie poczty z załącznikami



Aby wysłać pocztę plik, należy kolejno wybrać:

- ikonę *Załącz* w oknie tworzenia wiadomości
- wybrany plik zostanie umieszczony w okienku załączników

Pocztą elektroniczną można przysyłać dowolne typy plików. Do jednej wiadomości można załączyć więcej niż jeden zbiór, powtarzając opisaną wyżej procedurę. Należy jednak zaznaczyć, iż istnieje limit związany z sumą wielkości załączonych plików i treści maila. Większość serwerów pocztowych nie przesyła wiadomości większych niż 10–15MB. Czasem jest to nawet 5MB. Poczta elektroniczna nie powinna być wykorzystywana do przesyłania dużych ilości danych przez sieć. Do tego celu służą protokoły FTP i HTTP. Sensowna wielkość całej wiadomości (treści + załączników) to maksymalnie kilka MB. Jeśli spróbujemy wysłać pocztę, której wielkość przekracza określony na serwerze limit, operacja nie powiedzie się, a nadawca otrzyma stosowny komunikat o błędzie.

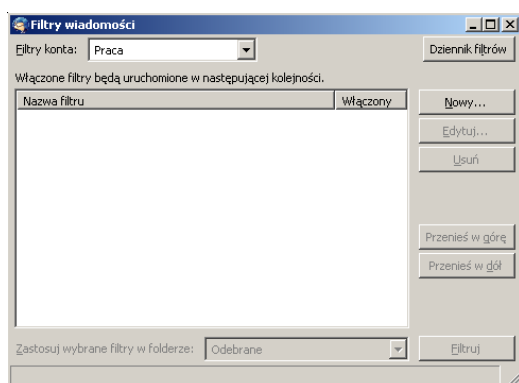
Używanie książki adresowej

Aby korzystać z książki adresowej należy kolejno wybrać:

- ikona *Adresy*,
- Z dodanych wpisów adresowych będzie można skorzystać wypełniając pola *Do:*, *Kopia:* itd. w oknie tworzenia wiadomości

5.2.4 (Trochę) bardziej zaawansowana konfiguracja

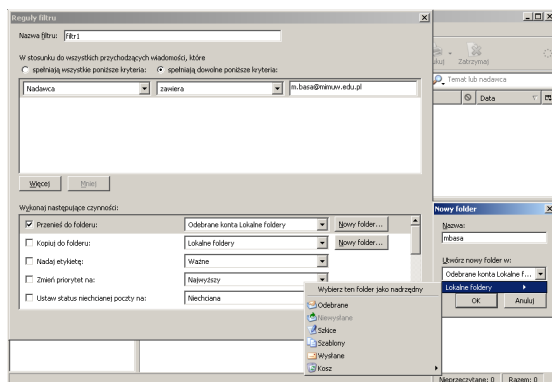
Filtry wiadomości



Aby utworzyć filtr, należy kolejno wybrać:

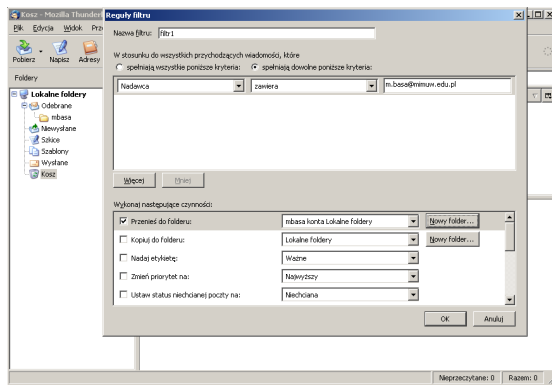
menu Narzędzia ► Filtrowanie wiadomości ► przycisk Nowy...

Tworzenie nowego filtra



- Z poziomu tworzenia filtra można założyć folder
- Jako nadrzędny folder, wybieramy *Odebrane*

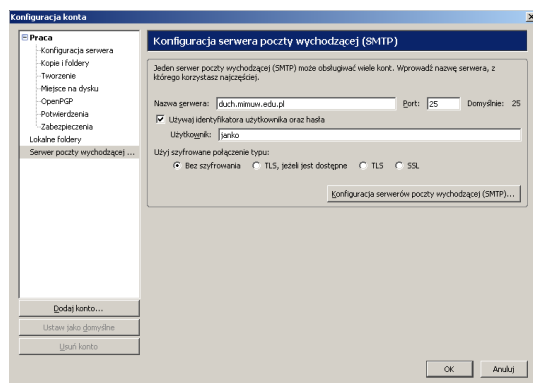
Filtrowanie poczty umożliwia tworzenie reguł dopasowania i przypisywanie im różnych akcji. Można kopiować, przenosić i np. nadawać określonym wiadomościom priorytetu. W tym miejscu należy wspomnieć, iż nadawanie priorytetu jest właściwe dla klienta poczty, przez tworzenie dodatkowych nagłówek w wiadomości, a nie jest wymaganą właściwością zdefiniowaną w dokumencie określającym tworzenie wiadomości (odpowiedni dokument RFC). Oznacza to, iż nie wszystkie programy klienckie obsługują priorytety. Wiadomość z ustawionym priorytetem będzie przez taki program traktowana jako zwykły email i nie zostanie w żaden sposób oznaczona.



Folder został utworzony

- Wiadomości od zdefiniowanego adresata będą trafiały do tego folderu
- Należy kliknąć przycisk *OK* w celu zapamiętania zdefiniowanego filtra

Autoryzacja SMTP



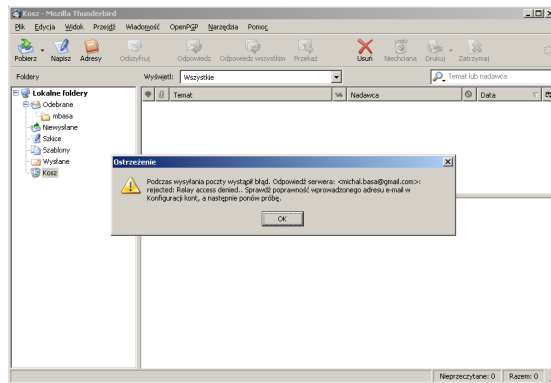
Aby włączyć autoryzację SMTP należy kolejno wybrać:

menu Narzędzia ► Konfiguracja kont ► Serwer poczty wychodzącej...

W Thunderbirdzie Opcja *Używaj identyfikatora użytkownika oraz hasła* jest domyślnie ustawiona. Jak pamiętamy, podawaliśmy nazwę użytkownika tylko przy konfiguracji poczty przychodzącej (do autoryzacji POP). Program Thunderbird zakłada, że do autoryzacji SMTP (poczty wysyłanej) należy użyć tej samej nazwy użytkownika. Tak to rzeczywiście najczęściej bywa. Gdybyśmy jednak chcieli, do wysyłania poczty użyć innego serwera niż do odbierania, bądź też wysyłać za pomocą innego konta na tym samym serwerze, jest możliwy ręczny wybór. Służy do tego okienko *Użytkownik*. Należy też pamiętać, że nie we wszystkich programach pocztowych opcja autoryzacji SMTP jest domyślnie ustawiana. Np. w programie Outlook Express trzeba to zrobić samemu. W przypadku ww. Outlooka opcja ta nazywa się

inaczej niż w Thunderbirdzie: *Serwer wymaga uwierzytelnienia*. Jeśli serwer przez który wysyłamy pocztę, faktycznie wymaga uwierzytelnienia, a nie jest ono ustawione w programie klienta (np. naszym Thunderbirdzie) wiadomość nie zostanie wysłana a serwer odpowie komunikatem: *Relay access denied*.

Problem z uwierzytelnieniem SMTP



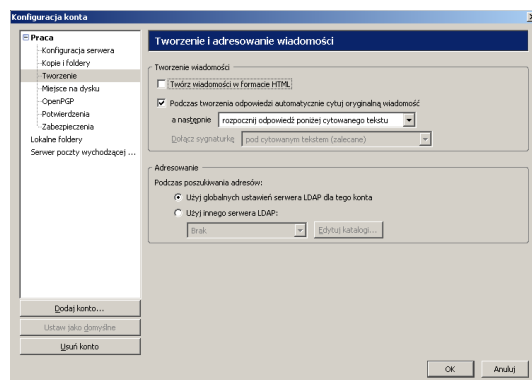
Komunikat *Relay access denied* oznacza

- nie mamy uprawnień do wysyłania poczty przez dany serwer
- w tym przypadku wynika to z braku ustawienia autoryzacji SMTP

Często serwery są konfigurowane w następujący sposób:

- Przy wysyłaniu do niektórych (zaufanych) domen autoryzacja SMTP nie jest potrzebna
- Przy wysyłaniu z niektórych sieci lub z niektórych adresów IP klientów, autoryzacja również nie jest wymagana

Wyłączenie HTML-a



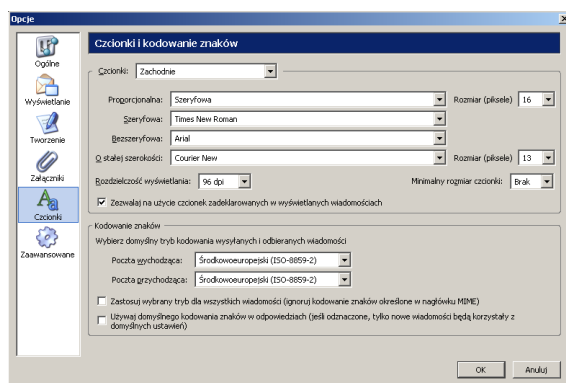
Aby wyłączyć tworzenie wiadomości w formacie HTML należy kolejno wybrać:

menu Narzędzia ►Konfiguracja kont ►

Tworzenie ►Twórz wiadomości w formacie HTML

- Bardzo sensowne, z przyczyn opisanych wcześniej jest nie stosowanie HTML-a w poczcie elektronicznej
- Opcja *Podczas tworzenia odpowiedzi automatycznie cytuj oryginalną wiadomość a następnie rozpocznij odpowiedź poniżej cytowanego tekstu* powoduje, iż w momencie użycia funkcji *Odpowiedz* w wiadomości zamieszczany jest e-mail na który odpowiadamy a kursor jest umieszczony poniżej cytatu. Jest to sensowne ustawienie
- Szczególnie, jeśli piszemy wiadomości do grup dyskusyjnych, należy zawsze odpisywać pod cytatem

Ustawianie kodowania znaków narodowych



Aby ustawić kodowanie znaków narodowych należy kolejno wybrać:

menu Narzędzia ►Opcje ►Czcionki

- domyślne ustawienie *ISO 8859-2* jest dobre
- można też używać *UTF-8*
- niektóre klientki programy pocztowe (np. *pine*) nie potrafią obsługiwać UTF-8

5.3 Zagadnienia związane z bezpieczeństwem poczty elektronicznej

Poczta elektroniczna służy do przesyłania wszelkiego typu danych na duże odległości przy zastosowaniu rozległej sieci Internet. W tym świetle, zagadnienia bezpieczeństwa mają nietrywialne znaczenie. Konto pocztowe użytkownika, oraz jego stacja robocza mogą być narażone na wiele niebezpieczeństw.

5.3.1 Zagrożenia dla poczty elektronicznej

- Przechwycenie hasła
 - z tym związane jest uzyskanie dostępu do konta pocztowego użytkownika
- Otrzymywanie wirusów w poczcie
 - tym samym narażenie stacji roboczej użytkownika
 - szczególnie zagrożeni są użytkownicy systemów operacyjnych Microsoftu
- Otrzymywanie dużej liczby niechcianych wiadomości (tzw. spamu)

5.3.2 Sposoby wyeliminowania zagrożeń

- Poprawna konfiguracja oprogramowania klienta
 - np. Thunderbirda
- Poprawna konfiguracja oprogramowania MTA na serwerze
 - to już nie leży w gestii użytkownika poczty
- Dodatkowe oprogramowanie na komputerze klienta i/lub na serwerze
- Używanie szyfrowania

5.3.3 Algorytmy szyfrujące w zastosowaniu do poczty elektronicznej

Dwie główne koncepcje szyfrowania danych bazują na wykorzystaniu symetrycznego lub niesymetrycznego algorytmu szyfrującego.

Szyfrowanie symetryczne polega na użyciu tylko jednego klucza, ten sam klucz musi być w posiadaniu nadawcy i odbiorcy, gdyż nim się szyfruje i deszyfruje wiadomości. Szyfrowanie z użyciem algorytmu symetrycznego ma poważną wadę – musi istnieć niepodważalnie bezpieczny kanał dostarczenia klucza korespondentowi. Można to łatwo zrobić np. spotykając się

i przekazując klucz osobiście. Problem występuje, jeśli korespondent jest zbyt oddalony, aby możliwy był osobisty kontakt.

Szyfrowanie niesymetryczne polega na użyciu pary kluczy – publicznego i prywatnego. Wiadomość zaszyfrowana kluczem prywatnym może być odszyfrowana jedynie odpowiadającym mu kluczem publicznym. I analogicznie, wiadomość zaszyfrowana kluczem publicznym, może być odszyfrowana tylko odpowiadającym mu kluczem prywatnym. Jeśli chcemy zaszyfrować wiadomość do naszego korespondenta, musi on przysłać nam swój klucz publiczny. Klucz nie musi jednak być przysłany w bezpieczny sposób. Jeśli ktoś przechwyci klucz publiczny nie będzie mógł odszyfrować wiadomości nim zaszyfrowanych. Użytkownicy poczty, którzy chcą szyfrować do siebie wiadomości muszą się wymienić jedynie kluczami publicznymi. Poczta może więc być szyfrowana wyżej opisanym mechanizmem.

5.3.4 Szyfrowanie poczty

- Korespondenci wymieniają się kluczami publicznymi
 - muszą je wprawdzie utworzyć
- Nadawca szyfruje wiadomość dla odbiorcy kluczem publicznym odbiorcy
- Odbiorca odszyfrowuje e-mail swoim kluczem prywatnym
- Potrzebne jest dodatkowe oprogramowanie szyfrujące
 - np. GNUPG
- Za pomocą oprogramowania szyfrującego użytkownik musi wygenerować parę kluczy
 - prywatny i publiczny
- Użytkownik musi udostępnić klucz publiczny
 - rozesłać do wszystkich osób, które będą przysyłać do niego zaszyfrowane wiadomości
 - można umieścić klucz publiczny na stronie WWW
 - korespondenci będą mogli go łatwo pobierać

5.3.5 Podpisywanie poczty

- Korespondenci wymieniają się kluczami publicznymi
- Nadawca podpisuje wiadomość dla odbiorcy swoim kluczem prywatnym
- Odbiorca weryfikuje podpis za pomocą klucza publicznego nadawcy
- Użytkownik musi przysłać klucz publiczny do osób, które będą weryfikować jego podpis

Stosowane tu określenie *podpis elektroniczny* nie jest podpisem elektronicznym zgodnym z obowiązującym prawem. Odpowiednia ustawa mówi, że do składania podpisu, należy używać m.in. karty mikroprocesorowej, na której znajdują się klucze. Więcej informacji można znaleźć np. na <http://www.certum.pl>.

Podpisywanie wiadomości przebiega w następujący sposób:

- Jest tworzony tzw. skrót wiadomości, która ma być podpisana. Należy wyjaśnić, iż skrót jest utworzony w taki sposób, aby był unikalny dla danej wiadomości. Dba o to oprogramowanie szyfrujące
- Skrót jest następnie szyfrowany kluczem prywatnym twórcy wiadomości i załączany do oryginalnej wiadomości
- Wiadomość oraz jej zaszyfrowany skrót (który jest właśnie podpisem elektronicznym) są przesyłane do odbiorcy

Weryfikowanie podpisu odbywa się w następujący sposób:

- Jest tworzony skrót otrzymanej wiadomości
- Skrót otrzymany od nadawcy jest odszyfrowywany kluczem publicznym nadawcy (odbiorca musi ten klucz posiadać)
- Jeśli dało się odszyfrować skrót przysłany przez nadawcę jego kluczem publicznym, to oznacza, że nadawca zaszyfrował skrót swoim kluczem prywatnym, więc można mieć pewność, że to nadawca jest faktycznym autorem wiadomości. Tylko nadawca ma dostęp do swojego klucza prywatnego, czyli to nadawca złożył podpis
- Jeśli odszyfrowany skrót, jest taki sam jak utworzony skrót, to dodatkowo istnieje pewność, że wiadomość nie została w drodze do odbiorcy zmodyfikowana. Pamiętajmy przecież, że skrót jest unikalny dla danej wiadomości

Podpis elektroniczny oprócz tego, że pozwala stwierdzić autentyczność wiadomości, dodatkowo zapewnia mechanizm wykrywania modyfikacji listu. Jak już wspomniano, tak rozumiany podpis nie jest podpisem elektronicznym w myśl ustawy. Ustawa narzuca dodatkowe wymagania, np. używanie kart mikroprocesorowych do przetrzymywania kluczy. A w naszym przypadku, klucze znajdują się po prostu na twardym dysku komputera.

5.3.6 Sprawdzanie autentyczności klucza publicznego

- *Ja jestem Jan Kowalski, a to jest mój klucz publiczny*
- Możemy uwierzyć w to, że to klucz Jana, jeśli znamy Jana
- Jeśli nie znamy Jana, to jest problem
- Ale jest też rozwiązanie bo klucze też można podpisywać !

Klucz publiczny Jana Kowalskiego został podpisany kluczem prywatnym organizacji CA (ang. Certification Authority). CA jest to organizacja, która zanim podpisała klucz publiczny Jana Kowalskiego sprawdziła jego tożsamość. W takim wypadku korespondenci Jana Kowalskiego, którzy będą weryfikować jego podpis, mogą zweryfikować najpierw jego klucz publiczny. Czyli sprawdzić podpis na kluczu publicznym Jana Kowalskiego za pomocą klucza publicznego CA, który można pobrać np. ze strony WWW CA. Gdy weryfikacja podpisu powiedzie się, będzie to oznaczać, że klucz publiczny rzeczywiście należy do Jana Kowalskiego.

Szyfrowanie poczty w komunikacji między serwerem a klientem oraz w komunikacji między serwerami

- Szyfrowanie poczty między klientami
 - zależne od woli użytkownika
 - użytkownik generował klucze, przysyłał je korespondentom itd.
 - omówione poprzednio
- Do szyfrowania było potrzebne dodatkowe oprogramowanie
- Szyfrowanie poczty między serwerem a klientem
 - SSL (Secure Socket Layer)
 - wbudowany mechanizm szyfrujący klienta i serwera
 - niezależny od użytkownika
 - użytkownik musi tylko włączyć ich obsługę w kliencie poczty
- Większość klientów pocztowych nie wymaga do tego dodatkowego oprogramowania
- Szyfrowana nie tyle wiadomość pocztowa
 - cały kanał komunikacji między serwerem a klientem bądź między serwerami
- Mechanizm zapobiega przechwytywaniu hasła użytkownika na drodze klient-serwer
- Poprzednio opisany mechanizm nie chronił przesyłanego od klienta do serwera hasła
 - chronił tylko wiadomość

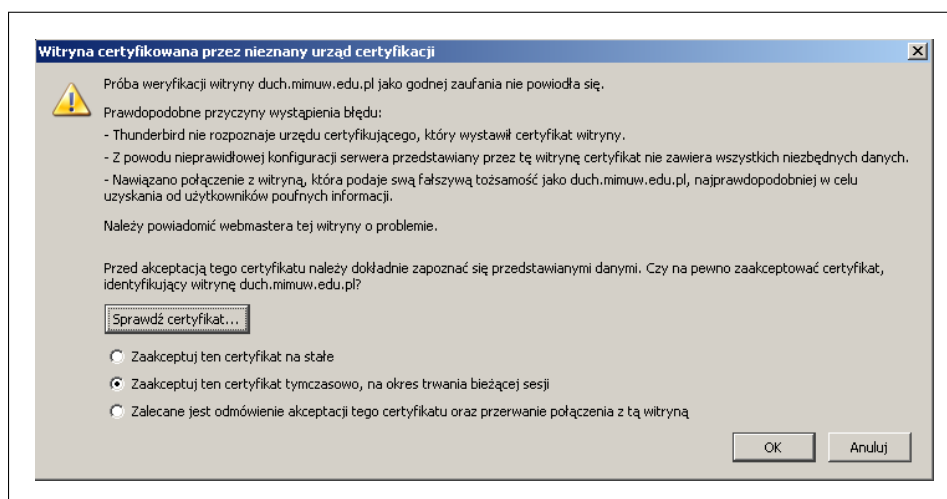
Ten mechanizm chroni hasło, ale nie chroni wiadomości, ponieważ po dojściu na serwer, wiadomość zostaje odszyfrowana. Musi tak być, ponieważ proces szyfrowania był wykonany przez klienta i serwer bez żadnej ingerencji użytkownika. W szczególności użytkownik nie znał kluczy wykorzystywanych do szyfrowania, i nie mógłby odczytać wiadomości.

Szyfrowania mechanizmem niezależnym od użytkownika, przy wykorzystaniu SSL/TLS jest dużo prostsze do skonfigurowania. Nie trzeba tworzyć kluczy, wymieniać się nimi itd. Tak jak już wspomniano, od strony klienta nie potrzeba dodatkowego oprogramowania, a konfiguracja serwera nie leży przecież w gestii użytkownika poczty. Jeśli serwer obsługuje szyfrowanie SSL/TLS, zaleca się także włączenie szyfrowania na kliencie poczty. Nie chroni to naszych wiadomości przed przeczytaniem, ale przynajmniej zapewnia bezpieczeństwo przy przesyłaniu hasła podczas autoryzacji (np. gdy chcemy pobrać pocztę). Ten niezależny od użytkownika mechanizm szyfrowania przebiega w dość prosty sposób.

Przebieg szyfrowania między klientem a serwerem

- Klient zgłasza chęć użycia szyfrowania serwerowi
- Serwer wysyła klientowi swój klucz publiczny
 - certyfikat zawierający klucz publiczny
- Klient generuje klucz sesji, szyfruje go kluczem publicznym serwera, przesyła do serwera
- Dalszy ciąg komunikacji odbywa się przez kanał szyfrowany kluczem sesji
 - z użyciem algorytmu symetrycznego

Rzecz, na którą należy zwrócić uwagę, to autentyczność klucza publicznego serwera. Powinien on być podpisany kluczem prywatnym CA. Klient poczty, który otrzyma np. klucz publiczny serwera podpisany nieznanym mu kluczem prywatnym CA, wyświetli ostrzeżenie:

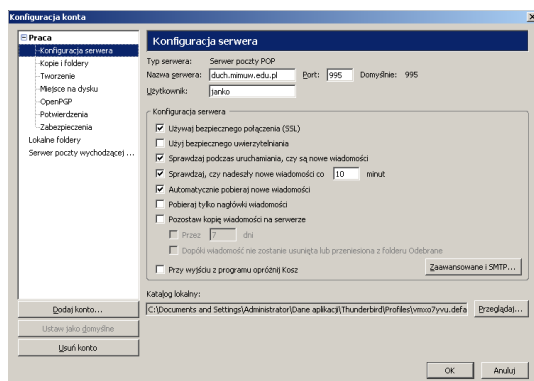


Sprawdzanie autentyczności klucza serwera

Klienckie programy pocztowe standardowo posiadają klucze publiczne znanych CA.

5.3.7 Włączenie szyfrowania w programie Thunderbird

Protokół pobierania poczty POP

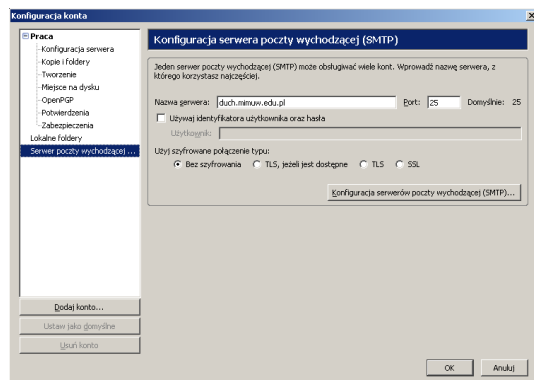


Aby włączyć szyfrowanie poczty dla POP należy kolejno wybrać:

menu Narzędzia ►Konfiguracja konta ►

Konfiguracja serwera ►Używaj bezpiecznego połączenia SSL

Protokół wysyłania poczty SMTP



Aby włączyć szyfrowanie poczty dla SMTP należy kolejno wybrać:

menu Narzędzia ►Konfiguracja konta ►

Serwer poczty wychodzącej ►TLS, jeśli dostępne

TLS jest nowszą wersją SSL. Jeśli zaznaczymy *TLS, jeśli dostępne*, będzie można używać też SSL, jeśli serwer nie obsługuje TLS.

5.3.8 Ochrona antywirusowa

Poczta elektroniczna jest narzędziem wykorzystywanym niestety także przez wirusy komputerowe. Niektóre wirusy posiadają mechanizmy, które umożliwiają im masowe rozsyłanie się z komputera użytkownika bez jego wiedzy. Dla zmniejszenia ryzyka otrzymywania wirusów w poczcie elektronicznej należy:

- Dowiedzieć się, czy na serwerze z którego korzystamy jest zainstalowane oprogramowanie antywirusowe. W obecnych czasach taka konfiguracja serwera powinna być standardem. Informacje takie możemy uzyskać od administratora serwera
- Zainstalować na stacji roboczej oprogramowanie antywirusowe
- Jeśli używamy systemu operacyjnego Windows, należy bezwzględnie zainstalować oprogramowanie antywirusowe. Większość współczesnych wirusów komputerowych jest tworzona dla systemu Windows

5.3.9 Ochrona antyspamowa

Słowo *spam* (w jęz. polskim - mielonka) określa niepożądane wiadomości e-mail, które mogą się pojawiać w naszej skrzynce. Są to często różnego rodzaju reklamy i oferty handlowe. Aby zminimalizować ilość tego typu korespondencji należy:

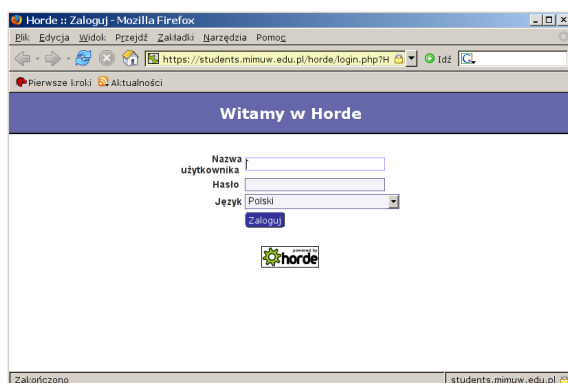
- Nie umieszczać swojego adresu e-mail na stronach WWW. Są firmy, które zajmują się pozyskiwaniem adresów ze stron WWW za pomocą odpowiedniego oprogramowania. Następnie, w ten sposób otrzymane adresy, są wykorzystywane do wysyłania wiadomości reklamowych.
- Podawać swój adres e-mail tylko tym osobom, które go rzeczywiście potrzebują
- Dowiedzieć się, czy na serwerze z którego korzystamy jest zainstalowane oprogramowanie antyspamowe. W obecnych czasach taka konfiguracja serwera powinna być standardem

5.4 Webmail

Oprogramowanie klienckie poczty elektronicznej takie jak Thunderbird lub Outlook Express, nie jest jedynym stosowanym do obsługi wiadomości elektronicznych. Popularnym obecnie sposobem korzystania z poczty jest tzw. Webmail.

Studencki serwer pocztowy *students.mimuw.edu.pl*, z którego korzystamy na tych zajęciach posiada zainstalowane oprogramowanie antywirusowe oraz antyspamowe.

5.4.1 Korzystanie z Webmaila



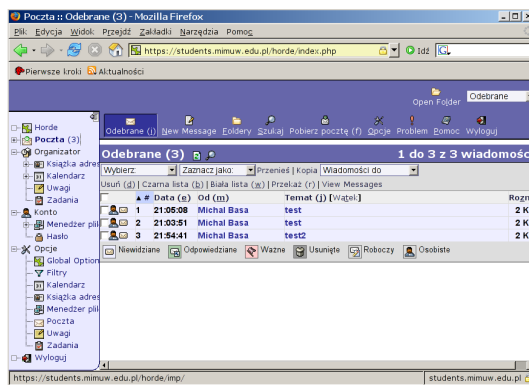
Webmail to obsługa konta pocztowego za pomocą przeglądarki internetowej:

- Nie jest potrzebne żadne dodatkowe oprogramowanie (np. Thunderbird)
- Trzeba się zalogować

Aby skorzystać z Webmaila, wystarczy znać adres URL, pod którym aplikacja ta jest dostępna. Do logowania zwykle używamy nazwy konta i hasła takich, jak do konfiguracji protokołu POP3 (pobierania poczty) w oprogramowaniu klienckim. Obecnie oprogramowanie Webmail stało się bardzo popularne, wielu administratorów serwerów pocztowych udostępnia je użytkownikom. Webmail jest (w pewnych okolicznościach) bardzo wygodny, gdyż możemy odbierać i wysyłać wiadomości jedynie za pomocą przeglądarki internetowej. Nie trzeba instalować/konfigurować żadnego dodatkowego oprogramowania. Jest to atutem chociażby w przypadku wyjazdu i konieczności obsługi poczty z kawiarenki internetowej.

Należy zaznaczyć, iż istnieje bardzo wiele aplikacji Webmail (lepszych i gorszych), ale zazwyczaj korzystanie z poczty za ich pomocą jest mniej wygodne niż przy użyciu *prawdziwego* oprogramowania klienckiego w rodzaju Thunderbirda. Zainstalujmy więc i skonfigurujmy klienta poczty, jeśli nie zachodzą okoliczności zmuszające nas do dostępu do skrzynki przez przeglądarkę.

5.4.2 Interfejs przykładowego Webmaila



Interfejs Webmaila jest zwykle zbliżony do tradycyjnego

- np. do interfejsu programu Thunderbird
- zazwyczaj jest dość prosty w obsłudze

Rozdział 6

Wstęp do arkuszy kalkulacyjnych

6.1 Wprowadzenie

Współczesny świat wymaga częstego przygotowywania prostych zestawień danych (np. spis zakupionego sprzętu wraz z policzoną łączną ceną, czy tygodniowy rozkład zajęć instruktorów narciarskich) oraz wykonywania mniej lub bardziej skomplikowanych obliczeń (wysokości poszczególnych rat spłacanego kredytu przy kwartalnej kapitalizacji odsetek czy choćby wysokość miesięcznej zaliczki na podatek dochodowy). Można te zadania wykonywać w wielu różnych narzędziach (np. dane można wpisać do edytora tekstu, a obliczenia wykonać na kalkulatorze, można użyć bazy danych albo napisać odpowiedni program, można wreszcie skorzystać z wyspecjalizowanego narzędzia obliczającego podatki). Okazuje się jednak, że bardzo duża część tego typu operacji nie wymaga całej potęgi baz danych czy języków programowania, a jednocześnie wymaga wsparcia komputerowego.

W ramach niniejszego kursu będziemy poznawać podstawowe zasady pracy z arkuszem kalkulacyjnym oraz zasady projektowania dobrych arkuszy.

Niniejszy moduł pozwoli *aktywnie studiującej* osobie osiąść umiejętności pozwalające na:

- tworzenie arkuszy kalkulacyjnych
 - w typowych współczesnych programach do obsługi arkuszy
- zapewnianie efektownej postaci arkusza
 - dzięki zastosowaniu różnorodnych rodzajów formatowania
- wbudowywanie w arkusz algorytmów przeliczania danych
 - wykorzystanie formuł
 - uaktualnianie przez arkusz wyników po każdej modyfikacji danych arkusza

- wykonywanie operacji na całych zestawach danych
 - takich jak np. sortowanie
- konstruowanie poprawnie zbudowanych arkuszy

6.2 Zaczynamy

6.2.1 Co to jest arkusz kalkulacyjny

Wiele obliczeń można wykonać na kartce papieru (np. można policzyć ile wyniosą opłaty za prąd i gaz przy przewidywanym zużyciu energii). Jednak zwykle okazuje się, że raz przeprowadzone obliczenia chciałoby się móc powtórnie wykonać dla nieco zmienionych danych (np. zauważamy, że popełniliśmy błąd przy wprowadzaniu danych o zużyciu gazu, albo chcemy zobaczyć o ile mniej zapłacimy, gdy zredukujemy zużycie prądu). Na tradycyjnej kartce musimy przeprowadzić wszystkie obliczenia od początku, co nawet gdy będziemy się wspomagać kalkulatorem nie jest wygodne, no i oczywiście może spowodować kolejne błędy.

- wielka pokratkowana płachta papieru
- pozwalająca kopiować/przesuwać swoją zawartość
- potrafiąca wykonywać obliczenia
- wykonująca automatycznie raz określone obliczenia
- także program obsługujący takie płachty

Gdyby tak można było raz opisać jak ma przebiegać obliczanie, a potem tylko zmieniać dane i od razu otrzymywać wyniki! A gdyby dodatkowo można było jeszcze ładnie sformatować zawartość płachty? Albo dodać wykres wyliczonych danych? A gdyby jeszcze ten wykres automatycznie się uaktualniał po każdej zmianie danych?

To wszystko, i jeszcze wiele więcej, możemy bez trudu osiągnąć stosując arkusz kalkulacyjny! Już po krótkim czasie używania tego narzędzia trudno jest uwierzyć, że dotąd jakoś można było się bez niego obyć.

6.2.2 Zalety arkuszy kalkulacyjnych

Mając arkusz kalkulacyjny nie musimy się martwić o szybkość i poprawność obliczeń — co jak co ale operacje na liczbach to prawdziwy żywioł komputerów!

- szybkość i bezbłądność obliczeń
- automatyczne przeliczanie i powtarzalność obliczeń

- łatwość tworzenia arkusza (kopiowanie formuł)
- wysoka jakość prezentacji danych
- łatwość modyfikowania (i możliwości jej ograniczania)

Tak jak już wspominaliśmy wcześniej arkusz pozwala nie tylko wykonać pewne obliczenia, ale także potrafi zapamiętać *co* ma liczyć (dokładniej — to my go uczymy jak *ma* liczyć) i wtedy gdy zajdzie potrzeba (czyli zmienia się dane) automatycznie dokonać ponownych obliczeń.

Arkusz uczymy tego *co* ma liczyć wpisując *formuły* (czyli wzory określające sposób wyliczania wyników). W typowym arkuszu potrzeba wpisać wiele formuł (np. kilkaset lub kilka tysięcy), na szczęście dzięki bardzo przemyślanemu mechanizmowi kopiowania zwykle wystarczy wpisać tylko kilka — kilkanaście formuł i potem je powielić. Dzięki temu praktycznie nie ma znaczenia, czy tworzy się arkusz mający kilka czy kilka tysięcy wierszy, nakład pracy w obu przypadkach jest praktycznie taki sam!

Dzięki bardzo rozbudowanym możliwościom formatowania (porównywalnym — jeśli nie przewyższającym — z tym co oferują edytory) można tworzyć bardzo efektowne arkusze, a dodatkowo można je uatrakcyjnić umieszczając w nich wykresy.

Ponieważ dane wpisujemy w komputerze a nie na kartce, to oczywiście możemy je bardzo łatwo w każdej chwili zmodyfikować (i natychmiast otrzymać uaktualnione wyniki). Jednak czasem (szczególnie w zastosowaniach finansowych) zależy nam na tym, by takie uaktualnianie nie było możliwe (np. arkusz z wynikami finansowymi firmy może być dostępny dla wielu osób, ale zmieniać w nim dane powinni móc tylko nieliczni, np. tylko księgowi). Takie zabezpieczanie arkusza przed zmianami daje się bardzo łatwo zdefiniować w arkuszu i może ono dotyczyć tak całego arkusza jak tylko wybranych jego fragmentów.

6.2.3 Struktura arkusza kalkulacyjnego

Wbrew pozorom plik arkusza kalkulacyjnego nie zawiera pojedynczego arkusza, lecz skoroszyt złożony z dowolnej (zwykle jest ich kilka) liczby arkuszy.

- *skoroszyt* (zeszyt) złożony z arkuszy
- *arkusz* składający się z *kolumn* i *wierszy*
- *komórki* leżące na przecięciu wierszy i kolumn
- arkusze mają nazwy (np. *Arkusz1*)
- kolumny mają identyfikatory (np. *C*)
- wiersze mają numery (np. *12*)

Czasem zamiast słowa *skoroszyt* używa się terminu *zeszyt*.

- komórki mają adresy (np. *C12*)

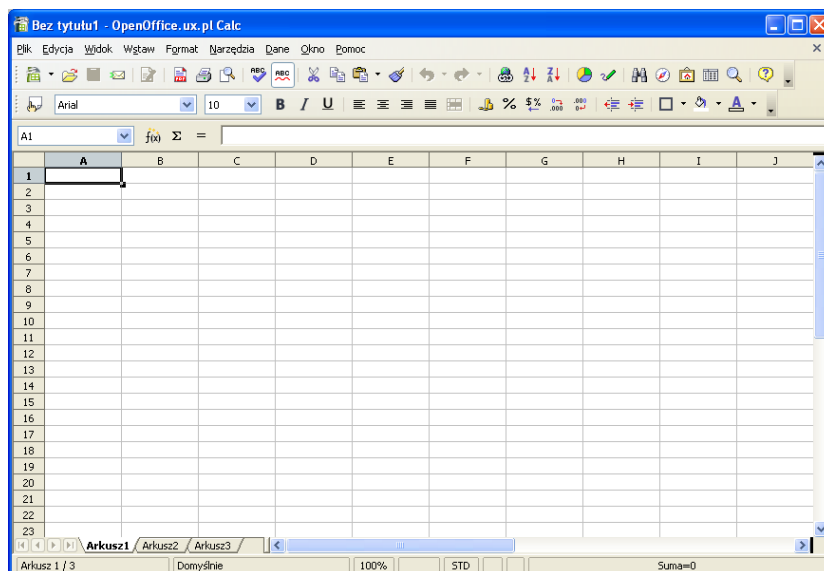
Sam arkusz jest pokratkowaną płachtą papieru, więc w naturalny sposób dzieli się na wiersze i kolumny, a składa się z poszczególnych komórek (odpowiadających kratkom na pokratkowanej płachcie papieru).

Arkusze mają nazwy, są nimi standardowo identyfikatory *Arkusz1*, *Arkusz2*, ... ale można oczywiście także samemu nadawać własne nazwy, jak np. *Podatki* albo *Wpływy za styczeń*.

Również kolumny i wiersze mają nazwy. W przypadku kolumn nazwami są litery (tylko alfabetu łacińskiego), po *Z* występują pary liter (*AA*, *AB*, ...). Zwykle arkusze mają 256 kolumn, więc ostatnia ma nazwę *IV* (ta nazwa nie ma oczywiście nic wspólnego z rzymską liczbą 4). Nazwami wierszy są kolejne liczby naturalne (pierwszy wiersz ma numer 1). Zwykle arkusze mają 65536 wierszy.

Elementem składowym arkusza są pojedyncze komórki, czyli przecięcia wierszy i kolumn. Ich nazwą (nazwy komórek będziemy określać słowem *adres*) jest nazwa kolumny, po której następuje nazwa wiersza (zatem pierwsza komórka w arkuszu ma nazwę *A1*). Jak łatwo policzyć (formułą w arkuszu oczywiście!), typowy arkusz ma 256*65536, czyli 16 777 216 (na szczęście nie musimy wszystkich wypełniać).

6.3 Wygląd arkusza kalkulacyjnego



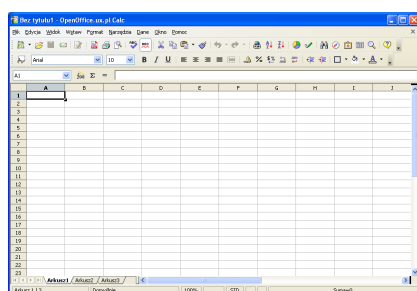
Różne programy-arkusze kalkulacyjne a nawet różne wersje tego samego programu inaczej wyglądają. Nie ma ani możliwości ani sensu by je wszystkie omawiać. Nie ma też znaczenia, czy w aktualnie używanej wersji arkusza ikonka przycisku podglądu wydruku ma jaśniejsze czy ciemniejsze tło. To co jest istotne to zasady organizacji, tworzenia i pracy z arkuszami kal-

kulacyjnymi, a nie to, którego narzędzia (czy której jego wersji) obecnie używamy.

Dlatego omówimy tylko ogólnie elementy interfejsu użytkownika, zwracając uwagę na to co się powtarza w różnych programach-arkuszach i bardzo starając się zaniedbywać wszystko to, co jest specyficzne dla konkretnego narzędzia.

Oczywiście, jeśli ktoś na co dzień pracuje z jakimś konkretnym programem, to zapamiętuje także te bardziej specyficzne udogodnienia oferowane przez używane narzędzie (co zresztą bardzo utrudnia potem przeniesienie się do innego produktu lub nawet nowszej wersji tego samego programu).

6.3.1 Elementy interfejsu użytkownika w typowym arkuszu



- menu
- paski narzędzi
- wiersz wprowadzania (więcej o nim dalej)
- płachta arkusza
- zakładki arkuszy
- pasek stanu

Typowe elementy interfejsu użytkownika to:

- *Menu*: obsługiwane dokładnie tak samo jak w każdym programie, dość rozbudowane.
- *Paski narzędzi*: jest ich co najmniej kilka, nie wszystkie są od razu wyświetlane, zwykle można nie tylko ustalać, które będą widoczne, ale także zmieniać ich zawartość.
- *Wiersz wprowadzania*: p. dalej.
- *Płachta arkusza*: najważniejsza część okna programu arkusza, to tu wpisujemy dane i oglądamy wyniki. Dzięki suwakom z boku i na dole możemy przeglądać całą zawartość płachty.

- *Zakładki arkuszy*: pozwalają wygodnie przechodzić między arkuszami (a także dodawać je i usuwać oraz zmieniać ich nazwy).
- *Pasek stanu*: wyświetla różne użyteczne informacje, np. sumę wartości z zaznaczonych komórek, czy informację o trybie wpisywania danych (z rozsuwaniem czy z zastępowaniem tekstu).

6.3.2 Wiersz wprowadzania w typowym arkuszu

- pole adresu
- kreator funkcji, suma, formuła
- anuluj, akceptuj
- pole formuły

6.3.3 Korzystanie z pomocy

Pomoc w arkuszu

- standardowa, kontekstowa
- menu Pomoc lub F1
- strzałka “Co to jest” z menu Pomoc

6.4 Podstawowe operacje

6.4.1 Pliki używane przez arkusz

- specjalny format (.ods, .sxc, .xls)
 - częściowo odczytywalne przez inne arkusze
- polecenia do zapisywania
 - *zapisz* i *zapisz jako*
- kopia bezpieczeństwa

Narzędzia ► Opcje ►

Ładuj/Zapisz ► Ogólne ► Zawsze wykonuj kopię zapasową

- zapisywanie z hasłem

Plik ► Zapisz jako ► Zapisz z hasłem

- polecenie wczytywania
- lista ostatnio zapisywanych arkuszy
- zamykanie pliku (a nie programu)
- automatyczne zapisywanie

Narzędzia ► Opcje ► Ładuj/Zapisz ►

Ogólne ► Zapisuj informacje automatycznego przywracania co

6.4.2 Wpisywanie danych do arkusza

Trzy rodzaje informacji związanych z komórką:

- zawartość
- wartość
- format

Rodzaje zawartości komórek:

- liczby
- napisy
- daty (tak na prawdę to są liczby)
- formuły

Zawartością komórki może być napis, liczba lub formuła.

- Liczby: 123, 123.45, -123.45, 50
- Daty i czas również są reprezentowane jako liczby. Przykłady: 92/3/23
2/3 3/95 18:23.
- Napis to dowolna zawartość komórki nie będąca liczbą ani formułą.
- Formuły będą omówione później.

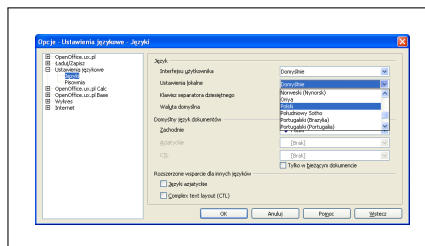
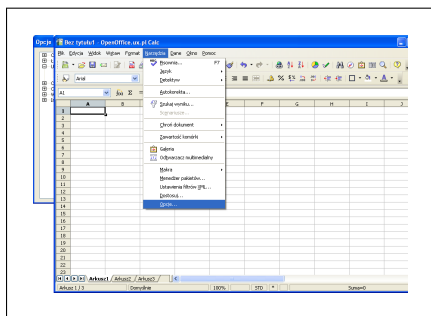
Uwaga sposób wyświetlania:

- separatora części ułamkowej (znak . lub ,),

- symbolu waluty (np. zł za liczbą lub \$ przed liczbą),
- dat i godzin (np. kolejność rok-miesiąc-dzień czy miesiąc-dzień-rok),

zwykle zależy od ustawień poczynionych w systemie operacyjnym.

Arkusze często pozwalają na zmienianie tych ustawień niezależnie od przyjętych w systemie operacyjnym. Np. w arkuszu *Calc* polecenie *Narzędzia* ► *Opcje* ► *Ustawienia językowe* ► *Języki* pozwala wybrać czy chcemy stosować ustawienia domyślne, czy ustawienia stosowane w wybranym kraju.



Przejsięcie do ustawień programu

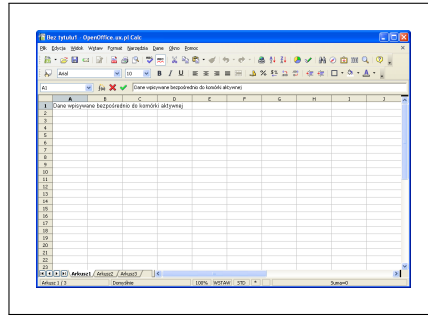
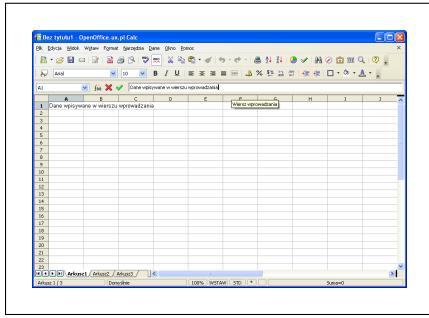
Wybór ustawień językowych

Jeśli napis nie mieści się w całości w komórce, to jest wyświetlany także w komórkach sąsiednich (o ile są puste) lub jest obcinany. Jeśli natomiast liczba nie mieści się w komórce, to wyświetlany jest ciąg znaków #. Ma to bardzo rozsądne wyjaśnienie: liczba bez kilku cyfr dalej wygląda na poprawną liczbę i wprowadzałaby w błąd czytających arkusz, zaś obcięcie napisu (np. w połowie słowa) zwykle jest łatwe do zauważenia przez czytelników.

O ile tego nie zmienimy (patrz dalej) liczby są dosuwane do prawej strony, zaś napisy do lewej.

Wprowadzenie danych

Dane zawsze są wprowadzane do komórki aktywnej. Można je wpisywać albo w wierszu wprowadzania, albo bezpośrednio w komórce aktywnej. Tak samo jest z modyfikowaniem zawartości komórek — można to robić w wierszu wprowadzania lub bezpośrednio w samej komórce. W każdym przypadku należy najpierw uczynić tę komórkę aktywną.



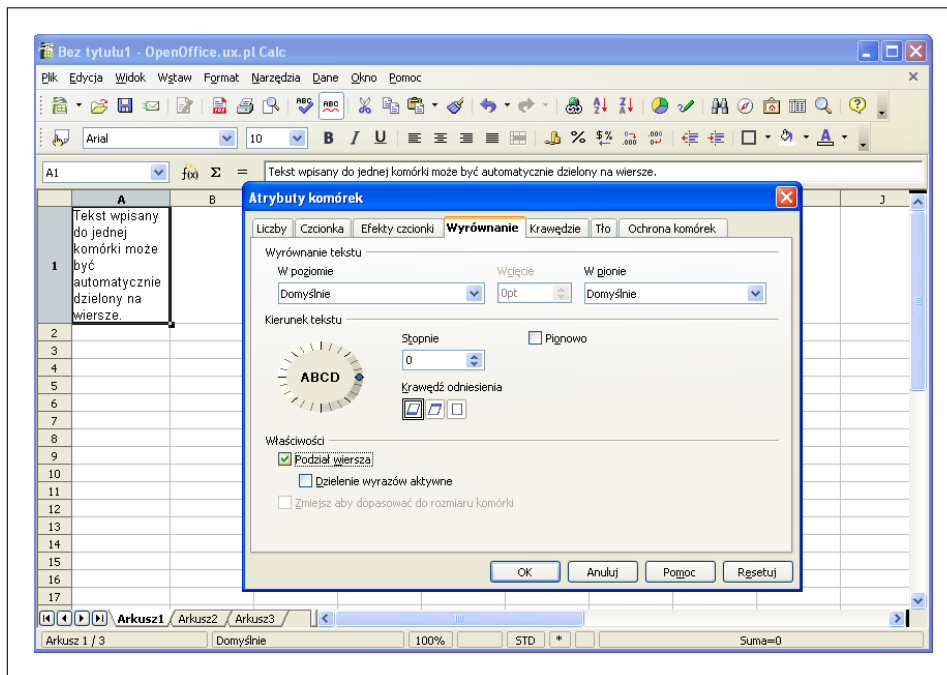
Dane można wpisywać i poprawiać zarówno w wierszu wprowadzania ...

... jak i w komórce aktywnej.

Wielowierszowy tekst w komórkach

- **Zawijanie tekstu w komórce**

Arkusze kalkulacyjne potrafią automatycznie zawijać tekst wpisany do komórek. Domyślnie ta własność jest wyłączona. Jeśli chcemy z niej skorzystać w jakichś komórkach, to musimy najpierw te komórki zaznaczyć, następnie wydać polecenie *Format* ► *Komórki* ► *Wyrównanie* i zaznaczyć właściwość *Podział wiersza*.

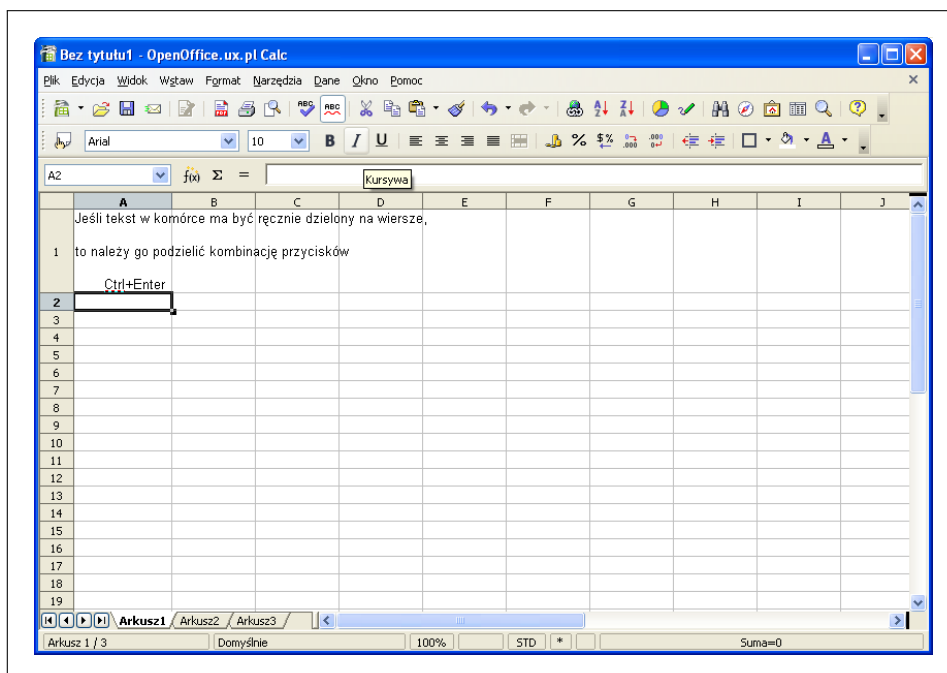


Tekst wpisany do komórki może być automatycznie dzielony na wiersze

- **Ręczne łamanie wierszy**

Możemy sami wskazać miejsca, gdzie wpisywany tekst ma być po-

dzielony na wiersze. Należy w tym celu wpisując tekst do komórki wprowadzić w miejscu podziału kombinację klawiszy *Ctrl+Enter*.



Tekst wpisany do komórki można ręcznie dzielić na wiersze

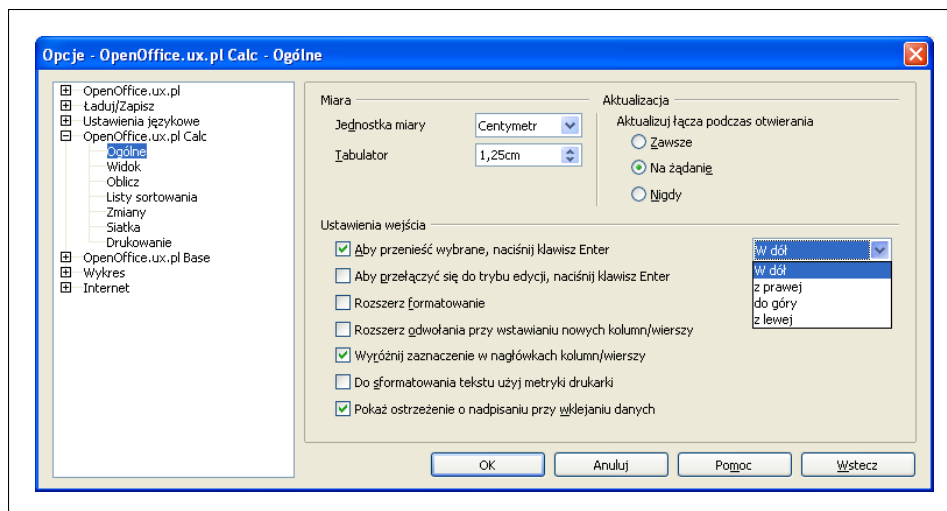
Ciekawostki

Polecenie *Format* ► *Komórki* ► *Wyrównanie* ► *Zmniejsz aby dopasować do rozmiaru komórki* powoduje, że arkusz automatycznie zmniejsza wyświetlaną wielkość czcionki zastosowanej w komórce tak, by — przy niezmienionej szerokości kolumny — tekst nie musiał być łamany na wiersze.

6.4.3 Nawigowanie po arkuszu

- *Komórka aktywna*
- myszka i suwaki
- klawisze strzałek
- przechodzenie na początek i koniec wiersza/kolumny lub całego arkusza
- przeskakiwanie wypełnionych obszarów
- wyszukiwanie (i zastępowanie)

Nawigowanie pomiędzy komórkami można realizować na wiele sposobów (wymienionych na ekranie). Można też ponadto spowodować, by naciśnięcie klawisza *Enter* przesuwało do innej komórki. W arkuszu *Calc* można wybrać to czy klawisz *Enter* będzie przesuwał aktywną komórkę oraz to w jakim kierunku. Wykonuje się to za pomocą polecenia *Narzędzia ► Opcje ► OpenOffice.ux.pl Calc ► Ogólne*.



Określenie sposobu działania klawisza Enter

6.4.4 Obszary arkusza

- *obszar*
- zaznaczanie obszaru
- adres obszaru (np. A3:B14)
- obszary wieloczęściowe
- kolejność czynności przy wykonywaniu poleceń na obszarach

Wskazywanie (zaznaczanie) obszarów

- przeciąganie myszką,
- można też wskazać jeden róg obszaru, a następnie, trzymając wciśnięty klawisz Shift, kliknąć myszką w przeciwległy róg obszaru,
- F5 lub *Edycja ► Nawigator*.

Wybrany obszar jest podświetlony. Można zaznaczać całe wiersze lub kolumny, grupy wierszy lub kolumn, cały arkusz. Można też zaznaczać obszary składające się z wielu prostokątów (wymaga to wciśnięcia klawisza Ctrl) nawet na kilku arkuszach. Jeśli jest zaznaczony obszar, to Tab przesuwa

komórkę aktywną w prawo/dół w zaznaczonym obszarze, zaś Shift/Tab w lewo/górze.

Jeśli chcemy wykonać jakąś czynność z menu dla pewnego obszaru arkusza, to najpierw zaznaczamy (podświetlamy) ten obszar, a dopiero potem wybieramy właściwe polecenie z menu.

6.5 Formatowanie arkusza

6.5.1 Operacje na wierszach i kolumnach

- wstawienie i usuwanie wierszy
- wstawienie i usuwanie kolumn
- wstawianie i usuwanie obszarów
- przesuwanie wierszy i kolumn
- formuły a operacje na fragmentach arkusza

Arkusz można formatować, co oznacza to że można:

- wstawiać i usuwać wiersze, kolumny, obszary (*Wstaw ► Wiersze*, *Wstaw ► Kolumny* oraz *Edycja ► Usuń*),

6.5.2 Formatowanie arkusza — komórki

- formaty liczb
- ramki
- wyrównywanie tekstu w komórkach
- formaty warunkowe

W arkuszu można także:

- zmieniać szerokość kolumn i wysokość wierszy (mysz lub podwójne kliknięcie lub *Format ► Wiersz ► Wysokość*, *Format ► Kolumna ► Szerokość*),
- zmieniać sposób wyświetlania liczb (liczby, procenty, waluta, data, itp.) (*Format ► Komórki ► Liczby*). Uwagi:
 - liczby w miarę potrzeby są zaokrąglane, a nie obcinane,
 - w obliczeniach używa się prawdziwej wartości, a nie wyświetlanej,

- uwaga na zaokrąglenia, mogą istotnie zmienić wymowę arkusza: np. 0.49 i 0.50 z jednym miejscem po przecinku i bez miejsc po przecinku,
- usuwać zawartość komórki: Ctrl/Del lub mały czarny kwadracik w prawym dolnym rogu lub *Edycja* ► *Wyczyść* lub Del (przedostania komenda pozwala także usunąć formatowanie),
- zmieniać krój pisma, wysokość liter, styl kroju (pogrubianie, kursywa), kolor liter (pasek narzędzi lub *Format* ► *Komórki* ► *Czcionka*),
- zmieniać dosuwanie wewnątrz komórki: w lewo, w prawo, na środek (pasek narzędzi lub *Format* ► *Komórki* ► *Wyrównanie*),
- wyśrodkowywać tekst w obrębie całego obszaru (*Format* ► *Komórki* ► *Wyrównanie* lub pasek narzędzi),
- wyświetlać kilka wierszy w jednej komórce (*Format* ► *Komórki* ► *Wyrównanie*). Można w obrębie komórki ręcznie łamać tekst: Alt/Enter,
- rysować ramki dookoła komórek (*Format* ► *Komórki* ► *Krawędzie*) lub pasek narzędzi,
- zmieniać kolor i wzór tła komórek (*Format* ► *Komórki* ► *Wzorki* lub pasek narzędzi),
- kazać arkuszowi automatycznie sformatować fragment arkusza (*Format* ► *Autoformat*),
- definiować własne style komórek (*Format* ► *Styl*),
- wyrównywać tekst w zadanym obszarze arkusza (*Edycja* ► *Wypełnij* ► *Wyjustuj*),
- zabezpieczać zawartość komórek przed modyfikacjami (patrz zabezpieczanie arkusza).

6.5.3 Ćwiczenie z formatowania arkusza

Przygotowanie tabelki ze spisem towarów w sklepie komputerowym. Kolumny:

- LP
- Nazwa
- Cena netto
- Waluta
- Cena netto w zł
- Stopa VAT
- VAT

- Cena brutto
- Liczba sztuk
- Łączna wartość brutto (wszystkich sztuk)

Arkusz powinien zawierać:

- nagłówek nad tabelką (Spis towarów sklepu komputerowego)
- nagłówki poszczególnych kolumn
- wiersz z podsumowaniami

Na razie nie przejmujemy się wyliczaniem, wpisujemy w kolumny dowolne liczby (żeby było widać ich formatowanie). Uwzględniamy: formaty liczb, kolory i czcionki, style.

6.5.4 Kopiowanie danych w arkuszu

- bardzo ważne
- kopiowanie i wstawianie można wykonywać standardowo (przez schowek)
- przy wklejaniu można wybrać, co ma być wklejone:
 - zawartości
 - formaty
 - wartości
 - wszystko

Fragmenty arkusza można kopiować:

- *Edycja* ► *Kopiuj* potem *Edycja* ► *Wklej* i Enter (lub przeciągnięcie myszą z wciśniętym klawiszem Ctrl lub prawym przyciskiem lub pasek narzędzi),
- *Edycja* ► *Kopiuj* potem *Edycja* ► *Wklej Specjalnie* (lub pasek narzędzi) np. kopiowanie tylko formatu lub zawartości,
- *Edycja* ► *Kopiuj* potem *Wstaw* ► *Skopiowane komórki* (lub przeciągnięcie z Ctrl i z Shift) łączenie kopiowania ze wstawianiem.

Kopiowanie w arkuszu kalkulacyjnym ma dużo większe znaczenie niż np. w edytorze (w szczególności ze względu na różne tryby adresowania). Kopiując obszar można jako miejsce przeznaczenia wskazać:

- obszar takiego samego kształtu i rozmiaru,

- lewy górny róg obszaru docelowego,
- obszar mieszczący w sobie całkowitą wielokrotność kopii obszaru ukopiowanego (powielanie), w szczególności kopiowanie komórki w obszar oznacza jej powielanie,
- kolumnę gdy kopiujemy wiersz (powielanie wiersza),
- wiersz gdy kopiujemy kolumnę (powielanie kolumny).

Obszar kopiowany i docelowy nie muszą być rozłączne. Można powielać komórkę, wiersz lub kolumnę tak jak przy tworzeniu serii. Można kopiować cały obszar arkusza (więcej się wtedy kopiuje — także szerokości wierszy i wysokości kolumn).

6.5.5 Przenoszenie danych w arkuszu

- przenoszenie formuł nie zmienia adresów w przenoszonych formułach (w przeciwieństwie do kopiowania)
- wykonujemy je standardowo
- dane nie znikają z arkusza zaraz po wycięciu do schowka (dopiero po wklejeniu)

Można przenosić fragmenty arkusza:

- *Edycja* ► *Wytnij* potem *Edycja* ► *Wklej* (lub przeciąganie myszą lub pasek narzędzi),
- *Edycja* ► *Wytnij* potem *Wstaw* ► *Wycięte komórki* (lub przeciąganie z Shift). Jeśli przesuwamy całe kolumny/wiersze będą one usunięte, a nie tylko ich zawartości.

Wskazywanie obszaru docelowego:

- obszar identycznego rozmiaru i kształtu,
- lewy górny róg.

6.6 Użyteczne dodatki

6.6.1 Tworzenie serii danych

- wygodny mechanizm o różnorodnych zastosowaniach
- wskazanie pierwszej komórki i rozciągnięcie
- wskazanie dwu pierwszych komórek i rozciągnięcie (daje dużo większe możliwości)

- rodzaje serii (liczbowe (w tym daty), nazwy dni tygodnia, ...)

Tworzenie serii danych pozwala na bardzo wygodne wpisywanie do arkusza ciągów arytmetycznych (i nie tylko). Najbardziej typowym przykładem jest ciąg kolejnych liczb, zaczynający się od wartości 1 (zwykle kolumna LP w tabelce).

Zasadniczo ciągi wartości dotyczą tylko liczb, ale arkusz kalkulacyjny pozwala też tworzyć ciągi innych niż liczby wartości. Jeśli do komórki wpisujemy napis kończący się liczbą, to arkusz pozwoli nam na wygenerowanie ciągu napisów kończących się kolejnymi liczbami (przećwiczmy to na przykład na napisie “Kwartał 1”).

Ponieważ daty również są przechowywane w arkuszu jako liczby, to możemy łatwo stworzyć ciąg kolejnych dat, albo dat odległych od siebie np. o tydzień (warto spróbować wygenerować ciąg np. kolejnych piątków). Takie właściwości generowania ciągów okazują się bardzo praktyczne przy generowaniu różnorodnych zestawień.

Ciągi kolejnych wartości można też generować za pomocą formuł. Ma to pewne zalety w stosunku do tworzenia serii danych, ale wymaga znajomości formuł i wymaga trochę więcej pracy, więc serie danych są bardzo często stosowane.

6.6.2 Zabezpieczanie arkusza

- czemu takie ważne
- zabezpieczanie poszczególnych komórek
- włączanie systemu zabezpieczeń w arkuszu
- można podać hasło

Zabezpieczanie arkusza ma niesłychanie duże znaczenie. W arkuszach często (a może nawet zwykle) przeprowadzamy obliczenia, od których poprawności wiele zależy (ocena końcowa ucznia, wysokość podatku jaki zapłacimy w urzędzie skarbowym, itp.). Zwykle przeprowadzamy te obliczenia dlatego, że same nie jesteśmy w stanie przeprowadzić ich w pamięci, zwykle też wynikami są liczby. Jesteśmy więc zdani na uwierzenie w prawdziwość wyników podanych przez arkusz, przy czym prawidłowy wynik (liczba) niewiele się różni od nieprawidłowego (też liczby). Zauważmy teraz, że jeśli w typowym arkuszu mającym kilkaset (lub więcej) wierszy i kilkadziesiąt (lub więcej) kolumn w jednej z komórek wpisujemy przypadkiem zamiast formuły liczbę (a wystarczy w tym celu na przykład na chwilę zamyślić się przy klawiaturze i łokciem nacisnąć część numeryczną klawiatury), to dostaniemy pozornie tak samo wyglądający arkusz, tyle że obliczający zupełnie złe wyniki. I zupełnie nie będzie widać miejsca gdzie jest błąd!

Na tradycyjnej płachcie papieru każda poprawka była od razu widoczna (widac było skreślenie i wpisaną nową wartość). W elektronicznym arkuszu

tak nie jest i dlatego musimy umieć zabezpieczyć się przed przypadkowymi modyfikacjami zawartości arkusza.

Przy okazji (ustalając hasła) zabezpieczymy się (w pewnym stopniu), przed złośliwymi modyfikacjami, które mogły by chcieć wprowadzić do naszego arkusza inne osoby.

Do zabezpieczania wybranych fragmentów arkusza przed modyfikacją służą dwa polecenia:

- *Format ► Komórki ► Ochrona* — wskazywanie obszarów, które mają być zabezpieczone (odbezpieczone),
- *Narzędzia ► Zabezpieczenie ► Zabezpiecz arkusz* — uaktywnienie systemu zabezpieczającego uprzednio wskazane obszary przed modyfikacją.

Uwaga: początkowo wszystkie komórki są zabezpieczone przed modyfikacją, ale system zabezpieczania nie jest aktywny.

Można zabezpieczyć cały skoroszyt przed odczytaniem przez nieuprawnione osoby (*Plik ► Zachowaj jako ► Opcje ► Hasło Zabezpieczające*). Hasło do 15 znaków, rozróżnia się wielkie i małe litery. Można też zabezpieczyć skoroszyt przed zapisywaniem przez nieuprawnione osoby (*Plik ► Zachowaj jako ► Opcje ► Hasło do zapisu*). Można wreszcie sugerować każdemu otwierającemu skoroszyt, by otwierał go w trybie tylko do czytania (*Plik ► Zachowaj jako ► Opcje ► Tylko do odczytu*).

6.6.3 Praca z dużymi arkuszami

- problem: arkusze są duże i nie mieszczą się na ekranie ani na kartce
- (częściowe) rozwiązania:
 - zamrażanie części arkusza na ekranie
 - dzielenie okna na części
 - otwieranie kilku okien z tym samym arkuszem
 - zmiana skali wyświetlanego arkusza
 - odpowiednie parametry drukowania

Ponieważ praktycznie żaden arkusz nie mieści się na ekranie (poza niektórymi przykładami z kursów) musimy jakoś zadbać o sensowne wykorzystanie powierzchni ekranu (i papieru przy drukowaniu).

- zamrażanie wierszy i kolumn z nagłówkami (*Okno ► Zablokuj okienka*),
- podział okna dokumentu na części (*Okno ► Podziel okno lub mysz*),
- otwieranie kilku okien z jednym dokumentem,

- zmiana skali wyświetlanego arkusza,
- patrz też parametry drukowania.

Rozdział 7

Arkusz zaczyna działać

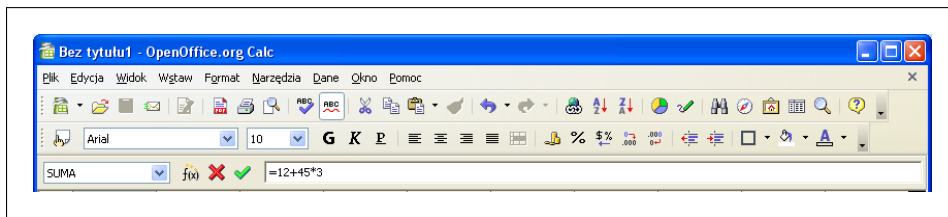
7.1 Obliczenia w arkuszu — pojęcia podstawowe

7.1.1 Formuły

Największą zaletą arkusza kalkulacyjnego jest to, że można go “nauczyć” jak przeprowadzać potrzebne nam obliczenia. Uczenie polega na wpisaniu do arkusza wzorów, które opisują sposób uzyskiwania pożądaných rezultatów. Takie wzory wpisywane do arkusza nazywamy formułami. To co stanowi o sile formuł w arkuszu to po pierwsze fakt, że mogą one odwoływać się do komórek arkusza, a o drugie umiejętność arkusza wyliczania poszczególnych formuł wtedy, gdy ich wartości mogły się zmienić, czyli gdy zmieniły się wartości komórek, do których odwołuje się formuła.

- Co to jest formuła
- Składnia formuł
- Kreator formuł

Składnia formuł jest bardzo prosta — przypomina wzory pisane na lekcjach matematyki. Można w nich stosować zamieszczać liczby i adresy komórek, można stosować cztery podstawowe działania i można wreszcie wywoływać w nich liczne zdefiniowane w arkuszu funkcje (na przykład funkcję sumującą wartości swoich argumentów). Specyficzną dla arkusza cechą formuł jest to, że muszą się one zaczynać od znaku “=”. Dzięki niemu arkusz rozpoznaje czy ma do czynienia z formułą (której wartość musi wyliczyć), czy napisem (który musi tylko wypisać). Tak wygląda przykładowa formuła:



Przykładowa formuła

Formuły można wpisywać tak w pasku formuł jak i w samych komórkach. Dużym ułatwieniem dla początkujących jest kreator formuł (wywołuje się go klikając w symbol funkcji znajdujący się na pasku formuł). Kreator formuł prowadzi krok po kroku przez wpisywanie formuły, podpowiadając jakie można wpisać parametry wywoływanych funkcji.

7.1.2 Zasady pisania formuł

Reguły dotyczące tworzenia arkuszy kalkulacyjnych:

- (prawie) nigdy nie wpisujemy liczb do formuł — parametryzacja arkusza
- nigdy nie wpisujemy tej samej liczby dwa razy do arkusza — nie dublujemy danych
- nigdy nie umieszczamy w jednej kolumnie tabelki różnych danych (np. różnie obliczanych)
- liczby w jednej kolumnie formatujemy, tak by miały tę samą liczbę miejsc po przecinku

7.1.3 Tryby adresowania

- tryb względny
- tryb bezwzględny
- tryb mieszany
- tryby dotyczą tylko operacji kopiowania (przenoszenia już nie)

7.1.4 Ćwiczenia z trybów adresowania

- Stwórz tabliczkę mnożenia 10 na 10.
 - pierwszy wiersz i kolumnę (z kolejnymi liczbami) stwórz jako serie danych
 - wpisz do pierwszej wolnej komórki w pierwszej kolumnie formułę, która po skopiowaniu do pozostałych komórek w tej kolumnie policzy prawidłowe iloczyny

- spróbuj wymyślić formułę (jedną), która po powieleniu (skopowaniu) policzy wszystkie iloczyny w całej tabelce (wskazówka: jakiego trybu adresowania należy użyć?).
- Uzupełnij kolumny przykładu ze sklepem o formuły wyliczające wartości

7.1.5 Operacje na datach

- jak wpisywać
- jak są reprezentowane
- jak dostać bieżącą datę
- jak policzyć odstęp między dwiema datami
- funkcja DATA (uwaga na zmianę wieku)

7.1.6 Definiowanie nazw

- czemu warto definiować nazwy
- różne sposoby definiowania
- odwoływanie się do nazw obszarów w formułach
- niebezpieczeństwa związane z nazwami

7.2 Najpopularniejsze funkcje z arkusza kalkulacyjnego

7.2.1 Funkcja SUMA

- witamy mistrza
- składnia: SUMA(liczba 1; liczba 2; ...)
- przycisk sumujący
- przypomnienie o wyświetlaniu bieżącej sumy
- uwagi o sumowaniu nie-liczb

7.2.2 Funkcja JEŻELI

- witamy wice-mistrza
- składnia: JEŻELI(test_logiczny; wartość_jeżeli_prawda; wartość_jeżeli_fałsz)
- pojedyncza funkcja JEŻELI

- zagnieżdżanie funkcji JEŻELI
- kolumnowe wykorzystanie funkcji JEŻELI

7.2.3 Ćwiczenie z funkcjami SUMA i JEŻELI

Mając tabelę z cenami z poprzednich ćwiczeń:

- policz sumy cen brutto i VATu
- policz osobno sumę cen VATu 0 (zastosuj dla każdej stawki VAT dodatkową kolumnę z formułą używającą funkcji JEŻELI)

7.2.4 Proste funkcje statystyczne

- wartości minimalne i maksymalne: MIN i MAX
- składnia: MIN(liczba 1; liczba 2; ...)
- MAX(liczba 1; liczba 2; ...)
- liczenie średniej: ŚREDNIA
- składnia: ŚREDNIA(liczba 1; liczba 2; ...)

7.2.5 Funkcje zliczające

- zliczanie proste liczb: ILE.LICZB
- składnia: ILE.LICZB(liczba 1; liczba 2; ...)
- zliczanie proste wypełnionych komórek: ILE.NIEPUSTYCH
- składnia: ILE.NIEPUSTYCH(liczba 1; liczba 2; ...)
- zliczanie ciekawe: LICZ.JEŻELI
- składnia: LICZ.JEŻELI(Zakres; Kryteria)
- sumowanie ciekawe: SUMA.JEŻELI
- składnia: SUMA.JEŻELI(Zakres; Kryteria; zakres_sumowania)

7.2.6 Funkcja ZAOKR

- zaokrąglanie: funkcja ZAOKR
- składnia: ZAOKR(liczba; liczba_cyfr)
- różnica między zaokrąglaniem a zmienianiem formatowania
- błędy zaokrągleń (zaokrąglaj najpóźniej jak tylko się da)
- rodzaje zaokrągleń

7.2.7 Funkcje wyszukiujące

- wyszukiwanie: funkcja WYSZUKAJ.PIONOWO
- składnia: WYSZUKAJ.PIONOWO(Kryterium_wyszukiwania; Macierz; Indeks; Porządek_sortowania)
- wyszukiwanie względem widełek
- wyszukiwanie konkretnych wartości
- wyszukiwanie w poziomie

7.2.8 Ćwiczenie z funkcją WYSZUKAJ.PIONOWO

- dodaj do arkusza tabelkę z kursami walut, których użyłeś w arkuszu, zadбай o właściwą kolejność kolumn i wierszy w tej tabelce (pierwsza kolumna z oznaczeniami walut, posortowana rosnąco)
- wpisz do kolumny Cena netto w złotych formułę wyliczającą tę cenę na podstawie nazwy waluty i ceny w tej walucie (zastosuj wyszukiwanie w pionie)

7.2.9 Proste funkcje tekstowe

- długość tekstu: funkcja DŁ
- składnia: DŁ(Tekst)
- wycinanie fragmentów tekstu: funkcje LEWY, PRAWY, MID
- składnia: LEWY(Tekst; Liczba)
- PRAWY(Tekst; Liczba)
- MID(Tekst; Liczba_początkowa; Liczba)
- wyszukiwanie tekstu w tekście: funkcja SZUKAJ.TEKST
- składnia: SZUKAJ.TEKST(Znajdź_tekst; Tekst; Pozycja)
- formatowanie liczb: funkcja TEKST
- składnia: TEKST(Liczba; Format)

7.3 Tworzenie arkusza - podsumowanie

7.3.1 Dodatkowe uwagi o pisaniu formuł

- nie wpisuj adresów — wskazuj je
- nie wpisuj nazw — wybieraj je z listy

- pisz nazwy funkcji z małej litery
- korzystaj z kreatora funkcji
- pamiętaj, że możesz kopiować i przenosić części formuł
- możesz wyliczać części formuły (podświetlenie fragmentu formuły i klawisz *F9*)
- pamiętaj o podwójnym kliknięciu przy zaznaczaniu elementów formuły

7.3.2 Błędy obliczeń sygnalizowane przez arkusz

- #DZIEL/0!
- #LICZBA!
- #NAZWA?
- #ZERO!
- #ARG!
- #ADR!
- #N/D!

7.3.3 Pętle adresowe

Czasami tworząc arkusze doprowadzamy do powstania w nich pętli adresowych. *Pętla adresowa to zestaw komórek z formułami, które wzajemnie się do siebie odwołują, tak że wyliczenie wartości każdej z tych komórek odwołuje się pośrednio lub bezpośrednio do niej samej.*

- *co to jest*
- *pętle bezpośrednie i pośrednie*
- *wykrywanie i usuwanie*

Najprostszy przykład takiej pętli to komórka, która zawiera formułę odwołującą się do niej samej, na przykład formuła “=A1+1” wpisana do komórki A1. Taką pętlę nazywamy pętlą bezpośrednią. Pętle adresowe mogą być oczywiście bardziej skomplikowane, na przykład komórka A1 może zawierać formułę “=B1+1”, zaś komórka B1 może z kolei zawierać formułę “=A1+1”. W takim przypadku mówimy o pętli pośredniej.

Występowanie pętli w arkuszu jest szkodliwe, gdyż uniemożliwia obliczenie wartości komórek występujących w pętli (pomijamy tu pewne specyficzne

techniki przeprowadzania obliczeń w arkuszach). Dlatego arkusze sygnalizują wystąpienie pętli adresowej wypisaniem komunikatu o błędzie w komórkach oraz na pasku stanu (czasami używają w tym celu sformułowania “odwołanie cykliczne”).

Dużą pomocą przy wykrywaniu tych błędów — oraz ogólnie przy analizie zawartości arkusza — jest polecenie Narzędzia ►Detektyw. Po jego wybraniu mamy do dyspozycji kilka poleceń służących do graficznego (za pomocą strzałek) wskazywania zależności między danymi w arkuszu. Jest to niezwykle cenne. W szczególności gdy nie jesteśmy pewni jak wygląda pętla adresowa, którą wykryliśmy w naszym arkuszu, możemy przejść do dowolnej komórki z komunikatem o błędzie i wydać polecenie Narzędzia ►Detektyw ►Śledź błędy. W odpowiedzi arkusz wyświetli nam za pomocą strzałek cykliczne powiązania ze wskazanej pętli adresowej.

Gdy już rozwiążemy problem i będziemy chcieli objrzeć arkusz w normalnej postaci (bez strzałek), to będziemy mogli wydać polecenie Narzędzia ►Detektyw ►Usuń wszystkie strzałki.

7.3.4 Etapy tworzenia arkusza kalkulacyjnego

- wpisujemy nagłówki wierszy i kolumn tabeli
- wpisujemy przykładowe dane
- nazywamy kolumny (wiersze) tabeli
- wpisujemy do pierwszego wiersza (kolumny) formuły
- kopiujemy formuły do reszty tabeli
- formatujemy tabelkę
- zabezpieczamy przed zapisem obszar nagłówek i formuł
- uzupełniamy dane

7.3.5 Ćwiczenie z dużym przykładem

- stwórz arkusz z ocenami uczniów jednej klasy, który dla każdego ucznia policzy średnią z ocen cząstkowych i dla każdego ucznia na podstawie tej średniej (potrzebna będzie pomocnicza tabelka progów) wystawi każdemu uczniowi słowną ocenę (bardzo dobry, dobry, dostateczny, ...)
- stwórz tabelkę wyliczającą miesięczną pensję pracownika uwzględniając składki na różnorodne ubezpieczenia społeczne, zaliczkę na podatek (dodaj kolumnę z sumą dochodów z poprzednich miesięcy), dodatek stażowy i potrącenie składki na ubezpieczenie na życie dla tych pracowników, którzy sobie tego życzą (kolejna kolumna, tym razem z zawartością np. “TAK” lub “NIE”).

7.4 Operacje bazodanowe

7.4.1 Sortowanie

- *obszar sortowania*
- *sortowanie proste (przyciski)*
- *sortowanie złożone (polecenie)*
- *jak uzyskać pożądaną kolejność danych przy wielu kryteriach*
- *wsortowywanie nagłówków*
- *jak przywrócić oryginalną kolejność*
- *zachowanie formuł podczas sortowania*

7.4.2 Filtrowanie

- *co to jest*
- *czy zmienia zawartość arkusza*
- *filtrowanie automatyczne*
- *filtrowanie zaawansowane*

7.4.3 Ćwiczenie z sortowaniem i filtrowaniem

- *posortuj dane z tabelki z danymi o cenach w sklepie komputerowym wg rosnących cen a następnie posortuj je alfabetycznie wg nazw*
- *w tej samej tabelce zastosuj automatyczne filtrowanie tak by pokazać tylko produkty o cenie wyrażonej w Euro, a potem zastosuj filtr pokazujący jedynie towary o cenie brutto większej niż 100 zł.*

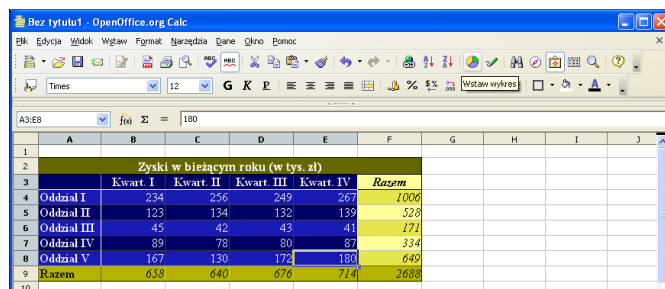
Rozdział 8

Dalsze operacje na arkuszach

8.1 Wykresy w arkuszu kalkulacyjnym

8.1.1 Tworzenie wykresów za pomocą kreatora

Uruchamianie kreatora wykresów



Zyski w bieżącym roku (w tys. zł)					
	Kwart I	Kwart II	Kwart III	Kwart IV	Razem
Oddział I	234	256	249	267	1006
Oddział II	123	134	132	139	528
Oddział III	45	42	43	41	171
Oddział IV	89	78	80	87	334
Oddział V	167	130	172	180	649
Razem	638	640	676	714	2688

- *Kreator wykresów*
- *Serie danych*
- *Właściwe zaznaczanie obszarów danych*

Kreator wykresów to narzędzie dostępne we współczesnych arkuszach kalkulacyjnych, służące do poprowadzenia użytkownika za rękę przez proces tworzenia arkusza.

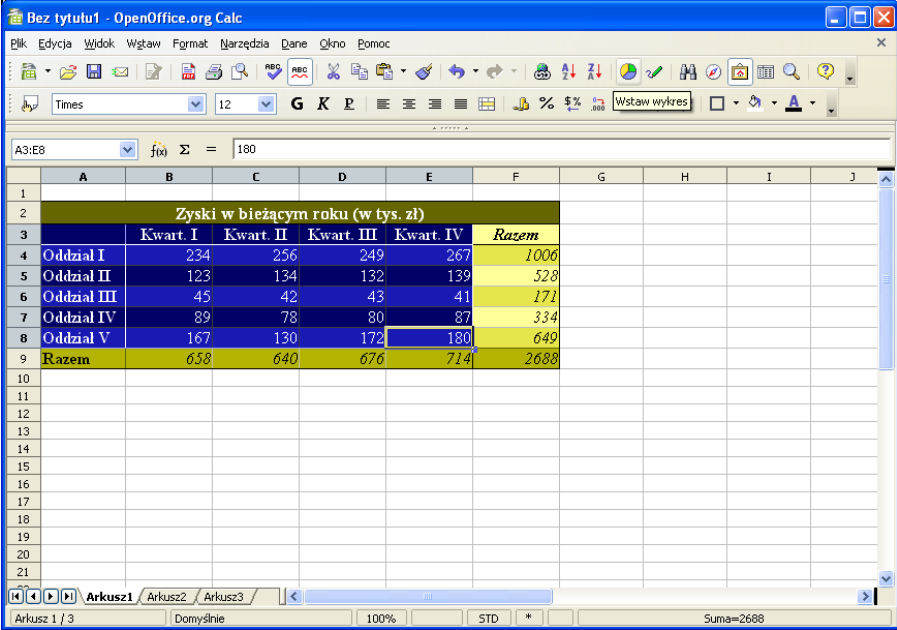
Przy tworzeniu wykresu należy wskazać dane, które mają się znaleźć na wykresie. Najczęściej okazuje się, że na wykresie chcemy pokazać kilka zestawów danych w celu ich porównania ze sobą (np. dane o populacji w kilka państwach). Pojedynczy zestaw danych nazywa się serią danych. Mając prostokątną tabelkę z danymi (np. licznosc populacji kilku państw w kilku kolejnych latach) możemy zdecydować co traktujemy jako jedna serię danych (czy dane o jednym państwie czy o jednym roku) i w ten sposób wypuklać na wykresie różne aspekty tych samych danych (np. zmiany w latach czy różnice pomiędzy państwami).

Tworząc wykres można:

- najpierw wywołać kreatora, a potem wskazać dane,
- wskazać pojedynczą komórkę w obrębie tabeli, a potem wywołać kreatora, który sam ustali obszar danych do przedstawienia na wykresie,
- wskazać obszar danych, a potem wywołać kreatora.

Zdecydowanie najlepsze jest to ostatnie rozwiązanie (wygodniej zaznaczyć obszar przed wywołaniem kreatora, zaś obszar wybrany przez kreatora rzadko będzie właściwy (np. w załączonym przykładzie będzie obejmował zarówno nagłówek tabeli jak i kolumny razem).

Zaznaczając obszar danych należy wskazać także nagłówki wierszy i kolumn, które kreator umieści przy opisie osi wykresu i w legendzie. Załączony rysunek pokazuje prawidłowe (kolor niebieski) zaznaczenie danych.



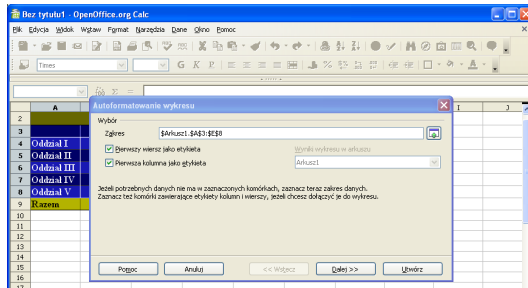
	A	B	C	D	E	F	G	H	I	J	
1											
2		Zyski w bieżącym roku (w tys. zł)									
3		Kwart. I	Kwart. II	Kwart. III	Kwart. IV	Razem					
4	Oddział I	234	256	249	267	1006					
5	Oddział II	123	134	132	139	528					
6	Oddział III	45	42	43	41	171					
7	Oddział IV	89	78	80	87	334					
8	Oddział V	167	130	172	180	649					
9	Razem	658	640	676	714	2688					
10											
11											
12											
13											
14											
15											
16											
17											
18											
19											
20											
21											

Uruchamianie kreatora wykresów

8.1.2 Miejsce na wykres

Po wywołaniu kreatora musimy wskazać miejsca na arkuszu, gdzie będzie się znajdował wykres (oczywiście można je później zmieniać). Robimy to myszką (wskaźnik myszki przybiera w tym czasie charakterystyczny kształt).

Pierwsze okienko kreatora wykresów

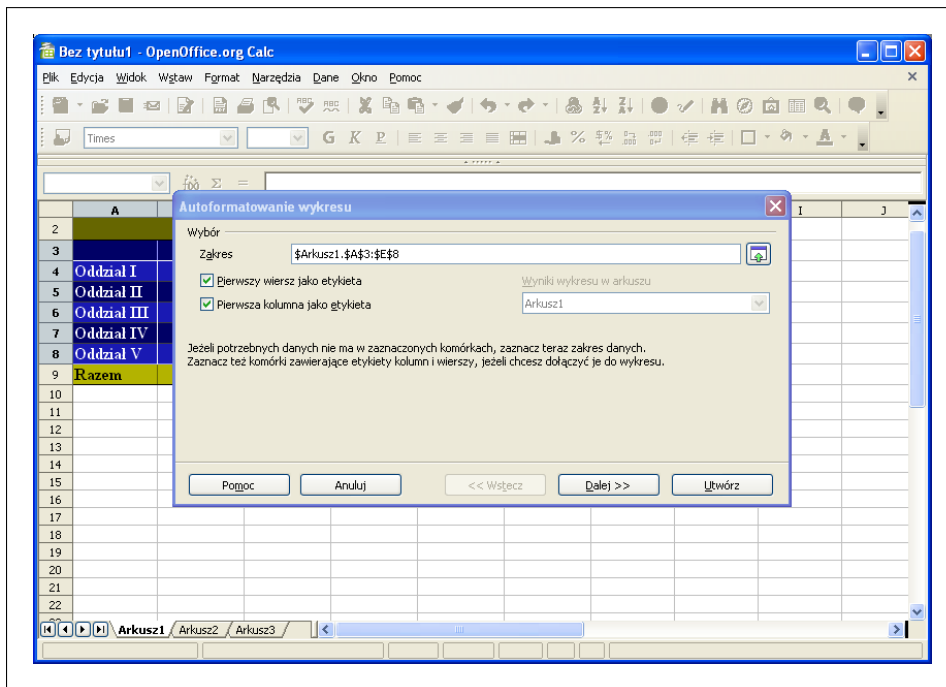


Pierwszy krok kreatora:

- *miejsce na wykres*
- *zakres danych*
- *wiersz/kolumna etykiet*
- *przyciski “dalej” ale i “wstecz”*

Jeśli poprawnie zaznaczyliśmy obszar danych przed wywołaniem kreatora, to w pierwszym okienku kreatora nie musimy nic robić poza kliknięciem w przycisk “dalej”. Pamiętajmy żeby wskazać wraz z danymi wiersze nagłówek danych i upewnić się, że są zaznaczone pola “pierwszy/wiersz/kolumna jako etykieta”.

Zwróćmy uwagę na to, że okna kreatora mają nie tylko przycisk “dalej” ale i “wstecz”, co pozwala wrócić i poprawić wcześniejsze ustalenia, o ile nie były właściwe. To ważne, bo dalsze okna umożliwiają podgląd gotowego wykresu, a co za tym idzie pozwalają dostrzec błędy popełnione we wcześniejszych etapach.

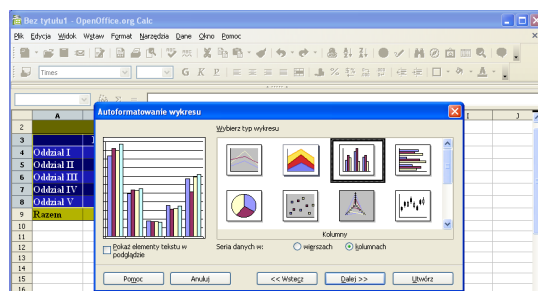


Pierwsze okienko kreatora wykresów

8.1.3 Typ wykresu

W drugim kroku pracy z kreatorem wybieramy przede wszystkim typ wykresu. Typ wykresu to sposób przedstawienia danych. Niektóre typy wykresów pasują tylko do pewnych rodzajów danych (np. kołowy tylko do danych z jedną serią).

Drugie okienko kreatora wykresów:



Drugi krok kreatora:

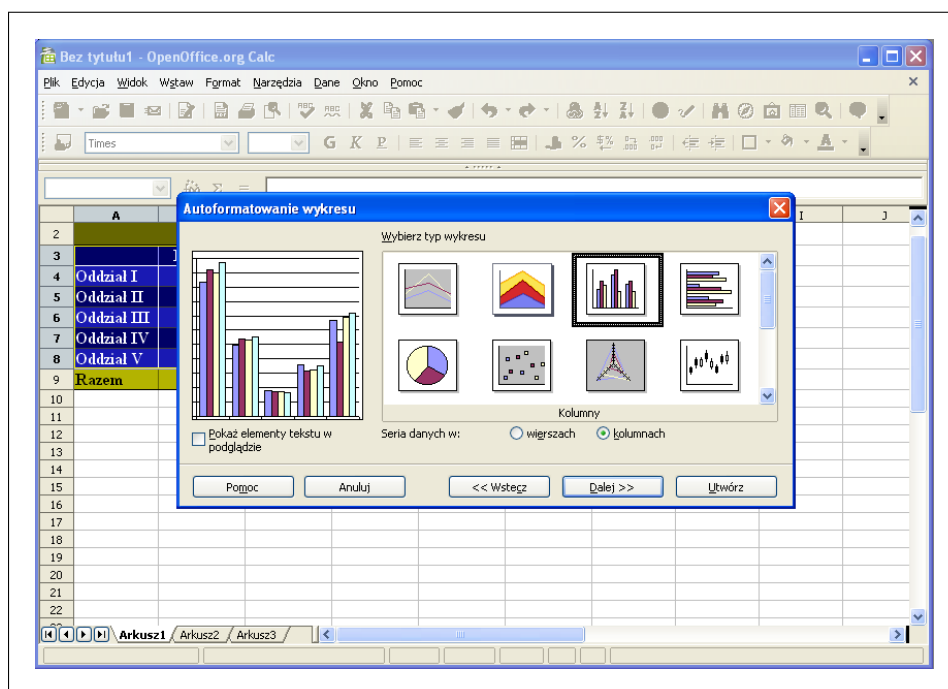
- *typy wykresów*
- *ułożenie serii danych*
- *etykiety na wykresie*

Współczesne arkusze udostępniają wiele typów wykresów (każdy w wielu wariantach) więc nie jesteśmy w stanie wszystkich tu omówić. Warto zapamiętać najczęściej stosowane typy wykresów, czyli:

- kolumnowy
- słupkowy
- kołowy

W prezentowanym przykładzie skorzystamy z wykresu kolumnowego.

W tym kroku tworzenia wykresu mamy też możliwość wskazania ułożenia serii danych. W naszym przykładzie przyjmujemy, że są one ułożone w kolumnach, czyli jedna seria danych dotyczy jednego kwartału (zatem chcemy na wykresie uwypuklić różnice pomiędzy wynikami poszczególnych oddziałów naszej firmy w kolejnych kwartałach, a nie różnice pomiędzy poszczególnymi kwartałami).

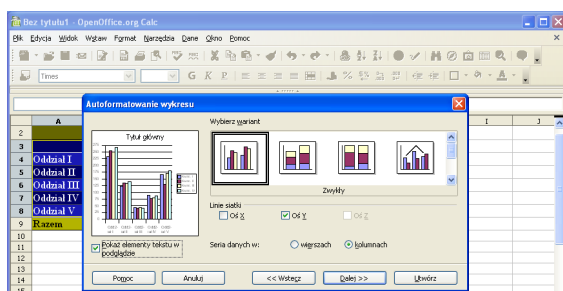


Drugie okienko kreatora wykresów

8.1.4 Format wykresu

W trzecim kroku pracy z kreatorem wybieramy wariant wykresu. Każdy typ ma kilka wariantów (podtypów). Na przykład dla wykresu kolumnowego można zdecydować, czy poszczególne kolumny mają być pokazywane obok siebie, czy jedna na drugiej (jest to poręczne np. wtedy, gdy chcemy pokazać, jak wyglądał w poszczególnych kwartałach procentowy udział poszczególnych oddziałów w wypracowaniu łącznego zysku).

Trzecie okienko kreatora wykresów



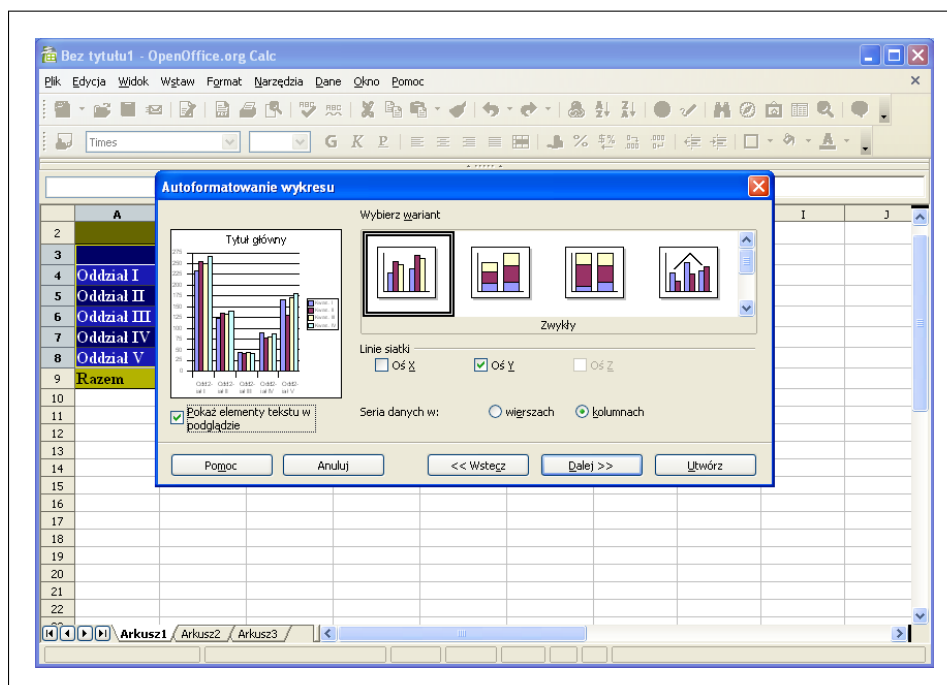
Trzeci krok kreatora:

- wariant wykresu
- widoczne linie siatki
- widoczne elementy tekstowe
- ułożenie serii danych (raz jeszcze)

Linie siatki pozwalają lepiej opisać dane znajdujące się na wykresie.

Warto włączyć polecenie “Pokaż elementy tekstu w podglądzie”, bo dzięki temu zobaczymy jak nasz wykres będzie wyglądał w pełnej okazałości.

Mamy ponownie możliwość zmienić ułożenie serii danych. Teraz jest to zdecydowanie prostsze, bo widzimy wynikowy wykres.

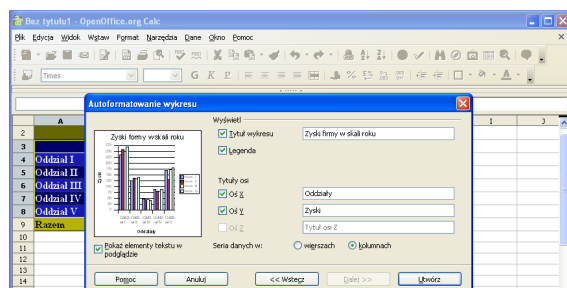


Trzecie okienko kreatora wykresów

8.1.5 Opisy wykresu

W ostatnim (czwartym) kroku pracy z kreatorem ustalamy tytuł wykresu oraz tytuły poszczególnych osi. Decydujemy także, czy ma być pokazana legenda.

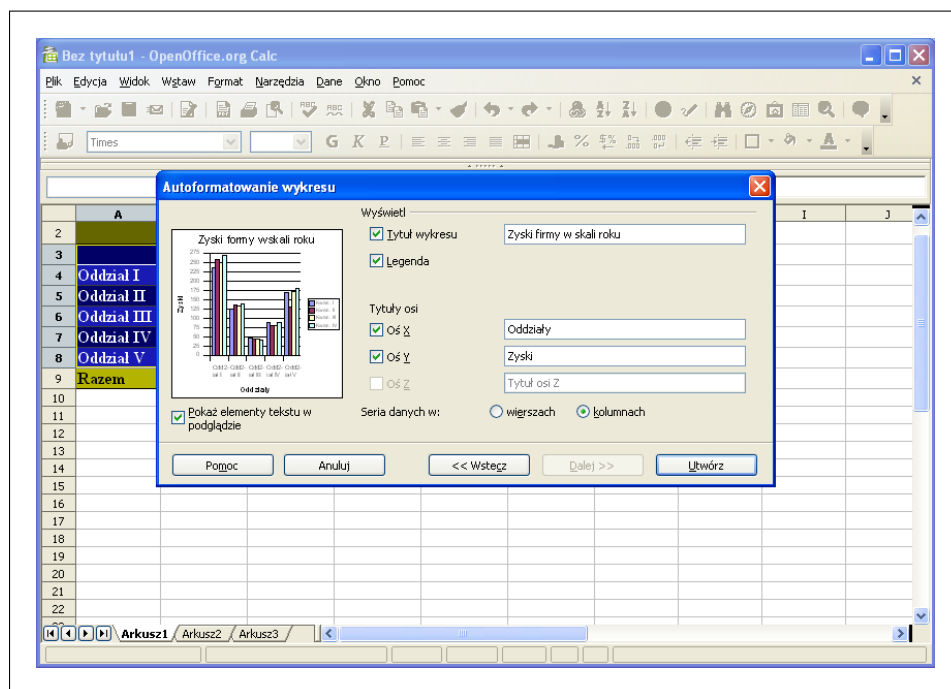
Czwarte i ostatnie okienko kreatora wykresów



Czwarty i ostatni krok kreatora:

- tytuły wykresu i osi
- legenda
- widoczne elementy tekstowe (ponownie)
- ułożenie serii danych (ponownie)

Ponownie mamy możliwość włączenia lub wyłączenia pokazywania elementów tekstowych oraz wyboru ułożenia serii danych.

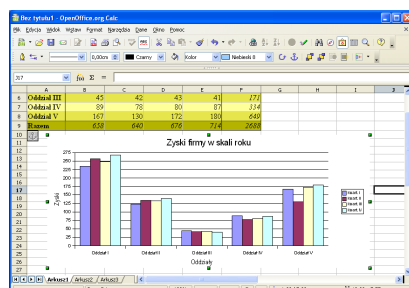


Czwarte okienko kreatora wykresów

8.1.6 Wykres gotowy!

Po przejściu wszystkich kroków z kreatorem wykresów mamy w arkuszu gotowy wykres.

Gotowy wykres



8.1.7 Tworzenie wykresów — co dalej?

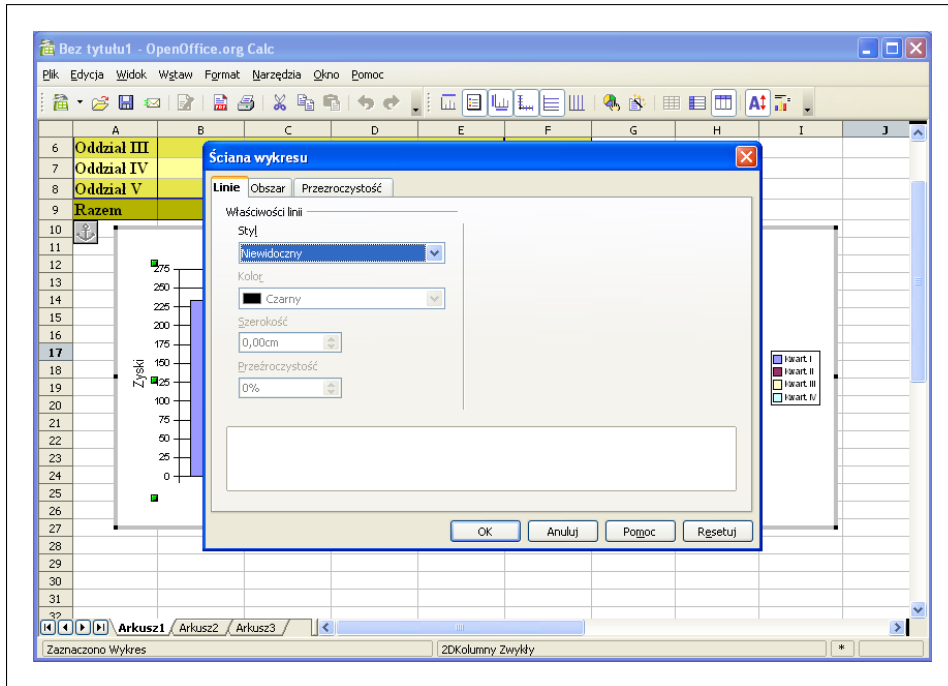
Oczywiście na stworzeniu wykresu możliwości pracy z nim się nie kończą. Po pierwsze — wykres jest żywy, nie jest wklejonym statycznym rysunkiem. Jeśli w arkuszu zmienią się dane, to automatycznie zmieni się także wykres! To bardzo cenna cecha wykresów w arkuszach kalkulacyjnych. Gdy raz je dobrze przygotujemy, będą cały czas pokazywać faktyczny stan danych w arkuszu.

Co można robić z gotowym wykresem

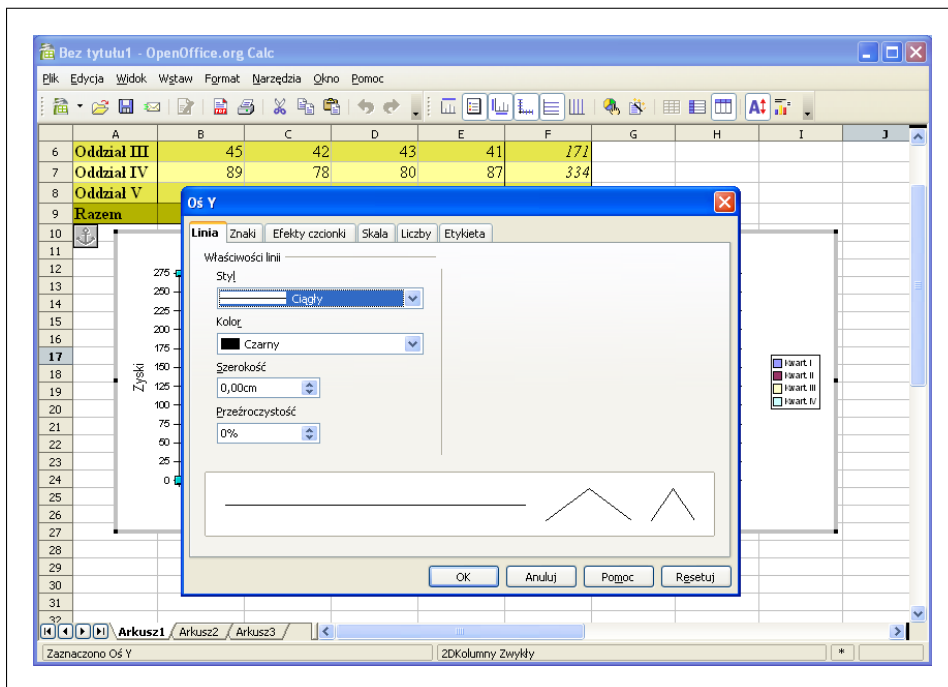
- zmieniać obrazowane na nim dane!
- przesuwać go w arkuszu
- zmieniać jego rozmiar
- edytować jego elementy (linie siatki, osie, obszar wykresu)
- usuwać

Po utworzeniu wykresu może się zdarzyć, że będziemy chcieli go zmodyfikować (najczęściej będzie to dotyczyło rozmiaru i położenia wykresu). Możemy to zawsze zrobić (w standardowy sposób myszką).

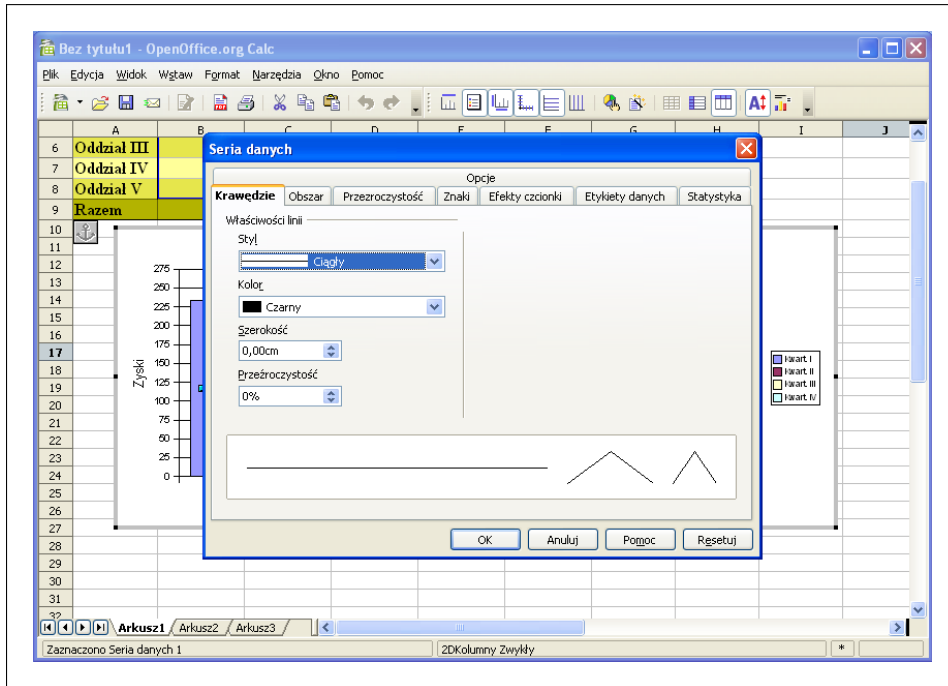
Możemy przeprowadzać bardziej zaawansowane modyfikacje wyglądu wykresu, ich pełny opis wykracza poza ramy tej pracy, tu zamieszczamy tylko kilka zrzutów ekranu ilustrujących niektóre możliwości edycji gotowych wykresów.



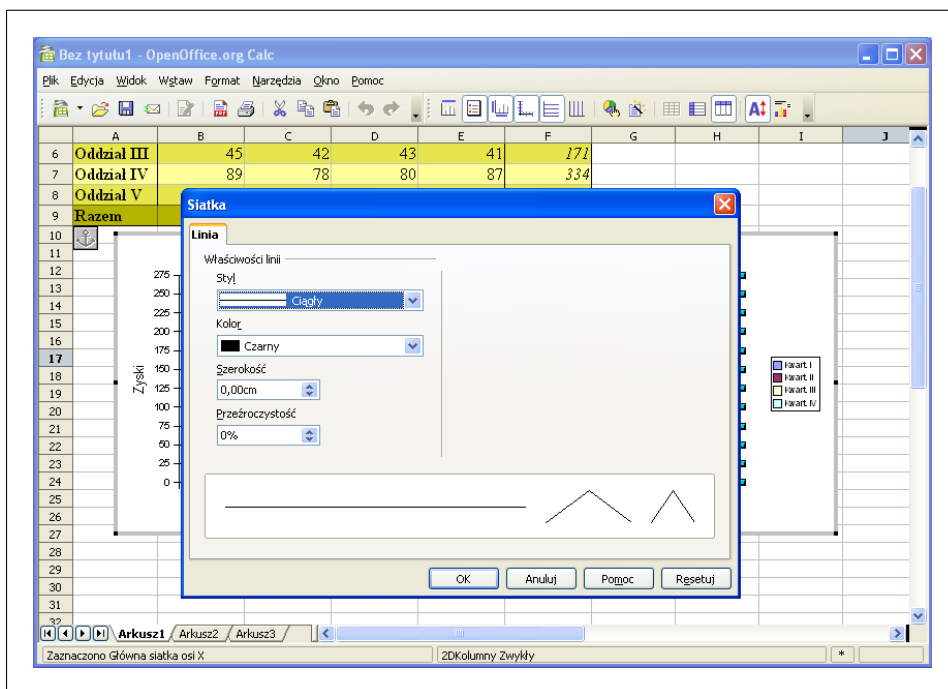
Edycja obszaru wykresu



Edycja osi wykresu



Edycja serii danych wykresu



Edycja linii siatki wykresu

8.1.8 Ćwiczenie z tworzenia wykresu

Stwórz wykres danych używanych we wcześniejszych ćwiczeniach:

- przede wszystkim zastanów się, **co** chcesz pokazać na wykresie (te same dane można pokazać na różne sposoby, uwypuklając różne ich aspekty)
- zaznacz zakres danych jeszcze przed wywołaniem kreatora
- wykonaj poszczególne kroki kreatora

8.2 Drukowanie arkusza kalkulacyjnego

8.2.1 Drukowanie arkusza

Naturalnym podsumowaniem pracy z arkuszem kalkulacyjnym jest wydrukowanie go. Można to zrobić za pomocą jednego przycisku. Ale nie warto — żeby wydruk miał sens należy nad nim trochę popracować. Szczegóły są omówione dalej, w tym punkcie koncentrujemy się na podstawach.

- Podgląd wydruku



- Obszar drukowania
- Drukowanie



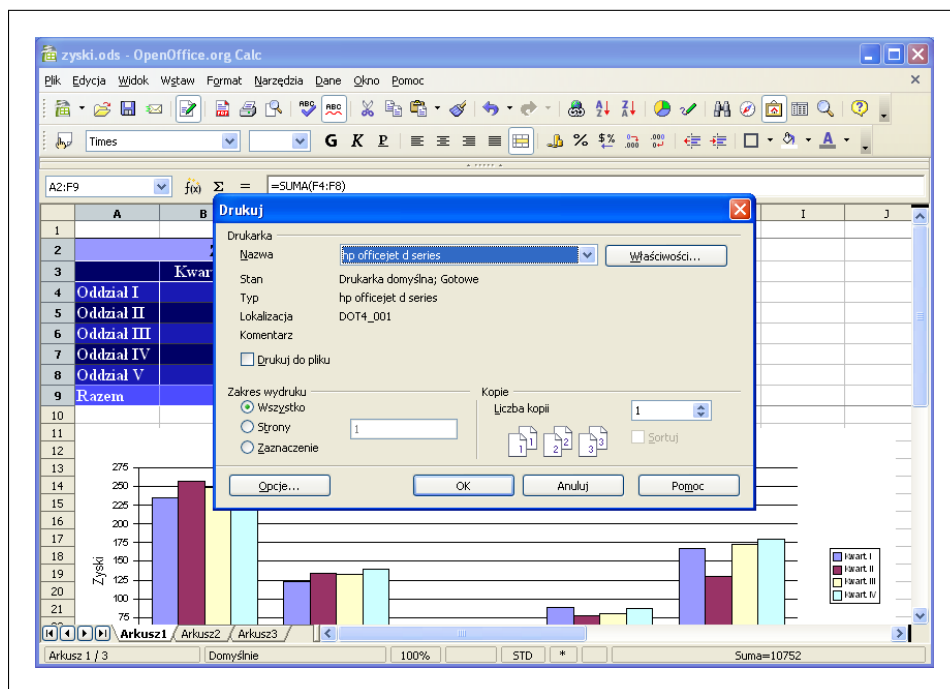
Drukowanie wywołujemy odpowiednim przyciskiem lub poleceniem z menu

Plik ► Drukuj

Nie należy jednak tego robić bez uprzedniego ustalenia obszaru wydruku. Standardowo arkusz drukuje się cały (cała wykorzystana jego powierzchnia, od komórki A1 do ostatniej kolumny zawierającej dane i do ostatniego wiersza z danymi). Często chcemy na wydruku pokazać tylko fragment arkusza (np. bez dodatkowych opisów, albo pomocniczych tabel). Obszar wydruku ustalamy zaznaczając wybrany obszar, a następnie wykonując polecenie **Format ► Zakresy wydruku ► Definiuj**. Oczywiście zawsze można zmienić obszar wydruku (definiując go ponownie), czy usunąć poleceniem: **Format ► Zakresy wydruku ► Usuń** (oczywiście usunięcie obszaru wydruku usuwa tylko definicję obszaru, a nie dane z arkusza).

Najważniejszym pewnie poleceniem związanym z drukowaniem jest podgląd wydruku — pozwala on zaoszczędzić tony papieru! Zanim coś wydrukujemy,

zobaczymy najpierw jak to będzie wyglądać na drukarce. Zwykle okaże się, że źle! Dlaczego? Dlatego, że arkusze są zwykle bardzo duże i nijak nie mieszczą się na kartce papieru. Dlatego zanim arkusz uda się sensownie wydrukować, trzeba nad nim popracować i sprawdzać na ekranie jak się prezentują kolejne wersje.

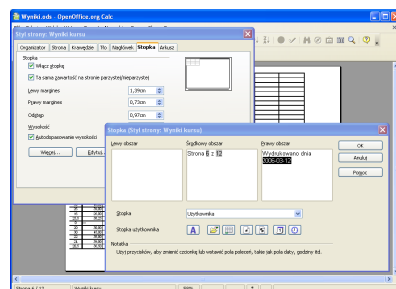


Okno polecenia drukowania

8.2.2 Nagłówki i stopki

Aby wielostronicowy wydruk miał sens (np. żeby można było go ponownie złożyć, gdy kartki rozsypią się na ziemię), każda strona musi być opisana nazwą dokumentu, numerem strony i łączną liczbą stron oraz datą wydruku. Wszystkie te (i jeszcze inne) elementy możemy łatwo umieścić zarówno w nagłówku jak i w stopce dokumentu. Nagłówek i stopkę definiujemy niezależnie jedno od drugiego.

Ustalanie treści stopki



- Nagłówki i stopki ustawia się niezależnie
- Polecenie *Format ► Strona ► Stopka ► Edytuj*
- Możliwość wyboru predefiniowanych postaci
- Podział na trzy części
- Gotowe elementy (liczba stron, data, ...)

Do definiowania stopki (nagłówka) służą polecenia: *Format ► Strona ► Stopka ► Edytuj* i *Format ► Strona ► Nagłówek ► Edytuj*. Gdy nagłówek lub stopka jest już zdefiniowany można przejść do jego edycji także poleceniem *Edycja ► Nagłówki i stopki*

Współczesne arkusze są na tyle przyjazne, że mają kilka z góry zdefiniowanych wersji nagłówka i stopki, możemy wybrać jedną z nich lub zdefiniować własne.

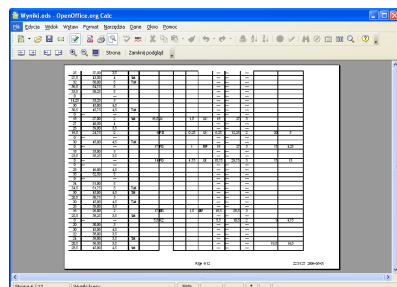
Treść nagłówka i stopki jest podzielona na trzy części (lewą, środkową i prawą), co pozwala wygodnie zdefiniować rozłożenie informacji w nagłówku/stopce.

Żeby jeszcze bardziej ułatwić tworzenie nagłówka/stopki współczesne arkusze takie informacje jak data wydruku czy liczba stron arkusza wstawiają do tworzonego przez nas tekstu automatycznie (wystarczy tylko przy tworzeniu nagłówka/stopki wskazać, gdzie taka informacja ma się pojawić).

8.2.3 Zamrażanie nagłówków kolumn i wierszy

Duża tabelka (a to jest zwykle zawartość arkusza) nie mieści się na jednej czy dwu stronach. Jeśli po prostu wydrukujemy ją na wielu stronach, to dane ze środka tabelki pojawią się na stronie bez nagłówków wierszy i kolumn. Dostaniemy zestaw wielu nic nie mówiących liczb. Perspektywa sklejania ze sobą wielu stron arkusza w jedną gigantyczną płachtę, raczej nie wzbudza entuzjazmu.

Przykład wydruku bez zamrożonych nagłówków

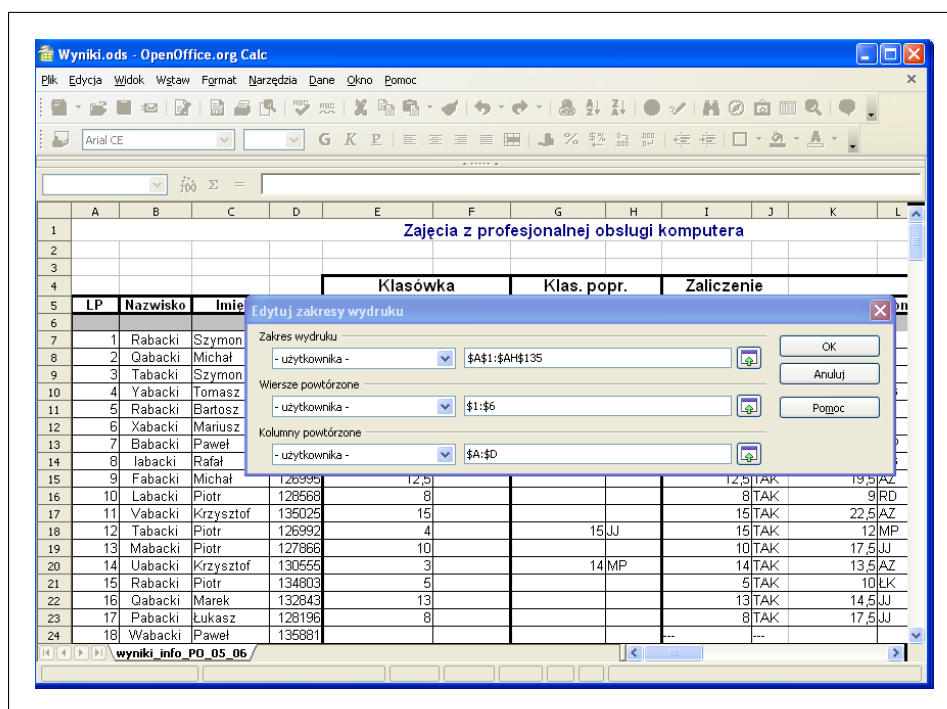


- Konieczne przy wydrukach wielostronicowych
- Podobne w działaniu do zamrożenia (przytwierdzenia) części okna

- Można zamrażać tylko wiersze albo tylko kolumny

Na szczęście można zrobić coś dużo bardziej sensownego — kazać na każdej stronie drukować wiersze i kolumny z nagłówkami. Robimy to za pomocą poleceń **Format ► Zakresy wydruku ► Edytuj ► Wiersze powtórzone i Format ► Zakresy wydruku ► Edytuj ► Kolumny powtórzone**

Przy wskazywaniu wierszy/kolumn do powtarzania warto korzystać z przycisku chwilowo zmniejszającego okienko — dzięki temu widzimy większą część arkusza. Wskazujemy obszar arkusza, ale arkusz zapamięta tylko numery wierszy bądź kolumn. Oto przykład wybrania nagłówków wierszy i kolumn powtarzanych na każdej stronie wydruku:



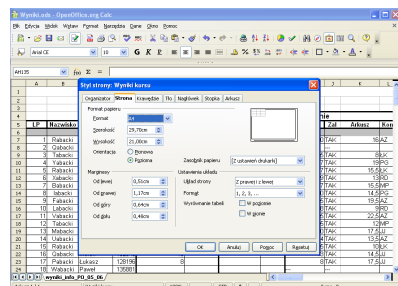
Wskazywanie nagłówków wierszy i kolumn powtarzanych na każdej stronie wydruku

8.2.4 Dopasowywanie rozmiarów wydruku

Ponieważ arkusz jest (prawie zawsze) większy niż strona wydruku, powstaje pytanie jak najlepiej rozmieścić go na kartkach wydruku. Pierwszą rzeczą jaką można zrobić jest odpowiednie zorientowanie stron wydruku. Przy pracy z edytorem tekstu właściwie zawsze drukuje się na kartkach ustawionych pionowo (portret). W przypadku arkusza jest inaczej. Zwykle drukuje się jedną, dużą tabelę, której poszczególne wiersze zawierają dane o jakiś obiektach (pracownikach firmy, książkach w bibliotece, towarach w magazynie). Dlatego bardziej zależy nam na tym, aby na jednej kartce mieć jak najwięcej danych z jednego wiersza (o jednym obiekcie), a nie z jednej kolumny (ten

sam aspekt wielu obiektów).

Ustawienie orientacji strony i marginesów

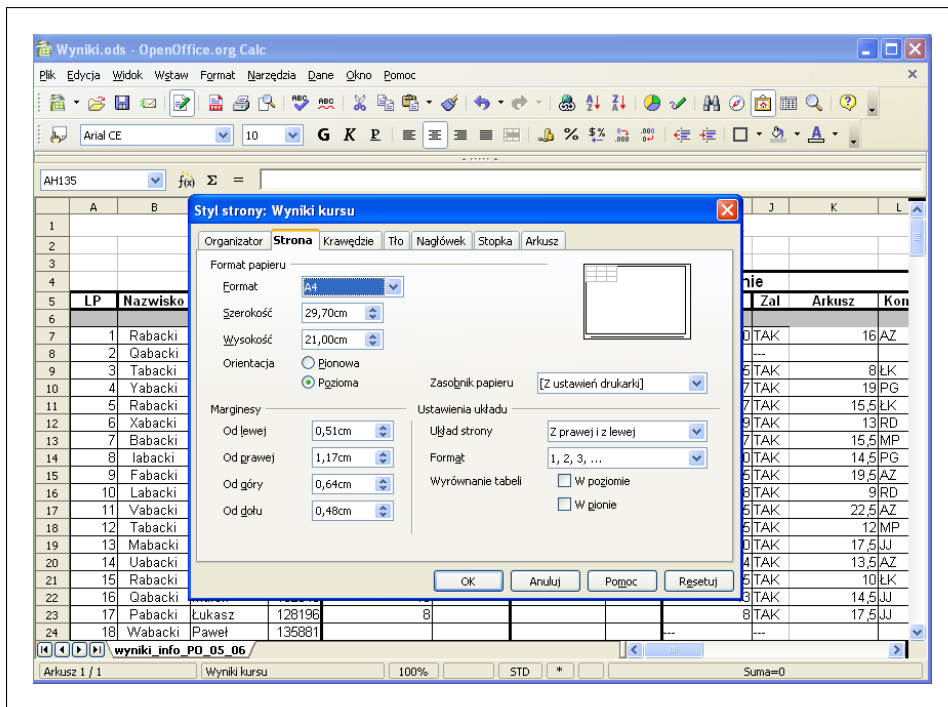


- Orientacja strony
- Marginesy
- Skalowanie

Z tego powodu arkusze kalkulacyjne zwykle drukuje się w położeniu poziomym (orientacja pozioma). Do ustawienia orientacji strony służy polecenie **Format ► Strona ► Strona**

Często udaje się uzyskać znacznie lepsze ułożenie danych na stronach zmniejszając wielkości marginesów. Trzeba jednak pamiętać, że zwykle drukarki nie pozwalają dowolnie zmniejszać marginesów, a poza tym jeśli ustawi się bardzo małe marginesy, to drobne przesunięcie kartki w drukarce spowoduje, że część tekstu nie zmieści się na stronie. Poza tym zawsze warto mieć trochę miejsca na wydruku, by móc robić notatki. Marginesy można zmieniać po wydaniu polecenia **Format ► Strona ► Strona**

Oto okienko służące do ustalania orientacji strony i marginesów:

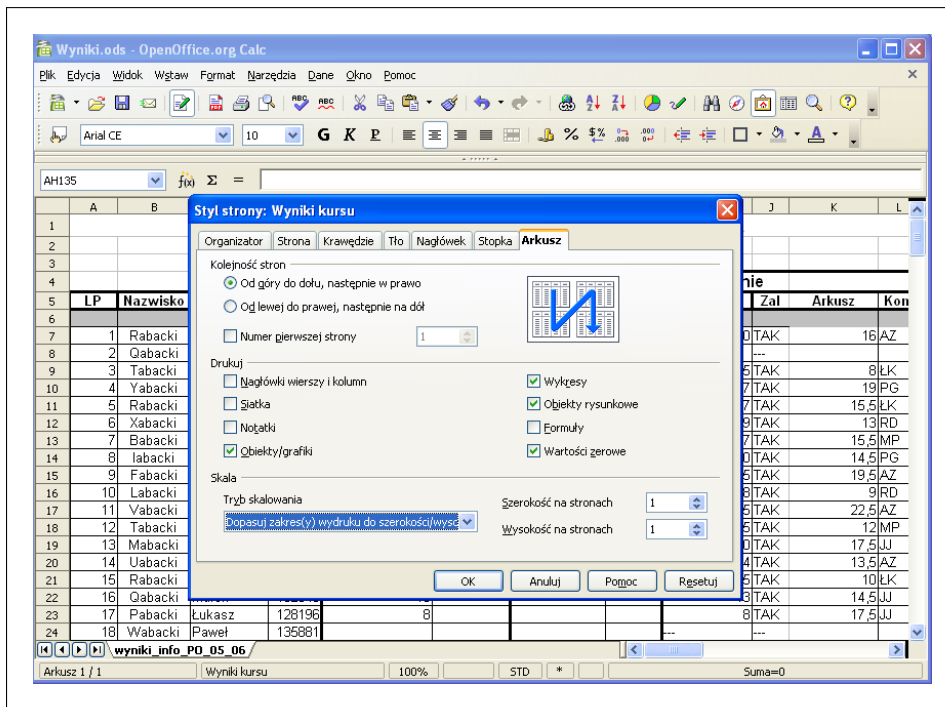


Ustawienie orientacji strony i marginesów

Jeszcze jednym sposobem na poprawienie dopasowania wydruku do kartek papieru jest skalowanie. Możemy podać do ilu procent chcemy pomniejszyć wydruk. Inna możliwość to podanie w ilu stronach w poziomie i pionie ma się zmieścić nasz wydruk. To bardzo wygodne, bo najczęściej nie zależy nam na uzyskaniu małej liczby stron w pionie, natomiast chcemy mieć jak najmniej stron w poziomie (z powodów wyjaśnionych wcześniej). Ustawiamy wówczas np. dwie strony w poziomie i np. 30 (jakaś duża liczba) w pionie (niektóre arkusze kalkulacyjne pozwalają podać, że w jednym z kierunków liczba stron nie ma dla nas znaczenia). Trzecia wreszcie możliwość to po prostu podanie ile stron łącznie ma mieć nasz wydruk. Pamiętajmy przy stosowaniu skalowania w jakiegokolwiek postaci, że zbytne zmniejszenie wydruku uczyni go po prostu nieczytelnym. Skalowanie wydruku ustalamy poleceniem

Format ► Strona ► Arkusz

Tak wygląda okienko służące do ustalania skalowania wydruku:



Ustawienie skalowania

8.2.5 Dodatkowe możliwości związane z drukowaniem

*Bardzo często zależy nam na tym, aby przejście do nowej strony następowo-
ło nie w przypadkowym miejscu ale w wybranym ręcznie przez nas. Można
to łatwo osiągnąć za pomocą dwu poniższych poleceń*

Wstaw ► Podział ręczny ► Podział wiersza

Wstaw ► Podział ręczny ► Podział kolumny

*Przed ich wydaniem należy ustawić bieżącą komórkę w wierszu (lub kolum-
nie), od którego (której) ma zacząć się nowa strona.*

- *Przejścia do nowej strony*
- *Ustawienia drukarki*
- *Wiele egzemplarzy wydruku*

*Jeśli chcemy usunąć tak wstawiony podział strony należy znów ustawić się
w pierwszym wierszu (pierwszej kolumnie) nowej strony i wydać jedno z
poleceń*

Edycja ► Usuń ręczny podział ► Podział wiersza

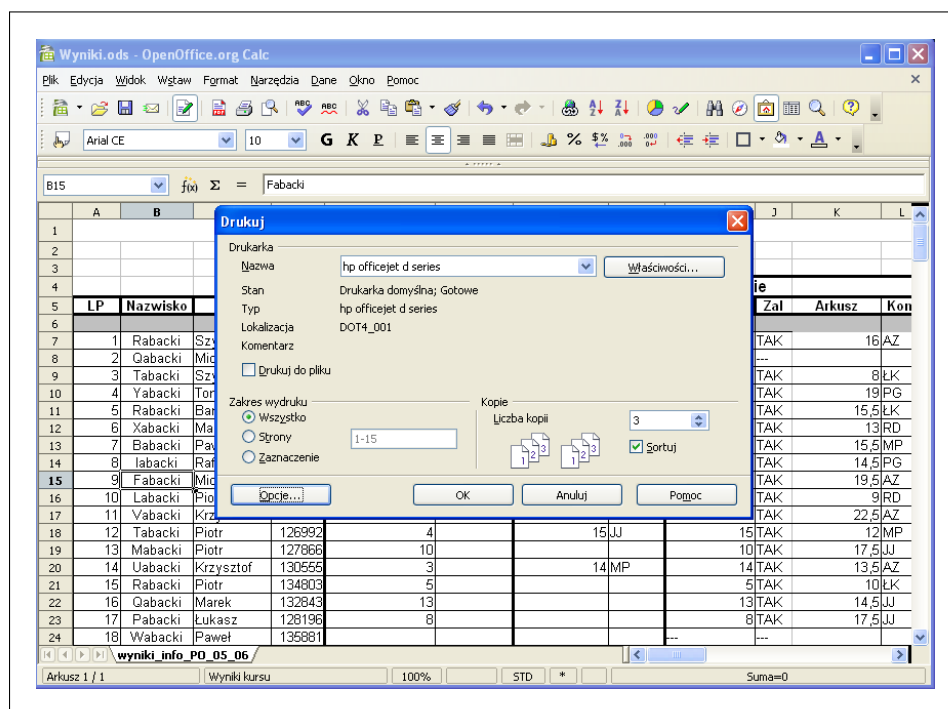
Edycja ► Usuń ręczny podział ► Podział kolumny

Dodatkowe ustawienia wydruku (na przykład wybór druku jedno- lub dwustronnego) można znaleźć w ustawieniach drukarki, dostępnych po wydaniu polecenia

Plik ► Ustawienia drukarki

Niestety dostępne tam ustawienia zależą od drukarki zainstalowanej w systemie i z tego powodu nie omawiamy ich tutaj.

Często zdarza się potrzeba wydrukowania kilku kopii tego samego arkusza. W okienku drukowania



Ustalanie liczby egzemplarzy wydruku

możemy wybrać liczbę egzemplarzy, a także kolejność drukowania stron (najpierw wszystkie pierwsze, potem wszystkie drugie itd., czy najpierw pierwszy egzemplarz, potem drugi, itd.).

We współczesnych arkuszach jest jeszcze wiele ustawień i poleceń związanych z drukowaniem, ale nie jesteśmy w stanie ich wszystkich omówić — przedstawiliśmy tu te najważniejsze.

8.3 Kilka nieco bardziej zaawansowanych tematów

8.3.1 Funkcje bazodanowe

Funkcje bazodanowe (nie mylmy ich z operacjami bazodanowymi takimi jak filtrowanie czy sortowanie) są specjalnym rodzajem funkcji wpisywanych do formuł, które pozwalają wykonywać obliczenia dla (pojedynczych) tabel baz danych z uwzględnieniem kryteriów wybierających wiersze. Oczywiście wymieniona tu tabela bazy danych to po prostu fragment arkusza kalkulacyjnego. Kryteria definiujemy poza samą formułą (co czasami bywa niewygodne) w postaci obszaru kryteriów.

- Czym są funkcje bazodanowe
- Obszar kryteriów
- Najczęściej używane funkcje bazodanowe
- Ogólna postać ich składni

Obszar kryteriów składa się z co najmniej jednej kolumny i dwu wierszy. Pierwszy wiersz zawiera nagłówki. Te nagłówki najczęściej są nagłówkami kolumn z tabelki z baza danych. Jeśli tak jest, to w kolumnie o takim nagłówku będziemy podawali kryteria dotyczące kolumny z bazy danych o takim samym nagłówku. Można też definiować kryteria odnoszące się do wielu kolumn, wówczas w wierszu nagłówków obszaru kryteriów wpisujemy nazwę różną od nazw kolumn w tabeli bazy danych. Warunki z kolumn w jednym wierszu są traktowane jak połączone spójnikiem **i** (koniunkcja). Jeśli podamy więcej niż jeden wiersz pod wierszem z nagłówkami obszaru kryteriów, to warunki z kolejnych będą traktowane jakby były połączone spójnikiem **lub** (alternatywa).

Pojedynczy warunek może mieć postać stałej (np. liczba lub napis) i wtedy są wyszukiwane rekordy mające w kolumnie, w której wystąpił warunek tę wartość. Warunek może mieć postać porównania (np. " $i=13$ "), a co najciekawsze, napis będący warunkiem może być wyliczany formułą.

Praktycznie wszystkie funkcje bazodanowe mają taką samą składnię. Pierwszy parametr jest adresem tabelki z baza danych (koniecznie z wierszem nagłówków), drugi to nazwa kolumny z bazy danych dla której chcemy wykonywać obliczenia, zaś trzeci to adres obszaru kryteriów.

Przykładowe funkcje bazodanowe to:

- `BD.ILE.REKORDÓW` — zliczanie wierszy spełniających kryterium,
- `BD.SUMA` — liczenie sumy liczb ze wskazanej kolumny (tylko z wierszy spełniających kryterium).

Niestety pełne omówienie funkcji bazodanowych wykracza poza ramy tych materiałów.

8.3.2 Ćwiczenia z funkcji bazodanowych

- Dla utworzonych wcześniej tabel zastosuj funkcje bazodanowe
- Pamiętaj o zdefiniowaniu obszaru (obszarów) kryteriów o właściwej strukturze

8.3.3 Analizy co-jeśli

Dzięki temu, że arkusz automatycznie sam wylicza wyniki, za każdym razem gdy tylko zmienimy dane, doskonale się nadaje do przeprowadzania analiz co-jeśli. Na przykład po utworzeniu arkusza obliczającego wysokość rat i czas spłacania kredytu, łatwo możemy (o ile dobrze zdefiniowaliśmy arkusz) policzyć ile będzie rat i jaka będzie ich wartość, gdy zwiększymy lub zmniejszymy kwotę kredytu.

- dlaczego arkusz się świetnie do nich nadaje
- korzystanie z parametrów arkusza
- polecenie *Narzędzia* ► *Szukaj wyniku*

Czasami chcemy wykonać odwrotne obliczenie - znamy wynik, a nie wiemy dla jakich danych można go osiągnąć (np. jesteśmy ciekawi, dla jakiej wysokości kredytu miesięczna rata nie przekroczy 500 złotych). Zamiast metodą prób i błędów podstawiać różne wysokości kredytu i patrzeć jakie wychodzą raty, możemy poprosić żeby takie obliczenie wykonał za nas arkusz — służy do tego polecenie *Narzędzia* ► *Szukaj wyniku*. Podajemy wynik, który chcemy uzyskać, wskazujemy komórkę z wyliczającą go formułą oraz wskazujemy komórkę z danymi, od których zależy wynik (niestety tylko jedną). Zwykle arkusz błyskawicznie poda nam rozwiązanie, choć dla bardziej skomplikowanych przypadków, może się okazać, że arkusz takiego rozwiązania nie potrafi znaleźć.

8.3.4 Uwagi o pisaniu i nagrywaniu makr

- co to jest makro
- jak nagrać makro
- problem trybów adresowania
- jak wygląda makro od środka

Pracując z arkuszem często zauważamy, że niektóre ciągi czynności często powtarzamy. Warto by było więc je zapamiętać w arkuszu i w razie potrzeby kazać je odtwarzać. Można to osiągnąć nagrywając makro, czyli wykonując te czynności, które arkusz ma zapamiętać. Nagrywanie makr włączamy poleceniem

Narzędzia ► Makra ► Zarejestruj makro

. Po jego wykonaniu pojawi się okienko “Zakończ rejestrację” z przyciskiem pozwalającym zakończyć nagrywanie makra. Aby wykonać uprzednio zarejestrowane makro należy wybrać z menu polecenie

Narzędzia ► Makra ► Wykonaj makro

.
Niestety w ten sposób można nagrywać tylko bardzo proste ciągi operacji. Żeby rzeczywiście dostosować arkusz do swoich potrzeb, trzeba napisać odpowiedni zestaw makr (być może zaczynając od nagrania prostego makra). Do pisania makr arkusze udostępniają specjalne języki programowania. Jest to jednak temat zdecydowanie wykraczający poza ramy tych materiałów.

Rozdział 9

Podstawy redagowania dokumentów elektronicznych

9.1 Wprowadzenie

Żyjemy w czasach, w których komputer stał się codziennym narzędziem pracy. Komputery znajdują się w większości gospodarstw domowych i służą nie tylko do zabawy, ale także jako narzędzie pracy czy kontaktu ze światem. Wraz z rozpowszechnieniem się komputerów zanikły dawne metody przygotowywania dokumentów: żmudne sporządzanie rękopisów, czy tworzenie tekstu na maszynie do pisania. O ileż wygodniej jest wprowadzić cały tekst do pamięci komputera, sformatować go nadając mu atrakcyjną postać graficzną, a następnie wydrukować go w tylu egzemplarzach, ile jest nam potrzebnych! A jaka przy tym wygoda przy nanoszeniu poprawek — nie trzeba nic skreślać ani używać korektora — wystarczy wyświetlić stosowny fragment na ekranie komputera i wprowadzić niezbędne poprawki. Do tego ten sam dokument można wykorzystywać wielokrotnie nieznacznie zmieniając pewne jego fragmenty, np.: dane adresata listu.

Nie ulega żadnej wątpliwości, że umiejętność komputerowego przygotowywania dokumentów jest w dzisiejszych czasach koniecznością. Można zaryzykować stwierdzenie, że składają się na nią dwa elementy:

- podstawowa wiedza dotycząca zasad przygotowywania dokumentu,
- umiejętność sprawnego posługiwania się konkretnym programem służącym do redagowania tekstu.

Są pewne uniwersalne zasady, którymi kierujemy się przygotowując dokument tekstowy, niezależnie od wykorzystywanego narzędzia. Dobrym przykładem jest to, że nie stawiamy spacji po znakach interpunkcyjnych (po przecinku, kropce itp). Rozpoczynając pracę z edytorem tekstowym musimy poznać kilka takich właśnie podstawowych zasad.

Wprowadzając tekst do komputera i poddając go obróbce wykorzystujemy odpowiednie oprogramowanie, które musimy dobrze poznać, aby proces re-

dagowania przebiegał sprawnie. Przygotowywanie krótkich tekstów nie wymaga przy tym dogłębnej znajomości narzędzia, którym się posługujemy — zazwyczaj wystarczy opanowanie kilku prostych poleceń i czynności. Inaczej jednak sprawa wygląda, gdy przygotowywanym dokumentem jest na przykład praca dyplomowa. Wtedy zależy nam na tym, aby pewne czynności (na przykład numerowanie rozdziałów) były wykonywane automatycznie. Należy też zadbać o spójność dokumentu, który powstaje w ciągu długiego czasu. Chcemy na przykład, aby wszystkie przykłady były składane w ten sam sposób, aby nagłówki wszystkich rozdziałów miały ten sam rozmiar czcionki itp. Jak widać zmieniają się wówczas wymagania stawiane edytorowi tekstów i zakres, w jakim go używamy.

W trzech kolejnych rozdziałach niniejszego podręcznika chcemy skupić się właśnie na powyższych dwóch aspektach pracy z edytorem i przedstawić niezależnie od siebie:

- elementarne mechanizmy formatowania tekstu,
- mechanizmy strukturalne i złożone narzędzia wspomagające tworzenie długich dokumentów.

Zacniemy od pierwszego zadania. Jako przykładowego edytora użyjemy edytora z pakietu *OpenOffice* w wersji *1.1.2*. Jest to oprogramowanie bezpłatne, dostępne zarówno dla rodziny systemów operacyjnych *Microsoft Windows* jak i systemu operacyjnego *Linux*. Interfejs użytkownika bardzo przypomina interfejs popularnego pakietu *Office* firmy *Microsoft* — opanowanie programu *OpenOffice* umożliwi więc również niemal bezproblemową obsługę programu *Microsoft Word*.

Niniejszy moduł pozwoli *aktywnie studiującej* osobie osiąść umiejętności pozwalające na:

- zrozumienie ogólnych zasad przygotowywania dokumentów elektronicznych,
- korzystanie z elementarnych możliwości wybranego edytora tekstu,
- stosowanie zasad strukturalnej pracy z dokumentem i narzędzi ułatwiających tworzenie długich dokumentów,
- korzystanie z alternatywy dla edytorów typu WYSIWYG (*what you see is what you get*).

9.2 Uruchamiamy i poznajemy edytor tekstu

Programy służące do redagowania tekstu nazywa się w różny sposób. Czasami mówi się o *edytorze tekstu*, czasem używa się nazwy *redaktor tekstu*, jeszcze w innych źródłach występuje *procesor tekstu*.

Jak już wspomnieliśmy, będziemy używać edytor tekstu pochodzący z pakietu *OpenOffice*.

□ **Ćwiczenie 9.2.1.** Znajdź skrót do edytora tekstu (procesora tekstu) w menu startowym używanego przez Ciebie systemu operacyjnego. Uruchom program. □

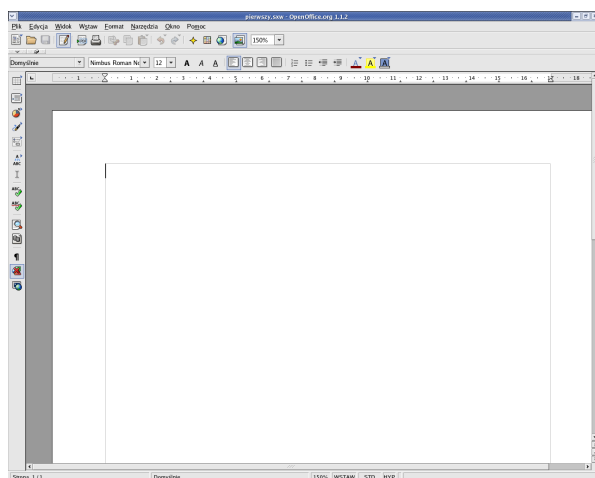
Uruchamianie edytora tekstu:

- jest zależne od systemu operacyjnego i konfiguracji konkretnego komputera,
- najczęściej w menu startowym znajduje się skrót do niego, na przykład w postaci

Menu ►OpenOffice ►Edytor tekstu

- załadowanie programu do pamięci komputera może chwilę potrwać.

Po załadowaniu programu na ekranie powinniśmy zobaczyć okienko programu.



Przyjrzyjmy się najpierw najważniejszym obszarom edytora.

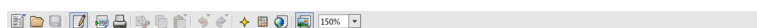
- Największy obszar zajmuje biała, czysta (na razie) kartka, na którą wpisujemy tekst
- U góry ekranu znajduje się nagłówek okienka:



- Pod nim jest menu:



- Jeszcze niżej pasek funkcji:



- Pod paskiem funkcji jest pasek obiektów:



W nagłówku okienka możemy odczytać wersję programu, który używamy oraz nazwę pliku zawierającego opracowywany dokument. Jeśli dokument nie został jeszcze zapisany w żadnym pliku, to pojawia się informacja *bez tytułu*.

Menu zawiera polecenia, które można wykonywać na opracowywanym dokumencie. Na przykład klikając na menu *Plik* otrzymamy podmenu zawierające listę operacji, które możemy wykonywać na plikach. Wybierając z tego podmenu polecenie *Zapisz jako* możemy zapisać opracowywany dokument (na razie pusty) w pliku o podanej nazwie. W dalszej części podręcznika odwołując się do tego typu poleceń wybieranych z podmenu będziemy używać notacji: *Plik ► Zapisz jako*. Taki ciąg poleceń może w niektórych przypadkach być całkiem długi!

Pasek funkcji zawiera ikonki stanowiące skróty do często używanych poleceń. Na przykład kliknięcie na ikonkę z drukarką powoduje przesłanie dokumentu na drukarkę — jest to polecenie, które można również wybrać z menu poprzez *Plik ► Drukuj*.

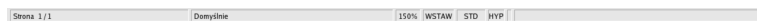
Pasek obiektów umożliwia wykonywanie czynności edycyjnych na aktywnym obiekcie. Jego postać zależy od tego jaki obiekt edytujemy, czyli od tego czy kursor znajduje się w akapicie zwykłym, tabeli, obiekcie graficznym itp. Za pomocą jednego kliknięcia na odpowiednim przycisku tego paska możemy szybko formatować dany obiekt. Przykładowo kliknięcie na przycisk z pogrubioną literką **A** powoduje wytłuszczenie zaznaczonego fragmentu tekstu.

Kolejne elementy wyświetlane w okienku edytora to:

- linijka:



- znajdujący się na dole pasek stanu:



Na linijce są zaznaczane różne odległości stosowane w dokumencie, na przykład wielkość wcięcia pierwszego wiersza, wylczenia, pozycje tabulatorów itd.

Pasek stanu przekazuje ważne informacje o stanie edytowanego dokumentu. Widzimy na przykład, że edytujemy pierwszą stronę spośród jednej dostępnej (napis *1/1*). Kolejna informacja to nazwa stylu użytego do formatowania

aktualnego akapitu (więcej na ten temat powiemy w następnym rozdziale), dalej powiększenie, z jakim oglądamy dokument, tryb edycji, tryb wyboru, informacja o tym, czy hiperłącza są aktywne. Znaczenie poszczególnych informacji zostanie wyjaśnione w dalszej części tekstu. Już teraz jednak warto zwrócić uwagę na *bardzo ważną* informację wyświetlaną w pasku stanu. Bezpośrednio za napisem *HYP* jest pole, które początkowo jest puste. Pole to sygnalizuje, czy dokument został zmodyfikowany od ostatniego zapisu na dysku. Jeśli dokonano w nim modyfikacji w polu tym znajduje się gwiazdka.

□ **Ćwiczenie 9.2.2.** Wprowadź kilka słów do edytora. Zaobserwuj jak zmienia się pasek stanu. □

Ważnym elementem edytora jest też pionowy pasek narzędzi.

Za jego pomocą można szybko wykonywać częste czynności edycyjne. Na przykład pierwsza ikona od góry na tym pasku służy do wstawiania różnych elementów.

□ **Ćwiczenie 9.2.3.** Ustaw kursor myszy na pierwszej od góry ikonie paska narzędzi. Nie klikaj na niego! Zauważ, że *OpenOffice* podpowiada, do czego służy dany przycisk. □

Posługiwanie się paskiem narzędzi jest dość specyficzne. Wyjaśnimy to na przykładzie. Jeśli ustawimy kursor myszy na przycisku *Wstaw* paska narzędzi (przypomnijmy: pierwszy od góry), wciśniemy lewy klawisz myszy **i przytrzymamy go nie puszczając**, to pojawi się podmenu, z którego można wybrać obiekt, który chcemy wstawić. Trzymając cały czas wciśnięty lewy klawisz myszy ustawiamy kursor na przykład na ikonce symbolizującej znak specjalny. System cały czas wyświetla podpowiedź, więc szukaną ikonkę łatwo zidentyfikować. Gdy ustawimy się już na żądanej ikonce puścimy klawisz myszy. Jeśli wybraliśmy polecenie *Wstaw znak specjalny* na ekranie pojawi się dialog, z którego można wybrać żądany znak. Jednocześnie ikona w pasku narzędzi zmienia się na ostatnio wybraną opcję, czyli w tym przypadku — na znak specjalny. Teraz chcąc ponownie wstawić znak specjalny wystarczy zwyczajnie kliknąć lewym klawiszem myszy na ikonę w pasku narzędzi. Chcąc jednak wstawić inny obiekt należy wykonać wszystkie kroki opisane w tym akapicie.

□ **Ćwiczenie 9.2.4.** Poeksperymentuj z paskiem narzędzi. Spróbuj wstawić dwa różne znaki specjalne. □

Warto wspomnieć, że widoczność i zawartość poszczególnych pasków narzędzi można dość dowolnie konfigurować, wykracza to jednak poza ramy tego podręcznika. Wspomnimy jedynie, że czynność tę realizujemy za pomocą zakładek *Pasek stanu* i *Pasek narzędzi* dialogu wyświetlanego poleceniem

9.3 Wprowadzamy tekst

Edycja tekstu polega na jego wprowadzeniu i nadaniu mu odpowiedniej postaci graficznej, czyli na jego *sformatowaniu*. Czynności te można wykonywać na dwa sposoby:

- Wprowadzanie tekstu i jednocześnie formatowanie go. Wprowadzając dokument musimy wówczas skupiać się nie tylko na jego zawartości ale i na prezentacji. Nie mając całego tekstu trudniej jest zaplanować układ strony.
- Formatowanie po uprzednim wprowadzeniu całego tekstu. W pierwszej fazie można skupić się na treści dokonując właściwej prezentacji później. Nie dokonując jednak żadnego formatowania trudno później nawigować po tekście.

Tak naprawdę w praktyce najlepiej połączyć obie metody od razu formatując kluczowe dla struktury dokumentu fragmenty, na przykład tytuły rozdziałów i podrozdziałów, a odkładając na później decyzję, dotyczącą na przykład tego, które fragmenty tekstu będziemy wyróżniać i w jaki sposób. Jeszcze inną metodę, szczególnie przydatną przy edycji, dużych dokumentów omówimy w następnym rozdziale.

Spróbujmy teraz wprowadzić krótki tekst nie dokonując na razie jego formatowania.

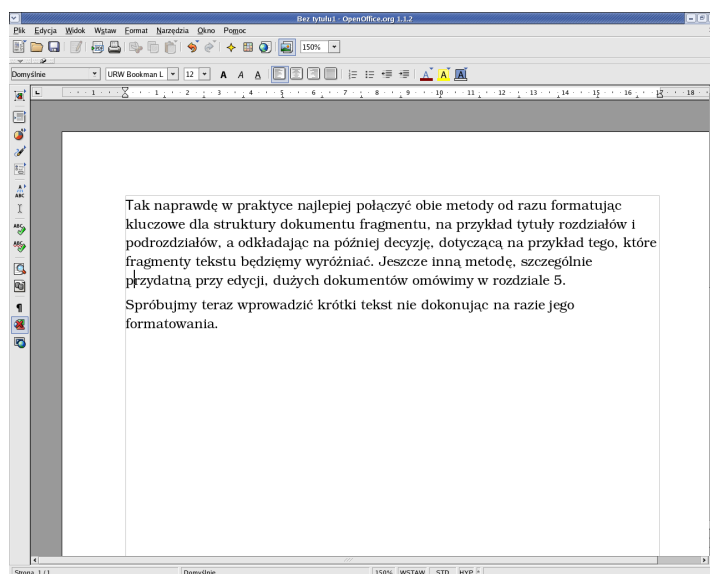
Wprowadzanie tekstu odbywa się w sposób zgodny z oczekiwaniami. Należy jedynie pamiętać o kilku rzeczach:

- Polskie litery uzyskujemy wciskając i przytrzymując *prawy* klawisz *Alt*, wciskając klawisz z odpowiednią literą, a następnie puszczając oba klawisze. Taki sposób wciskania klawiszy, na przykład wciśnięcie klawisza *Alt* razem z literą *a* będziemy w dalszym ciągu oznaczać jako *Alt+a*. Aby uzyskać „ź” wciskamy *Alt+x*.
- Komputer sam dba o przejście do nowego wiersza. Nie trzeba (*a nawet nie należy!*) się o to troszczyć. Jak za chwilę zobaczymy wciśnięcie klawisza *Enter* oznacza nie tylko zwykłe przejście do nowego wiersza.
- Poszczególne akapity oddziela się od siebie za pomocą klawisza *Enter*. W zależności od ustawień edytor może pozostawiać między kolejnymi akapitami nieco więcej miejsca, może też automatycznie rozpoczynać pierwszy wiersz każdego akapitu od wcięcia.
- Niektóre znaki, niedostępne bezpośrednio z klawiatury, wprowadza się w szczególny sposób. Wybieramy z menu *Wstaw ► Znaki specjalne* lub poprzez menu narzędzi. Pojawia się wówczas dialog *Znaki specjalne*, który omówimy za chwilę.

- System automatycznie zamienia niektóre znaki:
 - cudzysłów,
 - kreska, myślnik.

□ **Ćwiczenie 9.3.1.** Wprowadź do edytora krótki (kilka zdań) fragment tekstu złożony z kilku akapitów. Spróbuj umieścić w nim ujętą w cudzysłowy nazwę przedmiotu Przesposobienie Informatyczne. Zauważ, że cudzysłów otwierający jest składany w inny sposób niż cudzysłów zamykający.
□

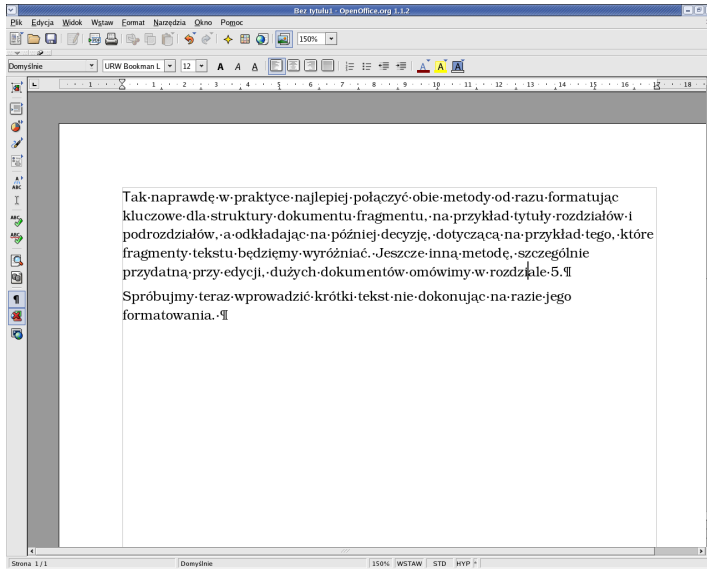
A oto przykładowy tekst wprowadzony do edytora.



Zwróćmy uwagę na to, że:

- Tekst składa się z dwóch akapitów. Jest to wyraźnie widoczne, gdyż między akapitami jest większa przerwa (*światło*) niż między poszczególnymi wierszami.
- Wiersze są wyrównywane jedynie do lewej strony.

Podział na akapity stanie się jeszcze lepiej widoczny, gdy włączymy wyświetlanie tzw. *znaków niedrukowalnych*. Są to znaki, które nie pojawiają się w drukowanym dokumencie i których zadaniem jest oznaczenie specjalnych miejsc w dokumencie. Wyświetlanie znaków drukowalnych włączamy/wyłączamy za pomocą *Widok ► Znaki niedrukowalne*. Można też zrobić to wciskając po prostu *Ctrl+F10* lub poprzez pasek narzędzi.



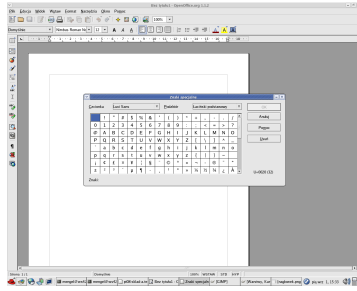
Teraz każda spacja wprowadzona przez użytkownika jest widoczna w postaci kropki, a każde przejście do nowego akapitu w postaci specjalnego znaku graficznego.

Wyświetlenie znaków niedrukowalnych umożliwia łatwiejsze zorientowanie się w strukturze dokumentu i często pozwala na zdiagnozowanie „dziwnego” zachowania się edytora.

Poeksperymentujmy chwilę z tym tekstem. Dodajmy kilka spacji przed słowem „rozdziale”. Zauważmy, że wyrażenie „w rozdziale 5” przesuwa się w stronę prawego marginesu aż w końcu liczba „5” zostaje przeniesiona do nowego wiersza. Takie rozdzielenie słowa „rozdziale” od liczby „5” nie wygląda dobrze. Możemy zapobiec rozdzielaniu kluczowych fraz wstawiając między „rozdziale” a „5” tzw. *twardą spację*. W tym celu usuńmy *wszystkie* spacje przed piątką i wstawmy twardą spację za pomocą *Ctrl+spacja*. Edytor nie dzieli wierszy w miejscach, w których znajduje się twarda spacja. Aby się o tym przekonać ponówmy eksperyment z dostawianiem spacji.

Oprócz twardych spacji możemy wstawiać do tekstu tak zwane *twarde Enter* za pomocą klawisza *Ctrl+Enter*. W miejscu wstawienia tego znaku występuje przejście do nowego wiersza, ale edytor nie kończy akapitu!

Jednym ze sposobów wprowadzania symboli jest dialog *Znaki specjalne*. Przypomnijmy, że wybieramy go za pomocą menu *Wstaw ► Znaki specjalne* lub krótko za pomocą kombinacji klawiszy *Ctrl+Shift+S*. Pojawia się wówczas specjalny dialog:



Wybieramy:

- czcionkę, np.: *Courier*,
- podzbiór znaków, np.: symbole literopodobne (w przypadku niektórych czcionek dysponujących tylko jednym podzbiorem znaków, pole wyboru podzbioru nie jest wyświetlane).

Podwójne kliknięcie na wybrany znak powoduje wstawienie go do dokumentu.

Jest kilka uniwersalnych zasad, którymi warto się kierować przygotowując dokument w edytorze tekstu. Jak już wspomnieliśmy, edytor sam dba o podział tekstu między poszczególne wiersze. Osoba wprowadzająca tekst nie musi tego robić. Edytor wybiera jako punkty podziału miejsca, w których występują spacje. W związku z tym warto pamiętać o tym, aby po znakach przestankowych umieszczać pojedynczą spację. Zwiększa ona czytelność tekstu wprowadzając światło po przecinku, dwukropku, średniku, kropce itp, a ponadto umożliwia „złamanie” wiersza po takim znaku. Z drugiej strony dużym błędem jest umieszczanie spacji przed znakami przestankowymi. Może się wtedy zdarzyć, że edytor podzieli wiersz właśnie w miejscu takiej nadmiarowej spacji, co spowoduje, że przecinek powędruje do nowego wiersza zamiast po prostu zakończyć wiersz poprzedni.

Jeśli nie chcemy, aby w danym miejscu doszło do podziału wiersza, to wpisujemy twardą spację zamiast zwykłej. Typowymi miejscami, których użyjemy twardej spacji jest miejsce między słowem rozdział a jego numerem w odsyłaczach, wyrażenia typu „por. p. 4”, które chcemy traktować jako całość.

Wciśnięcie klawisza *Enter* powoduje nie tylko rozpoczęcie nowego wiersza, ale także podział tekstu na akapity. Nie wciskajmy *Enter*, jeśli chcemy po prostu przejść do nowego wiersza. Używamy zamiast tego *Ctrl+Enter*.

Pamiętajmy więc:

- Poszczególne wyrazy oddzielamy od siebie pojedynczymi spacjami.
- Przed znakami przestankowymi nie stawiamy spacji, ale umieszczamy ją po nich.
- Poszczególne akapity oddzielamy od siebie klawiszem *Enter*.

9.4 Wędrujemy po tekście

Po wprowadzonym tekście możemy poruszać się nie tylko za pomocą kliknięć myszy i paska przewijania. Bardzo efektywną metodą przewijania tekstu jest posługiwanie się klawiaturą:

- klawisze kursora przesuwiają kursor o jedną pozycję w prawo, lewo, do poprzedniego lub następnego wiersza
- klawisze kursora z przytrzymanym klawiszem *Ctrl* umożliwiają szybkie przemieszczanie kursora po pełnych słowach
- *PgDn*, *Pgup* — przewijają ekran o pełną stronę
- klawisze *Home*, *End* umożliwiają szybkie przejście na początek lub koniec strony

Warto podkreślić, że efektywne korzystanie z programów do edycji tekstów wymaga opanowania przemieszczania się po dokumencie także z pomocą samej klawiatury — bez użycia myszki.

9.5 Wprowadzamy poprawki

Dokument można redagować w dwóch trybach: trybie *wstawiania* i *zastępowania*. Przełączanie między nimi odbywa się za pomocą klawisza *Insert*.

- Tryb wstawiania
 - nie wymazuje już istniejącego tekstu
 - wprowadzane znaki są wstawiane między istniejące
 - istniejący tekst „rozsusza się” robiąc miejsce na nowe elementy
- Tryb zastępowania
 - nowe znaki nadpisują istniejący tekst
 - wprowadzenie nowego znaku kasuje znak, który był już zapisany w tym miejscu
- Przełączanie trybów: klawisz *Ins*

W trybie wstawiania nowy tekst trafia między już wprowadzone fragmenty powodując ich „rozsunięcie”. Jest to bezpieczny tryb, gdyż nie jest możliwe przypadkowe zniszczenie fragmentów już istniejącego tekstu. Z tego powodu jest to domyślny tryb pracy. Z kolei tryb zastępowania (lub inaczej *nadpisywania*) powoduje, że wprowadzany tekst zamazuje tekst już istniejący. Stosując ten tryb możemy szybko zastąpić fragment tekstu nowym bez konieczności jego uprzedniego usuwania. Sygnalizacja aktualnego trybu pracy

znajduje się w pasku stanu: napis *WSTAW* sygnalizuje tryb wstawiania a napis *ZAST* — tryb zastępowania.

Czasem zdarza się, że w niezamierzony sposób dokonamy zmiany trybu edycji, wciskając bezwiednie klawisz *Ins*. Jeśli edytor zachowuje się w zaskakujący dla nas sposób kasując tekst, który już został wpisany w miarę wprowadzania nowego, sprawdźmy tryb pracy wyświetlany na pasku stanu i w razie konieczności zmieńmy go.

Czasem zachodzi potrzeba usunięcia jakiegoś fragmentu tekstu. Pojedyncze znaki usuwamy za pomocą naciśnięcia odpowiedniego klawisza, dłuższe fragmenty tekstu najpierw *zaznaczamy*, a potem wydajemy polecenie ich usunięcia.

- Klawisz *backspace* usuwa znak znajdujący się przed kursorem.
- Klawisz *del* usuwa znak, na którym jest kursor.
- Ale uwaga! Klawisz *del* usunie cały fragment tekstu, jeśli dokonaliśmy uprzednio zaznaczenia fragmentu.

Usuwanie tekstu odbywa się za pomocą klawiszy *backspace* lub *del*. Pierwszy z nich usuwa znak znajdujący się *przed* kursorem, drugi — znak *na* pozycji kursora. Jeśli masz problemy z wyborem właściwego klawisza, to staraj się na razie stosować tylko jedną z tych metod. Na przykład, zawsze ustawiaj kursor za usuwanym znakiem i używaj klawisza *backspace*. W miarę zapoznawania się z edytorem właściwe ustawianie kursora stanie się automatyczne — nie będzie potrzeby zastanawiania się, w którym momencie nacisnąć klawisz *backspace*.

Klawisz *del* zadziała jednak inaczej, jeśli zostanie wciśnięty w chwili, gdy w edytorze pewien fragment tekstu jest zaznaczony. Wówczas zostanie usunięty cały zaznaczony tekst. Choć nie mówiliśmy jeszcze o zaznaczaniu fragmentów tekstu, to jednak już teraz warto pamiętać o takim zachowaniu klawisza *del* — jego mimowolne wciśnięcie przy zaznaczonym, być może dużym, fragmencie tekstu, spowoduje jego usunięcie (na szczęście nie jest to jeszcze czynność nieodwracalna)! Jeszcze raz podkreślamy konieczność kontrolowania tego, co dzieje się na ekranie, aby w porę zauważyć efekty niezgodne z naszymi zamierzeniami.

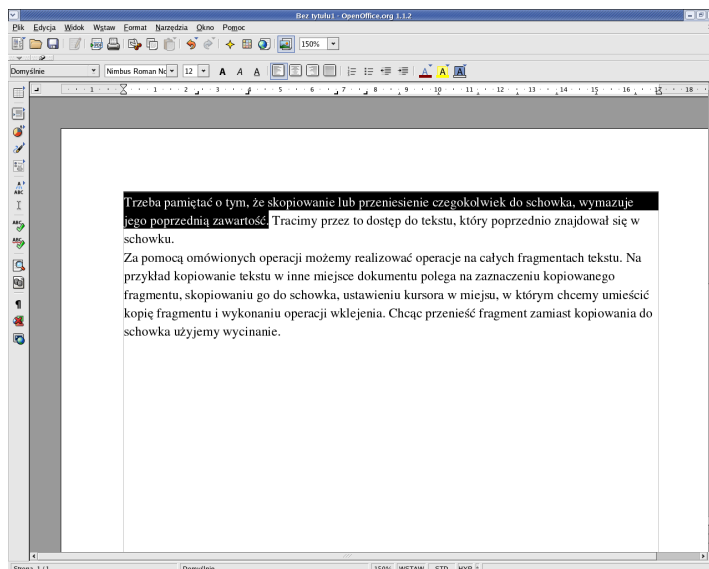
9.6 Zaznaczamy fragmenty tekstu

Wspomnieliśmy już poprzednio, że pewne operacje wykonuje się na fragmentach tekstu. Fragment, na którym chcemy wykonać operację trzeba jednak wcześniej *zaznaczyć*. Można to zrobić na kilka sposobów:

- myszką:
 - kliknąć lewym klawiszem na początku obszaru

- nie puszczać klawisza — trzymać go wciśnięty!
 - przenieść wskaźnik myszki na koniec fragmentu
 - puścić lewy klawisz myszki
 - sprawdzić czy podświetlony jest właściwy fragment
- klawiaturą:
 - ustawić kursor na początku obszaru
 - wcisnąć *Shift* i trzymać go wciśnięty!
 - przemieścić kursor na koniec fragmentu za pomocą klawiszy kursora
 - puścić klawisz *Shift*
 - sprawdzić czy podświetlony jest właściwy fragment

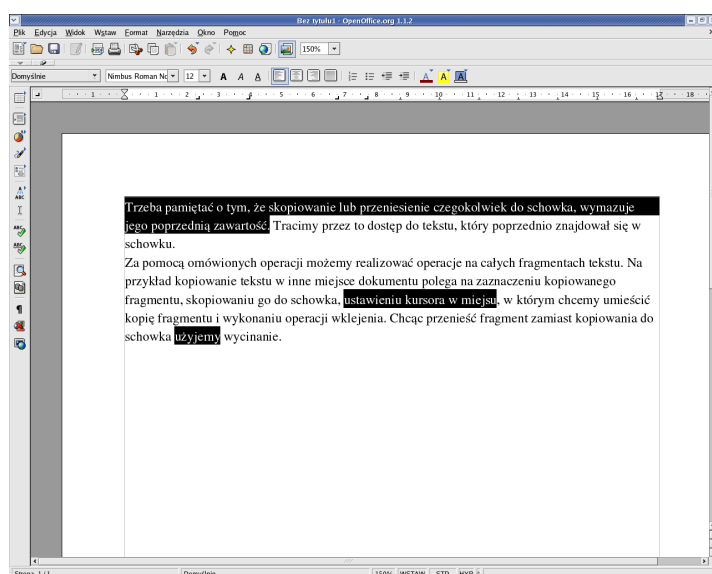
Jeśli wszystko przebiegnie pomyślnie, to zaznaczony fragment tekstu powinien być zaznaczony na ekranie poprzez zamianę jego koloru tła z kolorem czcionki jak w poniższym przykładzie:



Czasem zachodzi potrzeba zaznaczenia kilku fragmentów znajdujących się w różnych miejscach dokumentu. Taką czynność wielokrotnego zaznaczania wykonujemy w następujący sposób:

- zaznaczamy pierwszy fragment
- wciskamy i przytrzymujemy klawisz *Ctrl*
- trzymając wciśnięty klawisz *Ctrl* zaznaczamy drugi fragment
- puszczaemy klawisz *Ctrl*

Jeśli chcemy zaznaczyć więcej fragmentów to po prostu przed zaznaczeniem kolejnego wciskamy klawisz *Ctrl* i puścimy go dopiero po dokonaniu zaznaczenia:



Ćwiczenie 9.6.1. Spróbuj zaznaczyć kilka fragmentów w przygotowanym przez siebie dokumencie.

Przypomnijmy, że obecność zaznaczonego fragmentu zmienia zachowanie niektórych poleceń i klawiszy. Jeśli na przykład po zaznaczeniu fragmentu wcśniemy klawisz *del*, to cały zaznaczony fragment zostanie usunięty. Podobnie jeśli zaczniemy wprowadzać tekst po zaznaczeniu jakiegoś fragmentu, to nowo wpisywany tekst będzie zastępował zaznaczony fragment.

Ćwiczenie 9.6.2. Co się stanie, jeśli po zaznaczeniu fragmentu tekstu klikniesz na niego myszką i nie puścizając lewego klawisza przesuń kursor myszki w inne miejsce dokumentu, a następnie puścisz lewy klawisz myszki?

9.7 Operacje na fragmentach tekstu

Zaznaczony fragment tekstu można łatwo powielić, przenieść w inne miejsce lub usunąć. Zwłaszcza operacja kopiowania (powielania) zaznaczonego fragmentu jest niezwykle przydatna i umożliwia szybkie wprowadzanie podobnych do siebie fragmentów dokumentu.

Operacje kopiowania i przenoszenia wykonuje się za pośrednictwem tzw. *schowka*. Jest to miejsce, w którym możemy umieścić wskazany fragment tekstu. Tekst ze schowka można następnie *wkleić* w dowolne miejsce dokumentu.

Operacje na schowku wykonuje się za pomocą menu *Edycja*. Udostępnia ono między innymi podmenu *Kopiuj*, *Wytnij* oraz *Wklej*. Z każdą z tych

operacji jest także związany skrót klawiszowy. I tak operację kopiowania można szybko wywołać wciskając *Ctrl+c*, operację wycinania — za pomocą *Ctrl+x*, a operację wklejenia — poprzez *Ctrl+v*.

Wyjaśnimy teraz działanie każdej z tych trzech operacji oraz ich wykorzystanie do wykonywania działań na fragmentach tekstu.

- Kopiowanie — kopiuje zaznaczony fragment tekstu do schowka. Podkreślmy: jest to kopiowanie. Zaznaczony fragment pozostaje w dokumencie, a jego *kopia* jest umieszczana w schowku.
- Wycinanie — wycina zaznaczony fragment tekstu z dokumentu i umieszcza go w schowku. W odróżnieniu od kopiowania operacja ta usuwa fragment z dokumentu.
- Wklejanie — kopiuje zawartość schowka do tego miejsca dokumentu, w którym znajduje się kursor. I znów jest to kopiowanie, co oznacza, że zawartość schowka pozostaje bez zmian. Dzięki temu można wielokrotnie wstawić zawartość schowka w wiele miejsc dokumentu.

Trzeba pamiętać o tym, że skopiowanie lub przeniesienie czegokolwiek do schowka, wymazuje jego poprzednią zawartość. Tracimy przez to dostęp do tekstu, który poprzednio znajdował się w schowku.

Za pomocą omówionych operacji możemy realizować operacje na całych fragmentach tekstu. Na przykład kopiowanie tekstu w inne miejsce dokumentu polega na zaznaczeniu kopiowanego fragmentu, skopiowaniu go do schowka, ustawieniu kursora w miejscu, w którym chcemy umieścić kopię fragmentu i wykonaniu operacji wklejenia. Chcąc przenieść fragment zamiast kopiowania do schowka użyjemy wycinania:

- kopiowanie:
 - zaznaczyć fragment
 - wcisnąć *Ctrl+c* — fragment zostanie skopiowany do schowka
 - ustawić się w innym miejscu tekstu
 - wcisnąć *Ctrl+v* — tekst ze schowka zostanie wklejony w tym miejscu dokumentu
- przenoszenie:
 - zaznaczyć fragment
 - wcisnąć *Ctrl+x* — fragment zostanie przeniesiony do schowka
 - ustawić się w innym miejscu tekstu
 - wcisnąć *Ctrl+v* — tekst ze schowka zostanie wklejony w tym miejscu dokumentu
- usuwanie:
 - zaznaczyć fragment

– wcisnąć klawisz *del* lub *Ctrl+x*

Usuwać zaznaczone fragmenty nauczyliśmy się już poprzednio. Zamiast używać klawisza *del* możemy jednak użyć klawiszy *Ctrl+x*. Różnica polega na tym, że w tym drugim przypadku usunięty materiał wędruje do schowka i można go łatwo przywrócić wciskając *Ctrl+v*.

□ **Ćwiczenie 9.7.1.** Skopiuj dowolny fragment tekstu w inne miejsce dokumentu. □

9.8 Proste operacje formatowania

Omówmy teraz pasek obiektów. Znaczenie poszczególnych przycisków jest tu następujące:



- nazwa aktualnego stylu
- nazwa czcionki
- rozmiar czcionki
- grupa przycisków służących do wytłuszczenia, pochylania i podkreślenia zaznaczonego fragmentu tekstu
- grupa przycisków wyrównujących tekst: do lewego marginesu, do środka strony, do prawego marginesu oraz jednocześnie do lewego i prawego marginesu
- grupa przycisków sterujących wyliczaniem
- grupa przycisków ustawiających kolory: czcionki, tła oraz tła całego akapitu

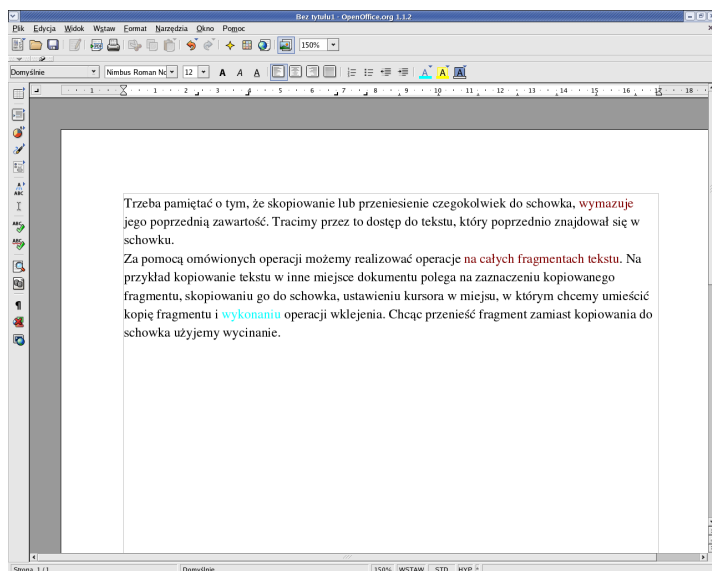
Stylami zajmiemy się szczegółowo nieco później. Pozostałe przyciski służą do szybkiego formatowania zaznaczonych fragmentów tekstu. Jeśli chcemy na przykład wyróżnić jakieś słowo w tekście składając go **grubszą** czcionką, to zaznaczamy to słowo w poprzednio omówiony sposób i naciskamy myszką przycisk z grubą literką „**A**” na pasku obiektów. Ponowne naciśnięcie tego przycisku powoduje powrót do czcionki zwykłej. W podobny sposób dokonujemy wyboru *kursywy* lub podkreślenia w wybranym fragmencie tekstu. Powyższe trzy sposoby można też ze sobą łączyć składając tekst wytłuszczoną kursywą.

Można też zmieniać czcionkę, którą składamy pewne fragmenty dokumentów. Robimy to wybierając po prostu nazwę czcionki w polu na pasku obiektów. Pamiętajmy, że w zależności od systemu operacyjnego i zainstalowanych w nim czcionek wybór ten może być bardziej lub mniej ograniczony.

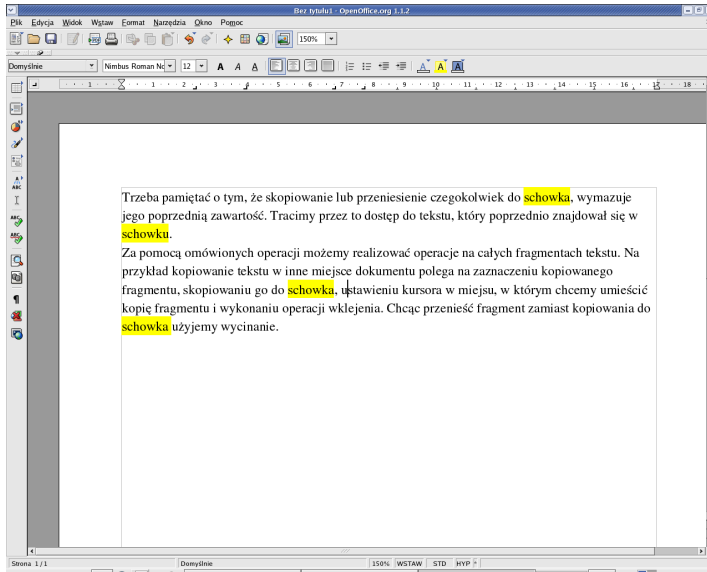
Mamy również wpływ na rozmiar użytej czcionki dzięki polu z rozmiarem czcionki na pasku obiektów. Dostępne rozmiary zależą od wybranego rodzaju czcionki. Nie przy każdej czcionce będzie na przykład dostępny rozmiar 7.

Wraz z rozpowszechnieniem się drukarek kolorowych zwiększyły się także możliwości uatrakcyjniania dokumentów poprzez wykorzystanie w nich umiejętnie dobranych kolorów. Kolory mogą nie tylko zwiększyć wizualną atrakcyjność dokumentów, mogą także zwiększyć czytelność tekstu. Należy jednak pamiętać, aby nie przesadzić! Zbyt pstrokaty tekst nie wygląda dobrze, poza tym odwraca uwagę od swojej zawartości. Przygotowując dokument należy także pamiętać o tym, że być może nie każdy, kto go będzie drukował, dysponuje kolorową drukarką.

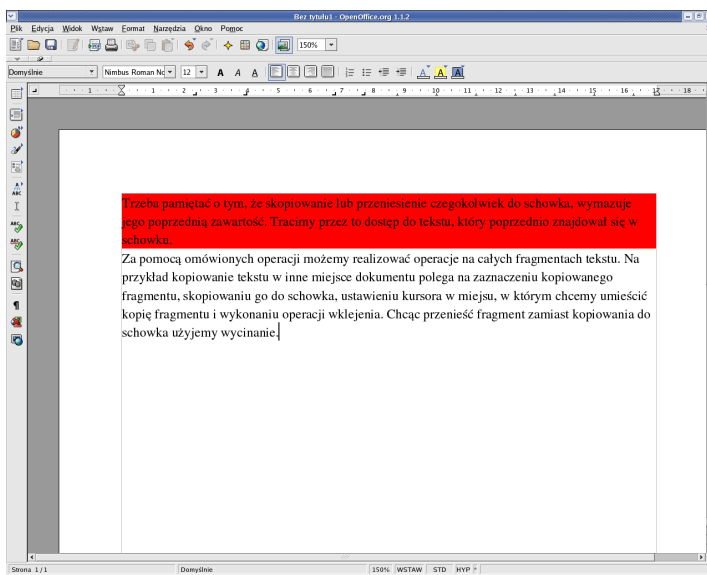
Edytor tekstu umożliwia wykorzystanie kolorów do trzech celów. Można zmienić kolor czcionki tak jak w poniższym przykładzie za pomocą pierwszego przycisku ostatniej grupy z paska obiektów:



Można zmienić kolor tła zaznaczonego fragmentu tak jak w poniższym przykładzie:



Można wreszcie zmienić kolor całego akapitu (ostatni przycisk na pasku narzędzi):



To ostatnie polecenie jest przykładem formatowania wykonywanego na całym akapicie, a nie tylko na zaznaczonym fragmencie tekstu. Akapit jest w edytorze pewną logiczną jednostką i jest wiele poleceń formatowania, które dotyczą całych akapitów. Z tego właśnie powodu warto zwrócić baczną uwagę na to, o czym już pisaliśmy: na właściwe użycie klawisza *Enter*.

Przypomnijmy raz jeszcze klawisz *Enter* służy do oddzielania od siebie akapitów. Nigdy nie używamy go do wymuszania przejścia do nowego wiersza. Pamiętajmy:

- Akapity stanowią logiczną jednostkę w całym dokumencie.

- Wiele operacji i atrybutów dotyczy całych akapitów.
- Akapity oddziela się od siebie wciskając klawisz *Enter*.
- Znaki oddzielające akapity są widoczne po włączeniu podglądu znaków niedrukowalnych.

Pozostała do omówienia jeszcze jedna grupa przycisków paska obiektów: przyciski służące do wyrównywania tekstu. Tak jak w przypadku koloru akapitu, przyciski te działają na całym akapicie. Aby sprawdzić ich działanie ustawmy się w dowolnym akapicie i wciśnijmy najpierw przycisk wyrównywania do lewej strony. Jest to domyślny sposób składania akapitu. Wszystkie jego wiersze (z wyjątkiem być może pierwszego) są wówczas wyrównywane do lewego marginesu (plus pewien dodatkowy odstęp). W analogiczny sposób działa równanie do prawego marginesu, które jest przydatne na przykład przy składaniu dat w nagłówku listu, czy też podpisu. Akapit można także wypośrodkować na stronie. Są cztery sposoby wyrównania tekstu na stronie:

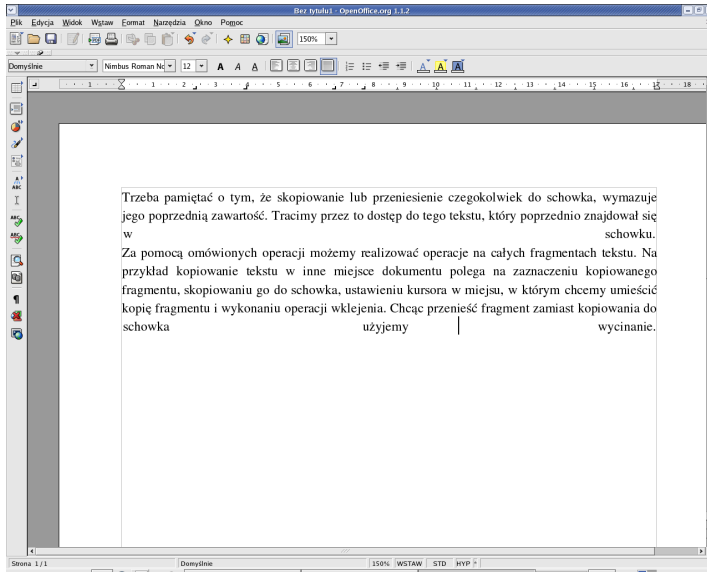
- do lewej
- do prawej
- na środek
- do lewej i do prawej

Ostatnia możliwość to równanie jednocześnie do prawego i do lewego marginesu. Edytor wykonuje to zadanie wstawiając dodatkowe spacje między wyrazami.

□ **Ćwiczenie 9.8.1.** Włącz równanie do lewego i prawego marginesu. Włącz podgląd znaków niedrukowalnych. Czy potrafisz powiedzieć, które spacje zostały wpisane przez Ciebie, a które dostawił sam edytor? □

Zauważmy, że wyrównywane są wszystkie wiersze danego akapitu z wyjątkiem ostatniego (choć można zmienić to ustawienie). Dzieje się tak, gdyż ostatni wiersz może być krótki (na przykład może składać się z dwóch wyrazów). Wyrównanie takiego wiersza jednocześnie do lewego i prawego marginesu spowodowałoby umieszczenie pierwszego wyrazu po lewej stronie, dosunięcie prawego do prawego marginesu i umieszczenie bardzo dużej przerwy między nimi, jak w następującym przykładzie:

Należy unikać wyrównywania ostatniego wiersza akapitu:



9.9 Zapisujemy swoją pracę

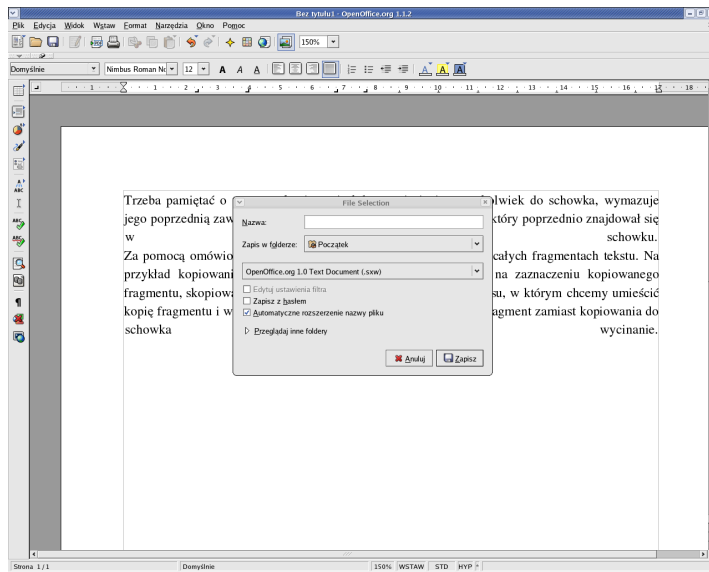
Dokumenty tworzone za pomocą edytora należy zachować na dysku komputera. Pamiętajmy, że:

- dokumenty są przechowywane na dysku w postaci plików
- dane zapisane na dysku są nieulotne
- warto regularnie i często zapisywać zmiany w dokumencie na dysku
- dokumenty są zapisywane w formacie pakietu OpenOffice
- można dokonać zapisu w innych formatach, m.in.:
 - Microsoft Office
 - PDF

Dokumenty, jak i inne informacje, są przechowywane na dysku. Aby nie utracić pracy, przez wyjściem z programu edytora należy zapisać ją na dysk wciskając *Ctrl+S* lub posługując się menu *Plik*, opcją *Zapisz*.

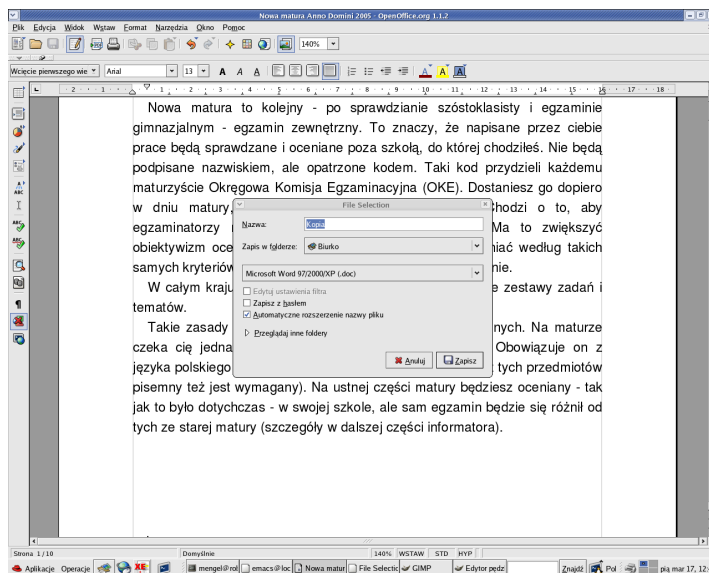
Należy wyrobić sobie nawyk częstego zapisywania tworzonego dokumentu na dysku. Pozwoli to uchronić się przed utratą znacznej części dokumentu w przypadku awarii zasilania, komputera czy oprogramowania.

Dokument jest zapisywany standardowo jako dokument programu *OpenOffice*. Sekwencja operacji niezbędna do zapisu dokumentu na dysk to *Plik* ► *Zapisz*. Przy pierwszym zapisie nowo utworzonego dokumentu na dysk pojawia się wówczas dialog, za pomocą którego można nadać nazwę plikowi, w którym zapisujemy dokument. Postać tego dialogu jest zależna od systemu operacyjnego, pod którym pracujemy, w systemie Linux wygląda on na przykład tak:



Dokumenty są zapisywane w plikach, których nazwy kończą się rozszerzeniem *.sxd*. Tak zapisany dokument można będzie później bez problemu wczytać do edytora i kontynuować pracę nad nim. Warto zapamiętać, że zapisanie dokumentu na dysk można szybko wykonać wciskając klawisze *Ctrl+S*. Warto co jakiś czas wciskać te klawisze!

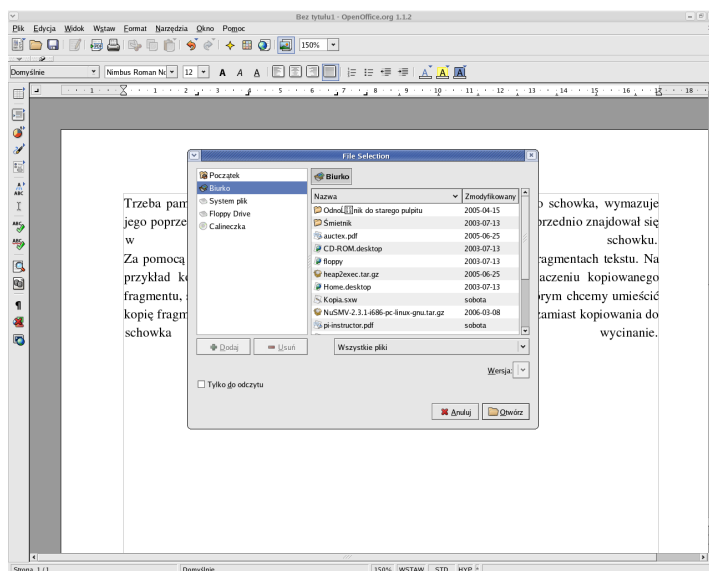
Istnieją jednak także inne możliwości. Czasem chcemy zapisać opracowywany dokument w innym formacie tak, aby można było na przykład otworzyć go za pomocą innych programów (np. pakietem *Microsoft Office*). Wówczas zamiast *Plik ► Zapisz* należy posłużyć się *Plik ► Zapisz jako* i dokonać wyborużądanego formatu z listy:



Zapis w innym formacie jest z pewnością bardzo atrakcyjny dla osób, które posługują się różnymi programami, gdyż umożliwia przenoszenie dokumentów między popularnymi programami do redakcji tekstów. Należy jednak

liczyć się z pewnymi różnicami w wyglądzie oryginalnego dokumentu a dokumentu zapisanego w formacie „obcym”.

Dokumenty zapisane na dysku można później bez problemów otworzyć i wznowić pracę na nimi. Odbywa się to poprzez menu *Plik* ► *Otwórz*. Tak jak przy zapisie dokumentów pojawia się dialog, za pomocą którego można wybrać plik do wczytania — i znów postać tego dialogu zależy od systemu operacyjnego, pod którym pracujemy. Przykład w systemie operacyjnym Linux wygląda tak:



Aby otworzyć dokument w edytorze należy:

- wybrać z menu *Plik* opcję *Otwórz*
- w okienku dialogowym wybrać rodzaj pliku (pliki z dokumentami stworzonymi w pakiecie OpenOffice mają końcówkę *.sxw*)
- wybrać nazwę pliku

Pakiet *OpenOffice* rozumie także wiele formatów „obcych”. Oznacza to, że można wczytać pliki stworzone za pomocą innych popularnych pakietów. Lista rozumianych przez pakiet formatów jest dość długa i można ją przejrzeć w okienku otwierania pliku. W tym celu klikamy na pole *Wszystkie pliki* i wybieramy żadaną grupę formatów (na przykład *Dokumenty tekstowe*) lub konkretny format (na przykład *Microsoft Word 97/2000/XP*). W okienku wyświetlane są jedynie nazwy tych plików, które mają nazwę zgodną z wybranym formatem — w przypadku wybrania formatu *Microsoft Word* będą to pliki których nazwy kończą się na *.doc*.

9.10 Linijka

Pasek linijki służy do ustawiania różnych odległości na stronie. Za jego pomocą można m.in. ustawić szerokość strony, wielkość lewego i prawego

marginesu, wielkość wcięcia stosowanego w pierwszym wierszu oraz pozycje tabulatorów.



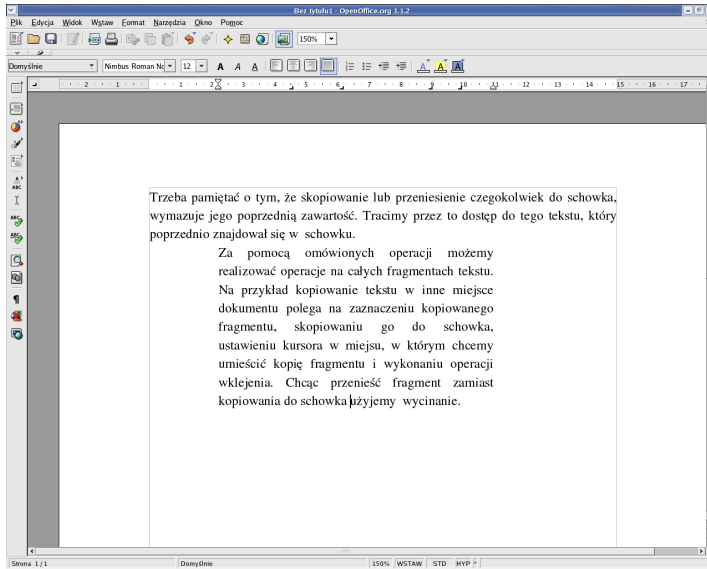
- umożliwia określenie rozmiaru lewego i prawego marginesu strony
- definiuje punkty wcięć, z których każdy może być:
 - lewy
 - prawy
 - lub środkowy
- umożliwia wyznaczenie dodatkowego wcięcia z lewej i prawej strony każdego akapitu

W górnej części linijki znajduje się podziałka, która może być wyświetlana w różnych jednostkach, m.in. w centymetrach, milimetrach, calach. Zmiany jednostki dokonujemy klikając prawym klawiszem na linijkę i zaznaczając potrzebną nam jednostkę. W przedstawionym przykładzie linijka składa się z trzech części:

- szarego obszaru rozciągającego się od początku do zera (nie zaznaczonego na podziałce). Jest to lewy margines — niezapełniany tekstem obszar po lewej stronie strony.
- białego obszaru oznaczającego obszar strony, w którym będzie pojawiał się tekst.
- ponownie szarego obszaru po prawej stronie — czyli prawego marginesu.

Rozmiary tych obszarów można zmieniać. Jeśli ustawimy kursor myszy na granicy obszarów (powinien zmienić się on wówczas na podwójną strzałkę), a następnie klikniemy lewy klawisz myszy i przytrzymując go *pociągniemy* myszkę w lewo lub prawo, to granica przesunie się zgodnie z ruchem myszki. W ten sposób możemy łatwo i szybko ustawiać lewy i prawy margines strony.

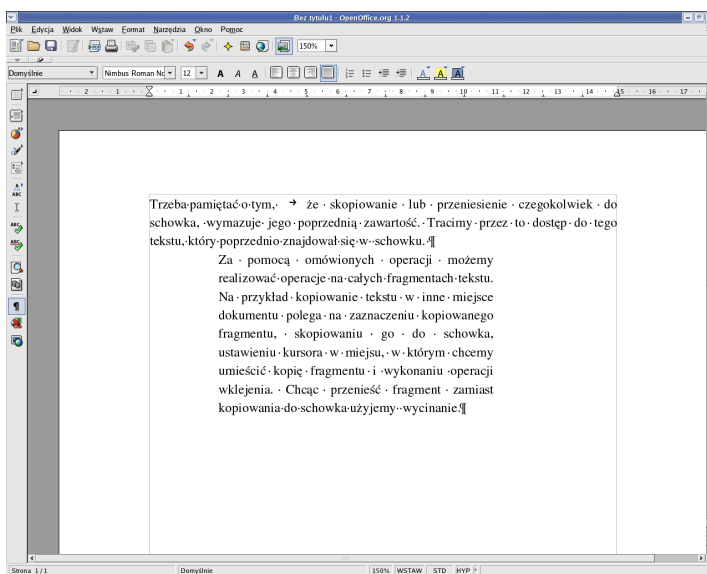
Na linijce znajdują się ponadto trzy trójkąty: dwa skierowane w górę i jeden skierowany w dół. Trójkąty skierowane do góry wyznaczają obszar zajmowany przez aktualnie edytowany akapit. Aby zobaczyć, jaki wpływ ma ich położenie na tekst przygotujmy krótki dwuakapitowy tekst. Następnie ustawmy kursor w *drugim* akapicie i przesunijmy lewy trójkąt nieco w prawo, a prawy — nieco w lewo. Powinniśmy uzyskać coś na kształt następującego obrazka:



Widać zatem, że w powyższy sposób można kontrolować ustawienie pojedynczego akapitu na stronie.

Górny trójkąt skierowany w dół po lewej stronie wyznacza z kolei rozmiar wcięcia dla pierwszego wiersza akapitu. Jeśli pokrywa się on z trójkątem dolnym tak jak w dotychczasowych przykładach, to pierwszy wiersz nie jest wcinany. Jego przesunięcie na prawo od dolnego trójkąta powoduje wcięcie pierwszego wiersza akapitu.

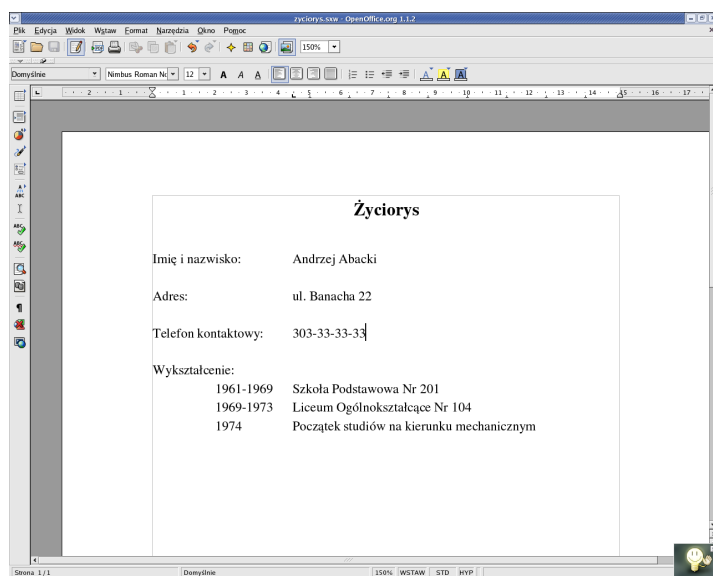
Dodatkowo na linijce znajdują się znaki *wcięć*. Aby zrozumieć ich działanie włączmy podgląd znaków niedrukowalnych. Następnie ustawmy się w dowolnym punkcie wiersza i wciśnijmy klawisz *Tab* na klawiaturze. Tekst na prawo od położenia kursora zostaje przesunięty w prawo do najbliższego znaku wcięcia, zaś wciśnięcie klawisza *Tab* zostało oznaczone specjalnym znakiem niedrukowalnym: strzałką.



□ **Ćwiczenie 9.10.1.** Poeksperymentuj wciskając klawisz *Tab* po kilka razy w różnych miejscach dokumentu. Spróbuj usunąć już wprowadzone znaki tabulacji. Obserwuj co dzieje się z tekstem. □

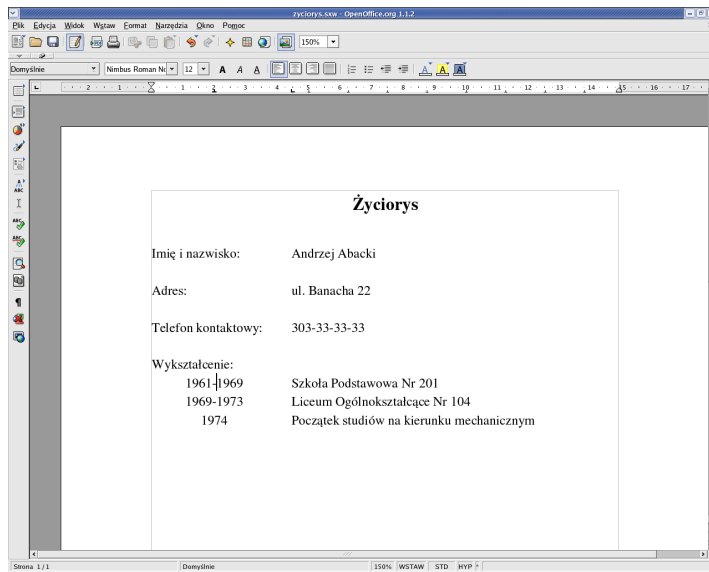
Pozycje znaków wcięć na linijce możemy zmieniać. Kliknięcie na linijce spowoduje wstawienie znaku wcięcia. Zwróć uwagę, że znaki wcięć wstawione przez użytkownika mają inny symbol niż standardowe znaki symbolizowane przez małe odwrócone literki „T”. Przeciągnięcie wstawionego znaku poza linijkę powoduje jego usunięcie, a przeciągnięcie go w inne miejsce linijki — zmianę położenia.

Tabulatory przydają się do przygotowywania dokumentów zawierających części, które muszą być względem siebie wyrównane. Oto przykładowy fragment życiorysu złożony z wykorzystaniem znaków wcięć. Zwróć uwagę na położenie znaków wcięć na linijce.



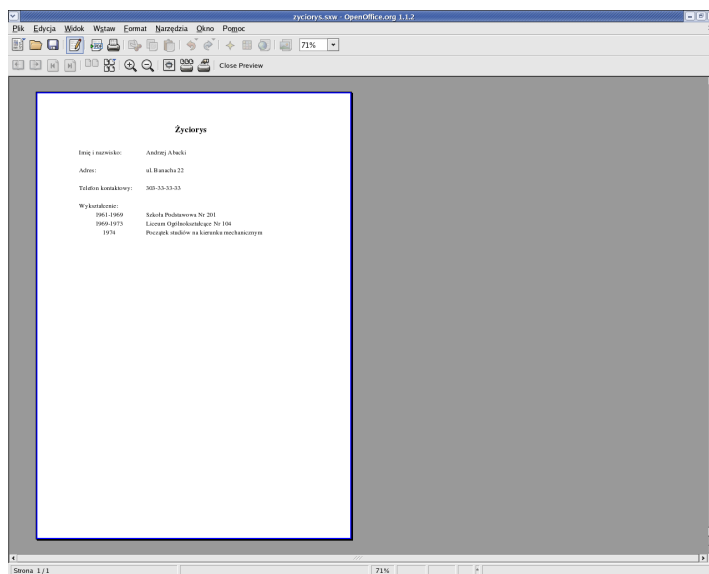
Z każdym znakiem wcięcia jest związany sposób ściągnięcia tekstu do niego. Każdy znak może wyznaczać albo lewą krawędź tekstu, albo jego środek albo też prawą stronę tekstu. Rodzaj tabulatora zmienia się klikając go na linijce prawym przyciskiem myszy i wybierając żadaną funkcję.

Wyjaśnijmy działanie tego mechanizmu na przykładzie poprzedniego życiorysu. W tym celu zmienimy rodzaj pierwszego tabulatora na *wyśrodkowanie*. Otrzymamy następujący tekst (zwróć uwagę na sposób złożenia dat w akapicie *Wykształcenie*):



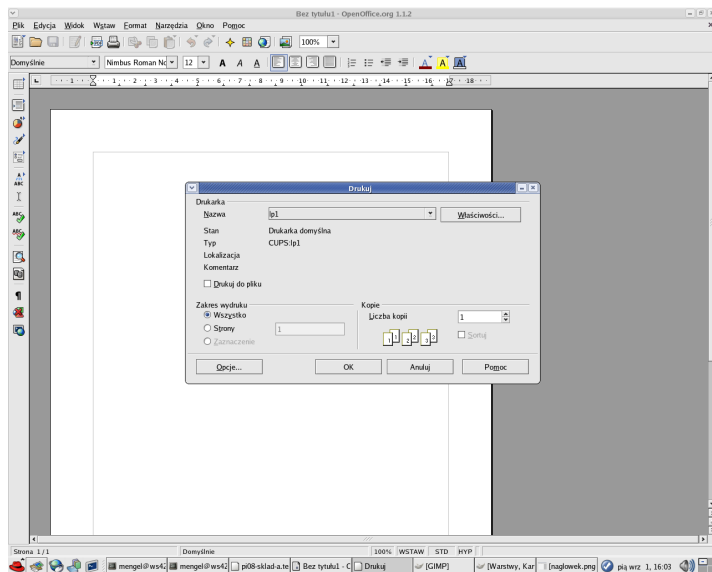
9.11 Przygotowujemy dokument do druku

Przygotowany dokument można wydrukować. Przed wysłaniem dokumentu na drukarkę warto jednak obejrzeć go na ekranie w takiej postaci, w jakiej pojawi się on później na papierze. W tym celu wybieramy opcję *Plik* ► *Podgląd wydruku*. Na ekranie zobaczymy wówczas układ strony. Pasek obiektów zmienia swoją postać, oferując teraz narzędzia służące m.in. do powiększania i zmniejszania podglądu i do przełączania w tryb oglądania wielu stron na raz:



Jeśli dokument wygląda tak, jak tego oczekiwaliśmy, można go wysłać na drukarkę. W tym celu wybieramy opcję *Plik* ► *Drukuj*. Na ekranie pojawia się dialog, za pomocą którego można wybrać drukarkę, na której ma się

pojawić wydruk (w przypadku gdy dysponujemy większą liczbą drukarek lub mamy dostępną drukarkę sieciową).



Klikając myszką na przycisk *Właściwości* można ustawić parametry wydruku, które mogą być zależne od konkretnej drukarki. Z kolei w polu *Kopie* możemy ustawić liczbę egzemplarzy wydruku, a w polu *Zakres* decydujemy, jaką część dokumentu wydrukować. Do wyboru mamy albo cały dokument albo konkretny zakres stron (wprowadzany w postaci listy numerów stron oddzielonych od siebie przecinkami lub znakiem „-”, na przykład *1-4, 7-10* oznacza strony od pierwszej do czwartej oraz od siódmej do dziesiątej). Jest także możliwe wydrukowanie uprzednio zaznaczonego fragmentu dokumentu. Kliknięcie klawisza *Opcje* na dole dialogu daje dodatkowe możliwości sterowania zawartością wydruku.

Gdy wszystkie parametry są już ustawione wystarczy już tylko zamknąć dialog klikając w klawisz *OK* i zdjąć wydrukowany dokument z drukarki.

9.12 Formatowanie akapitu, strony i dokumentu

Podsumujmy teraz poznane możliwości formatowania tekstu i poznamy dodatkowe narzędzia redagowania tekstu. Widzieliśmy już, że niektóre polecenia formatowania dotyczą „lokalnych” właściwości tekstu: można wytłuszczyć czcionkę, powiększyć jej rozmiar lub zmienić krój zaznaczając fragment tekstu i wybierając odpowiednie polecenie, np.: z paska obiektów lub z menu. Widzieliśmy też, że niektóre polecenia dotyczą całego akapitu, a nie jedynie zaznaczonego fragmentu tekstu. Są także polecenia, które służą do zmiany formatowania całej strony albo nawet całego dokumentu.

Rozpocznijmy od formatowania akapitu, które:

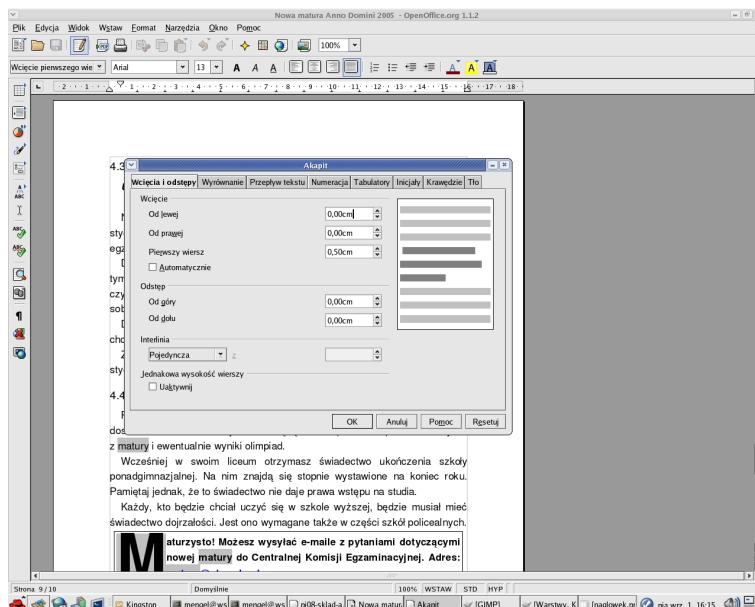
- ustala atrybuty *całego* akapitu

- jest wykonywane przez ustawienie kursora w edytowanym akapicie i wybranie opcji *Format ► Akapit*
- ustala m.in. następujące parametry:
 - odstęp edytowanego akapitu od lewego i prawego marginesu
 - rozmiar wcięcia w pierwszym wierszu akapitu
 - odstęp między poszczególnymi wierszami akapitu
 - odstęp od poprzedniego akapitu
 - wyrównanie tekstu w akapicie
 - wyrównanie tekstu w ostatnim wierszu akapitu
 - sposób przenoszenia słów między wierszami
 - sposób dzielenia akapitu na strony
 - widoczność krawędzi akapitu
 - kolor tła i tekstu
 - ...

Jak już wspomnieliśmy edytor tekstu umożliwia formatowanie różnych „jednostek” tekstu. Polecenia do tego celu znajdują się w menu *Format*.

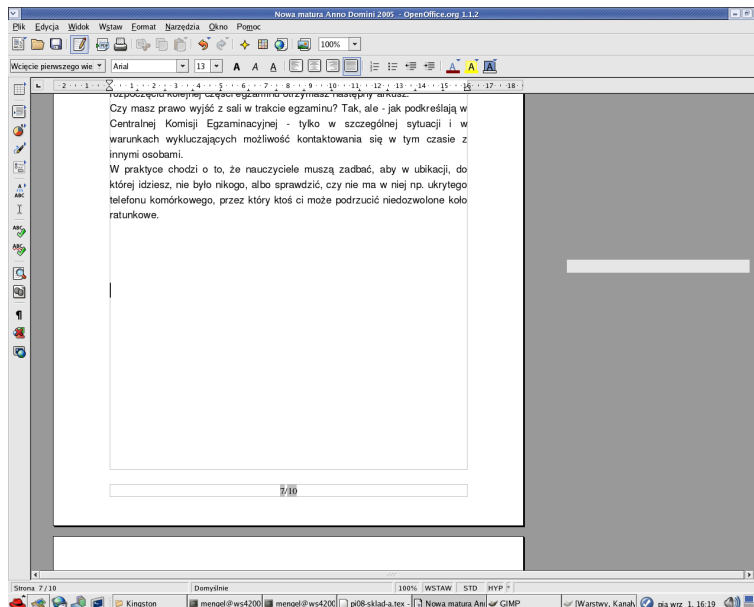
Chcąc zmienić sposób składania akapitu należy ustawić się w edytowanym akapicie i wybrać opcję *Format ► Akapit*. Inny sposób to kliknięcie prawym klawiszem myszy w edytowany akapit i wybór opcji *Akapit* z wyświetlonego menu. Na ekranie pojawi się wówczas dialog *Akapit* grupujący wszystkie polecenia i parametry dotyczące akapitu — także te, które poprzednio realizowaliśmy w inny sposób (na przykład z paska obiektów).

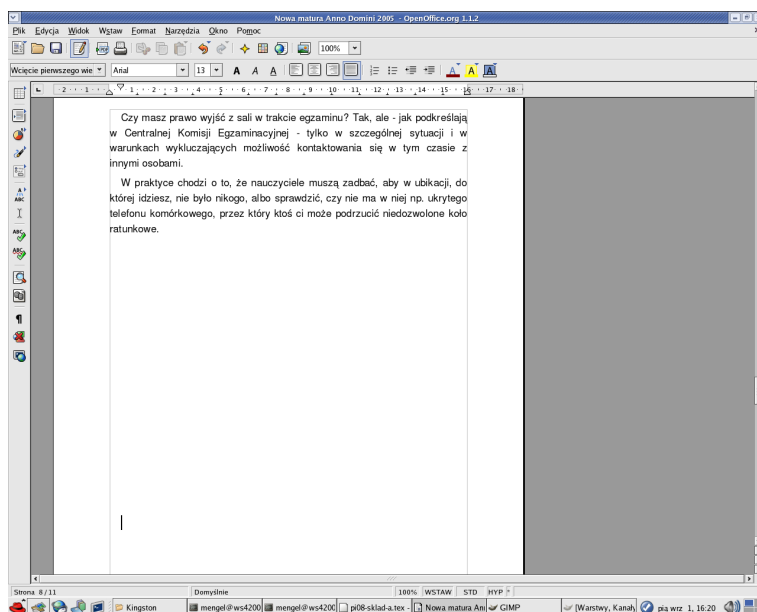
Dialog *Akapit* jest dość rozbudowany. Składa się z kilku zakładek, które można wybierać klikając w ich nazwy u góry dialogu.



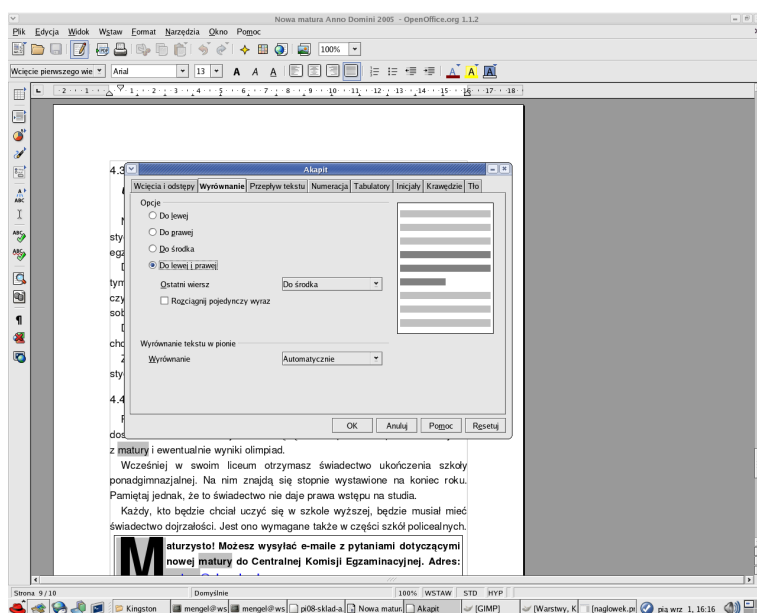
Zakładka *Wcięcia i odstępy* służy do ustawiania odstępów akapitu od lewego i prawego marginesu, a także odstępu, jaki ma być pozostawiony przed akapitem i po nim. Można tu również ustawić rozmiar wcięcia pierwszego wiersza w akapicie. Przypomnijmy, że wielkości te ustawialiśmy uprzednio korzystając z linijki. Okienko po prawej stronie dialogi na bieżąco pokazuje skutek dokonanych zmian. Za pomocą ustawienia interlinii decydujemy, jaki ma być odstęp między poszczególnymi wierszami akapitu. W ten sposób można na przykład tworzyć akapity, w których wiersze są rozmieszczane „rzadko” (podwójna interlinia).

W praktyce najbardziej przydatnymi parametrami jest możliwość ustawienia dodatkowego światła między akapitami i wcięcia pierwszego wiersza każdego akapitu. Ustawienie tych parametrów zdecydowanie polepsza czytelność dokumentu. Porównajmy dla przykładu następujące dwa sposoby sformatowania tekstu w dokumencie. Zwróć uwagę na wyraźne oddzielenie od siebie akapitów w drugim przykładzie, dzięki zastosowaniu dodatkowego światła między akapitami i wcięcia pierwszego wiersza:



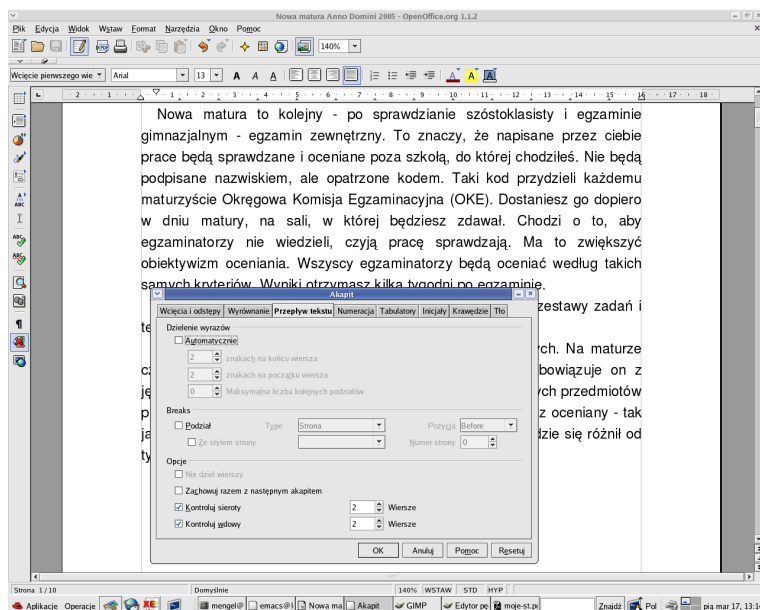


Kolejna zakładka to Wyrównanie.



Służy ona do ustawienia układu tekstu akapitu na stronie. Tekst może być wyrównywany do lewego lub do prawej marginesu, może być wypośrodkowany lub też wyrównywany zarówno do prawego jak i do lewego marginesu. Przy tym ostatnim ustawieniu możemy ponadto decydować o sposobie złożenia ostatniego wiersza. Jak już wspominaliśmy, wymuszenie wyrównania ostatniego wiersza zarówno do lewego jak i do prawego marginesu może spowodować nieelegancki wygląd tekstu.

Przeplływem tekstu do tej pory jeszcze się nie zajmowaliśmy. Do ustawiania związanych z nim parametrów służy kolejna zakładka dialogu:



Przeływ tekstu steruje sposobem podziału edytowanego akapitu na wiersze i strony. Domyślnie edytor nie dokonuje podziału ostatnich słów w wierszu na sylaby. Czasem jednak włączenie takiego podziału może polepszyć wygląd dokumentu (zmniejszając odstęp między wyrazami w akapicie wymuszane koniecznością równania do lewego i prawego marginesu). Automatyczne dzielenie wyrazów można włączyć zaznaczając opcję *Automatycznie*. Warto wtedy ustawić także parametry tego podziału: minimalną dozwoloną liczbę znaków na końcu i na początku wiersza oraz liczbę kolejnych wierszy, w których może dojść do podziału słów na wiersze. Omawiana zakładka umożliwia ponadto wymuszenie podziału strony przed lub po składanym akapicie. Włączenie takiej opcji powoduje, że edytor automatycznie dokonuje podziału strony odpowiednio przed lub po akapicie.

Ważnym elementem zakładki są opcje, które umożliwiają sterowanie podziałem akapitu na strony. Jak już wiemy edytor sam decyduje, w którym miejscu przejść do nowego wiersza. Podobnie automatycznie jest podejmowana decyzja o miejscu podziału strony. Czasem jednak miejsca podziałów stron nie są dobrze dobrane, przez co dokument wygląda nielogicznie. Nie jest dobrze, jeśli podział strony wypada pod koniec (lub na początku) długiego akapitu tak, że na nowej stronie znajduje się jedynie ostatni wiersz akapitu (lub też wszystkie wiersze z wyjątkiem pierwszego). Podział strony nie powinien występować także między nagłówkiem rozdziału a jego pierwszym akapitem. Jedną z możliwości radzenia sobie ze złym podziałem na strony jest ręczne wstawianie punktów podziału poprzez opcję *Wstaw ► Ręczny podział*. Często jednak lepszym i skuteczniejszym (bo odpornym na późniejsze zmiany w dokumencie) sposobem jest odpowiednie ustawienie opcji w omawianej zakładce. I tak: zaznaczenie opcji *Nie dziel wierszy* zabrania edytorowi dzielić na strony wewnątrz edytowanego akapitu. Oznacza to, że jeżeli cały akapit nie zmieści na stronie, to zostanie *w całości* przeniesiony na stronę następną. Z kolei zaznaczenie opcji *Zachowaj z następnym akapitem* blokuje możliwość podziału strony między edytowanym a następującym po nim akapitem. Jest to szczególnie przydatne w akapitach

zawierających nagłówki rozdziałów.

Należy zwrócić uwagę, że omawiane ustawienia są jedynie sugestiami dla edytora, który będzie się starał je wypełnić. Nie zawsze jest to jednak możliwe. Jeśli na przykład ustawilibyśmy opcje *Nie dziel wierszy* oraz *Zachowuj razem z następnym akapitem* we wszystkich akapitach przygotowywanego dokumentu, to cały dokument musiałby zmieścić się na jednej stronie! W takiej sytuacji edytor dokona jednak podziałów w miejscach, które uzna za najlepsze.

Kolejne ważne opcje to *Kontroluj sieroty* i *kontroluj wdowy*. Jeśli akapit jest dzielony na strony, to *sierotami* nazywamy te wiersze akapitu, które występują przed podziałem strony, a *wdowy* — to wiersze, które znajdują się po punkcie podziału (czyli na nowej stronie). Za pomocą opcji *Kontroluj sieroty* można ustawić minimalną liczbę wierszy w akapicie, jaka musi pozostać na stronie przed podziałem. Dzięki właściwemu ustawieniu tej liczby możemy zapobiec źle wyglądającym akapitom, których pierwszy wiersz znajduje się na końcu jednej strony, a cała reszta na końcu drugiej. Przy włączonej opcji kontroli sierot cały akapit w takiej sytuacji zostanie przeniesiony na następną stronę. Opcja *Kontroluj wdowy* działa w podobny sposób lecz dotyczy minimalnej liczby wierszy akapitu na nowej stronie. Kontrolując wdowy zapobiega się wstawieniu podziału na strony w miejscu, które powodowałoby przeniesienie pojedynczego wiersza na nową stronę.

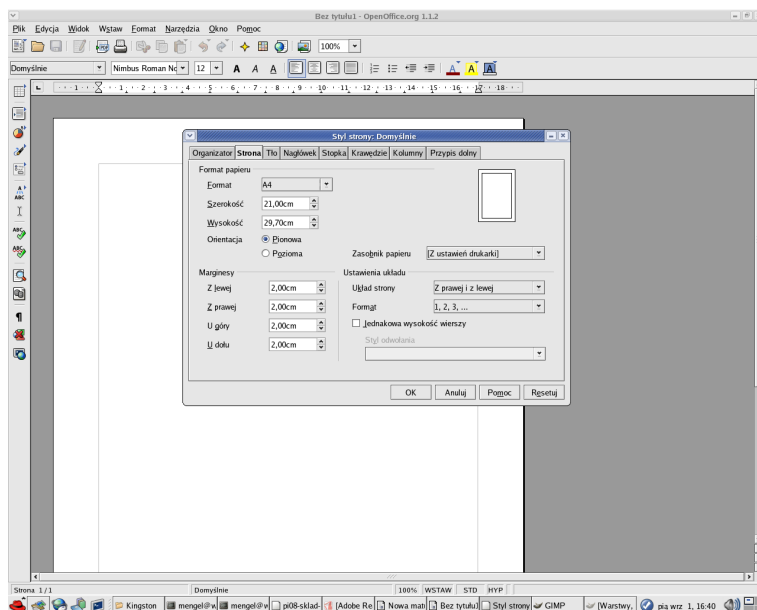
Pozostałe zakładki dotyczą odpowiednio sposobu numeracji wierszy w akapicie, miejsca ustawienia punktów wcięć (posługując się tym narzędziem możemy dokładniej ustawić wcięcia niż za pomocą linijki), wyświetlania krawędzi akapitu oraz koloru jego tła (omówionego przy okazji paska obiektów).

Ćwiczenie 9.12.1. Przygotuj dokument, który ilustrowałby:

- działanie kontroli wdów i sierot
- działanie opcji *Zachowuj razem z następnym*

Ćwiczenie 9.12.2. Poeksperymentuj z opcjami znajdującymi się na pozostałych zakładkach. Nie bój się próbować! Pamiętaj, że każdą zmianę możesz odwołać za pomocą *Edycja ► Cofnij*.

Przyjrzyjmy się teraz narzędziom do formatowania strony. Dostęp do nich uzyskujemy wybierając opcję **Format**→**Strona**. Zobaczymy wówczas na ekranie dialog *Styl strony*:



Podobnie jak poprzednio jest to dość rozbudowany dialog z wieloma zakładkami. Omówimy teraz niektóre z dostępnych opcji. Wybierzmy najpierw zakładkę *Strona*. Za jej pomocą możemy wybrać:

- format strony, na przykład A4, B5 itp.
- rozmiary strony: niezależnie od ustawień formatu
- orientację strony
- marginesy
- format numerów stron

Ponadto możemy zdecydować, czy ustawiony układ ma obowiązywać na wszystkich stronach, czy tylko na lewych lub prawych. Jest to przydatne, jeśli dokument ma być drukowany dwustronnie lub być później oprawiany.

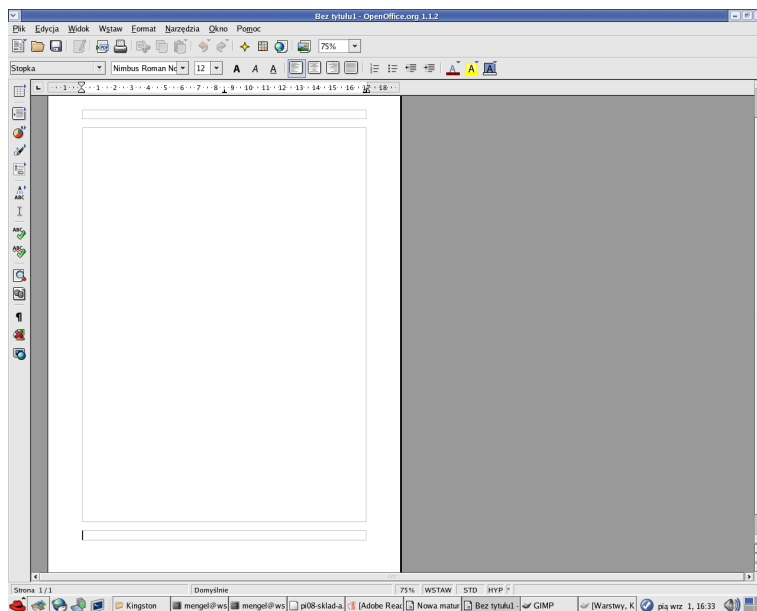
Posługując się zakładką *Tło strony* można ustawić tło strony. Tło to może być kolorem wybieranym spośród wielu dostępnych. Jeśli jednak w polu *Jako* wybierzemy *Grafika*, to tłem strony może być dowolna grafika. Należy wówczas wybrać plik z grafiką, która ma być wyświetlana na stronie i sposób jej wyświetlania. Mamy przy tym do wyboru następujące możliwości:

- Plik graficzny może być wyświetlany w wybranym miejscu strony.
- Plik graficzny może być wyświetlany w ten sposób, aby wypełnił całą stronę — wymaga to często przeskalowania: rozciągnięcia lub ściśnięcia grafiki.
- Plik graficzny może być wyświetlany w wielu egzemplarzach jedno koło drugiego tak, aby wypełnić całą stronę.

Warto przy tej okazji wspomnieć o jeszcze jednej możliwości. Tło może być połączone (opcja *Połącz* włączona) z konkretnym plikiem graficznym. Wtedy każda zmiana tego pliku powoduje zmianę tła w dokumencie. Możliwe jest również skopiowanie grafiki do dokumentu. Wtedy późniejsze zmiany w pliku graficznym nie powodują zmian w dokumencie (który przechowuje własną kopię grafiki). Nie ma też wówczas konieczności dołączania pliku graficznego do dokumentu, gdy chcemy go na przykład przekopiować na inny komputer lub przesłać pocztą elektroniczną.

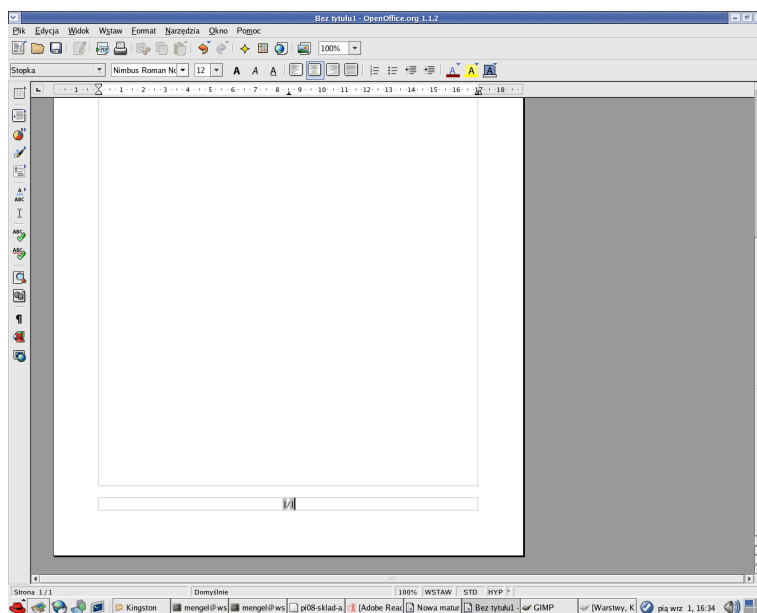
Kolejnym ważnym i przydatnym zagadnieniem są nagłówki i stopki. Nagłówek to fragment strony, który pojawia się na górze przed właściwym tekstem. Stopka to część strony występująca na samym jej dole. Domyślnie nagłówki i stopki są wyłączone, ale można je (niezależnie od siebie) włączyć w zakładkach *Nagłówek* i *Stopka* dialogu *Styl strony*. Opcje na tych zakładkach stają się dostępne po zaznaczeniu pola *Włącz nagłówek* lub *Włącz stopkę* w przypadku stopki. Służą one przede wszystkim do sterowania położeniem stopki i nagłówka na stronie. Jest jednak jedna ważna opcja: *Ta sama zawartość na stronach parzystych i nieparzystych*. Dzięki niej można różnicować tekst występujący w nagłówkach i stopkach na stronach parzystych i nieparzystych. Z możliwości tej chętnie korzystają redaktorzy książek, umieszczając w nagłówkach stron parzystych na przykład tytuł aktualnej części, a na stronach nieparzystych — tytuł aktualnego rozdziału.

Po włączeniu stopek/nagłówków na stronie pojawiają się dodatkowe obszary na górze/dole strony:



Tekst wpisany do tych obszarów będzie się pojawiał w nagłówku/stopce każdej strony. Ponieważ najczęściej chcemy, aby tekst ten zawierał pewne informacje zależne od strony, na której się znajduje (choćby numer strony), więc przygotowując nagłówek/stopkę posługujemy się *polami*.

Pola wstawia się za pomocą *Wstaw ► Pola*. Jest wiele różnych pól. Na przykład pole *Numer strony* oznacza numer aktualnej strony, pole *liczba stron* to liczba stron w dokumencie. Można również wstawić pole z datą, autorem czy tytułem. Przyjrzyjmy się bliżej przykładowemu nagłówkowi. Przypuśćmy, że chcemy, aby w nagłówku każdej stronie pojawiała się informacja o numerze strony i łącznej liczbie stron w dokumencie, na przykład w postaci: *1/30* na pierwszej stronie, *2/30* na następnej itd. Wtedy przygotowujemy nagłówek zawierający po kolei: pole *Numer strony*, znak „/” oraz pole *Liczba stron*:



Zauważmy, że wstawione pola są zacieniowane (jeśli jest włączona opcja *Widok ► Cieniowanie pól*). Cieniowanie oznacza, że napis nie został wstawiony bezpośrednio przez użytkownika lecz „wyliczony” przez edytor na podstawie dostarczonych przez użytkownika informacji (takich jak wstawione pole). Oczywiście cieniowanie, tak jak znaki niedrukowalne, jest widoczne jedynie na ekranie i w drukowanej wersji dokumentu nie będzie występować.

Pola zawierające bardziej rozbudowane informacje są dostępne po wyborze opcji *Wstaw ► Inne*. Ukaże się wówczas dialog, za pomocą którego mamy dostęp do różnych pól pogrupowanych względem typu. Na przykład wybierając typ *Rozdział* możemy wybrać wyświetlenie nazwy rozdziału, numeru rozdziału lub numeru i nazwy rozdziału. Dokładne omówienie znaczenia poszczególnych pól wykracza jednak poza ramy tego podręcznika.

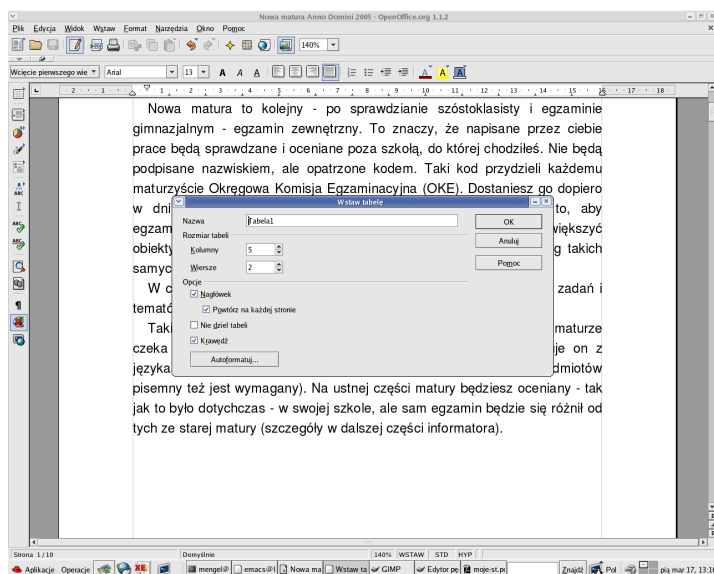
Jeszcze jednym elementem układu strony, który może na niej wystąpić są *przypisy*. Przypisy wstawiamy za pomocą opcji *Wstaw ► Przypis dolny*. W wyświetlonym dialogu możemy wybrać sposób numeracji przypisów: automatyczny lub ręczny za pomocą wybranego znaku oraz sposób jego wyświetlania. Jeśli zdecydujemy się na przypis dolny, to treść przypisu będzie wyświetlona na dole strony, pod tekstem, ale nad stopką (o ile jest włączona). Przypisy końcowe są składane na końcu tekstu.

9.13 Tabele

W praktycznie każdym większym dokumencie znajdują się tabele. Dzięki nim można przedstawić w sposób syntetyczny i przejrzysty informacje takie jak listy, zestawienia itp. Zapoznamy się teraz z podstawowymi operacjami na tabelach: wstawianiem, usuwaniem, edycją wierszy i kolumn oraz zmianą ich rozmiarów, a także łączeniem komórek.

- Wstawianie
 - za pomocą menu *Wstaw*
 - za pomocą menu narzędzi
- Edycja wyglądu tabeli:
 - usuwanie i wstawianie wierszy
 - usuwanie i wstawianie kolumn
 - scalanie i podział komórek
 - ustawianie parametrów krawędzi oddzielających poszczególne komórki od siebie
 - ustawianie tła komórek

Tabele można wstawić do dokumentu na dwa sposoby. Pierwszy to wybór opcji *Wstaw* ► *Tabela*. Pojawia się dialog, za pomocą którego decydujemy o rozmiarze tabeli oraz o jej wyglądzie.

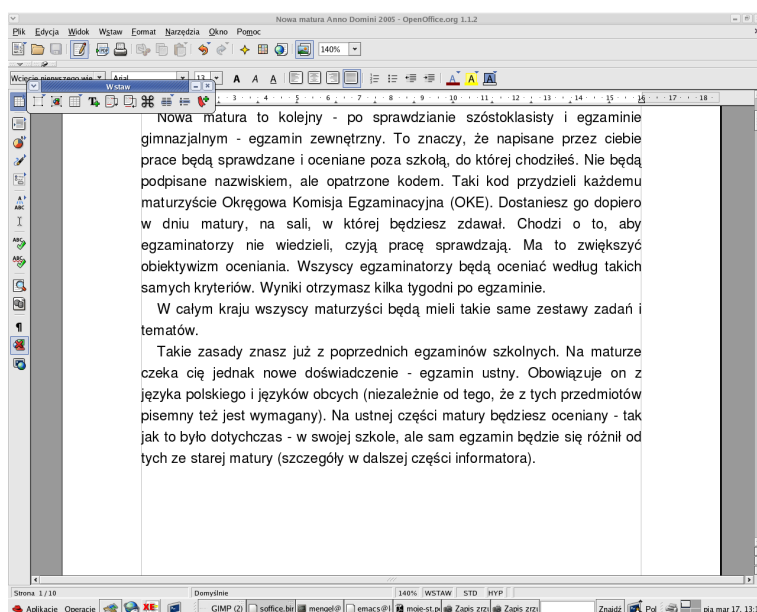


Oprócz standardowego ustawienia liczby wierszy i kolumn w tabeli można za pomocą opcji *Nagłówek* zdecydować, że pierwszy wiersz tabeli to jej nagłówek. Domyślny sposób składania nagłówka jest inny niż w sposób składaniu poszczególnych komórek tabeli. Zaznaczenie opcji *Powtórz* na

każdej stronie powoduje, że w przypadku tabel rozciągających się na kilka stron, nagłówki będą powtarzane na początku każdej strony.

Za pomocą opcji *Nie dziel tabeli* możemy zabronić dzielenia tabeli na strony, a za pomocą opcji *Krawędzie* decydujemy, czy krawędzie tabeli mają być widoczne. Po kliknięciu przycisku *OK* w miejscu położenia kursora w dokumencie jest wstawiana tabela o parametrach ustawionych w dialogu. Jeśli wyświetlanie krawędzi tabeli zostało wyłączone, to krawędzie tabeli są jedynie zaznaczone w postaci szarych kresek. Pomagają one podczas edycji i będą niewidoczne w drukowanej wersji dokumentu.

Drugi sposób wstawiania tabeli polega na wykorzystaniu paska narzędzi po lewej stronie. Klikamy na pierwszą ikonę w tym pasku (pojawia się wówczas dialog *Wstaw*):



i trzymając wciśnięty ten klawisz wybieramy trzecią ikonkę (*wstaw tabelę*). Cały czas trzymamy wciśnięty klawisz myszki!. Następnie rysujemy myszką rozmiar tabeli i dopiero gdy będzie on zgodny z naszym życzeniem puszczyemy lewy klawisz myszki.

Wprowadzanie tekstu do poszczególnych komórek tabeli najłatwiej przeprowadzić ustawiając kursor w odpowiedniej komórce (klikając na nią), a następnie wprowadzając tekst w zwykły sposób. Każda komórka zachowuje się przy tym tak, jak mały akapit: można ustawić wszystkie jej parametry za pomocą opcji *Akapit* tak, jak w przypadku zwykłego akapitu.

Zauważmy, że w chwili gdy kursor znajdzie się wewnątrz tabeli zmienia się pasek obiektów. Przybiera on teraz następującą postać oferując operacje na tabeli:



- grupa przycisków do zmiany trybu edycji tabeli. Decyduje jak zachowują się sąsiednie elementy tabeli, gdy wykonujemy operację wstawiania/usuwania wiersza lub kolumny. Omówienie tych parametrów wykracza poza ramy tego podręcznika, trzeba jednak wiedzieć, że ustawienie niektórych trybów może w pewnych okolicznościach uniemożliwiać wstawianie nowych kolumn. Jeśli zatem przycisk służący do wstawiania staje się nieaktywny, to warto zmienić tryb edycji tabeli.
- grupa przycisków do operacji na komórkach. Pierwszy przycisk służy do scalania komórek, drugi do ich dzielenia, trzeci do wykonywania czynności optymalizujących wygląd takich jak: równomierne rozmieszczanie wierszy i kolumn tabeli czy też ustawianie ich optymalnej wysokości/szerokości.
- grupa przycisków do wstawiania/usuwania wierszy i kolumn.
- grupa przycisków do sterowania wyglądem krawędzi i kolorem poszczególnych komórek tabeli.

Edycja tabeli to, oprócz wpisywania do niej tekstu, także czynności polegające na jej edycji graficznej: ustawianie szerokości kolumn i wysokości wierszy, dokonywanie scalenia komórek i ich podziału.

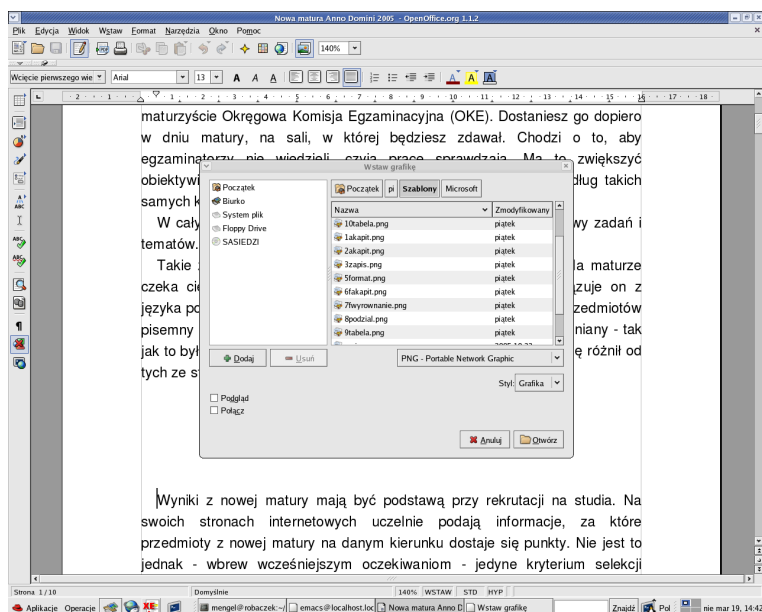
Zmianę szerokości kolumn wykonujemy ustawiając kursor myszy na pionowej krawędzi oddzielającej dwie kolumny od siebie. Gdy kursor zmienia kształt na podwójną strzałkę należy wcisnąć lewy klawisz myszki i pociągnąć linię w lewo lub w prawo. Inny sposób to kliknięcie prawym przyciskiem myszy na edytowaną kolumnę, a następnie ustawienie jej szerokości za pomocą opcji *Kolumna* ► *Szerokość*. Można też wybrać opcję *Szerokość optymalna*, która polega na automatycznym doborze szerokości kolumny tak, aby zmieścił się w niej tekst wpisany w poszczególne komórki. W podobny sposób dokonujemy ustawienia wysokości wiersza (wybieramy wówczas opcję *Wiersz*).

Jeśli chcemy składać bardziej złożone tabele, to może pojawić się konieczność połączenia kilku komórek w jedną. Odbywa się to poprzez zaznaczenie komórek przeznaczonych do połączenia, kliknięcie na zaznaczenie prawym klawiszem myszy i wybór opcji *Komórki* ► *Scal*. Opcje te są także dostępne za pośrednictwem omówionego poprzednio paska obiektów.

9.14 Grafika

Dokumenty tworzone w pakiecie *OpenOffice* mogą zawierać grafikę.

Aby wstawić obiekt graficzny do dokumentu należy wybrać z menu *Wstaw* opcję *Grafika*. Pojawia się wówczas dialog *Z pliku*, za pomocą którego możemy wybrać format wstawianej grafiki i plik, w której się ona znajduje.

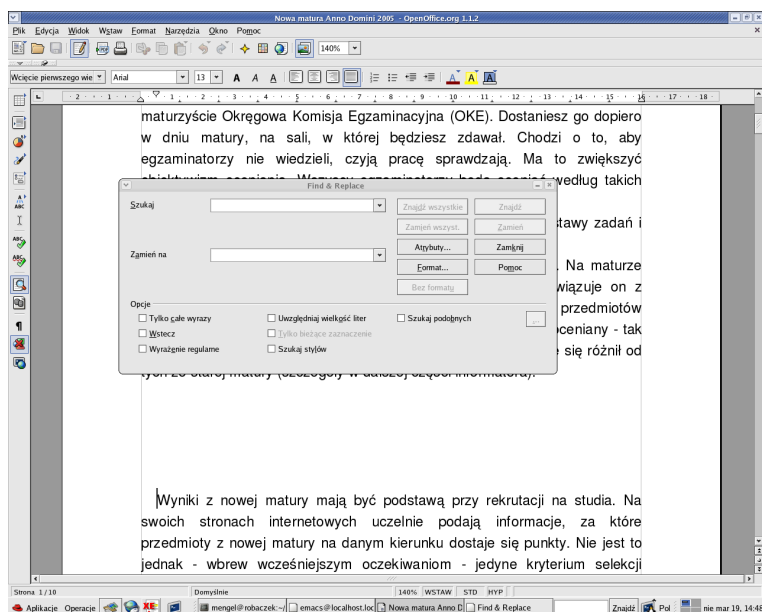


9.15 Wyszukiwanie wzorca w tekście

Bardzo ważnym narzędziem ułatwiającym edytowanie dokumentów jest wyszukiwanie wzorca oraz jego zastępowanie w tekście.

- uruchamianie
 - menu *Edycja* opcja *Znajdź i zastąp*
 - klawisz *Ctrl+F*
 - lupa z menu podręcznego po lewej stronie
- przeznaczenie
 - znajdowanie wszystkich wystąpień słowa lub jego fragmentu w tekście
 - znajdowanie słów pasujących do podanego wzorca
 - znajdowanie akapitów i fragmentów tekstu z ustawionymi wskazanymi atrybutami
 - zastępowanie znalezionych wystąpień pewnego słowa innym słowem

Dostęp do tego narzędzia zyskujemy wybierając *Edycja* ► *Znajdź i zastąp*, klikając ikonkę z lupą z lewego paska narzędziowego lub też po prostu wciskając *Ctrl+F*. Zobaczmy wówczas na ekranie następujący dialog:

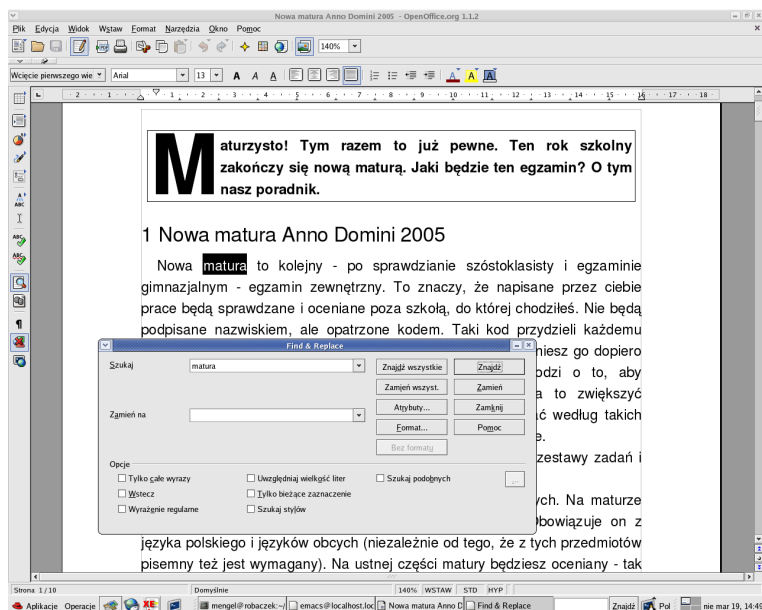


9.15.1 Wyszukiwanie

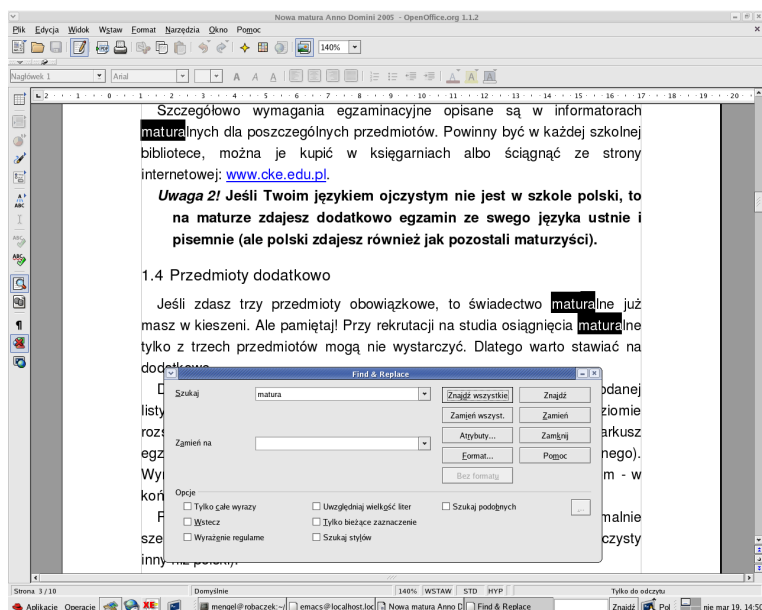
Wyszukiwanie wzorca może przebiegać w różny sposób. Jest możliwe:

- wyszukiwanie jednego wystąpienia wzorca
 - w całym dokumencie od miejsca kursora w przód
 - w całym dokumencie od miejsca kursora w tył
 - w zaznaczonym fragmencie tekstu
- wyszukiwanie wszystkich wystąpień wzorca
- wyszukiwanie pełnych wyrazów
- ignorowanie/uwzględnianie wielkości liter
- wyszukiwanie wzorców podobnych
 - tej samej długości, różniących się określoną liczbą znaków
 - różnej długości

Najprostszą funkcjonalnością omawianego dialogu jest wyszukanie wystąpienia pewnego wzorca w tekście. Przypuśćmy, że chcemy znaleźć wszystkie wystąpienia słowa „matura” w edytowanym tekście. W tym celu włączamy dialog *Znajdź i zastąp* i w polu *Szukaj* wpisujemy tekst, który chcemy znaleźć. Następnie klikamy przycisk *Znajdź* lub *Znajdź wszystkie*. Edytor rozpocznie przeszukiwanie dokumentu od miejsca, w którym znajdowaliśmy się w chwili uruchomienia dialogu i podświetli pierwsze wystąpienie szukanego słowa, jeśli wybraliśmy opcję *Znajdź*:

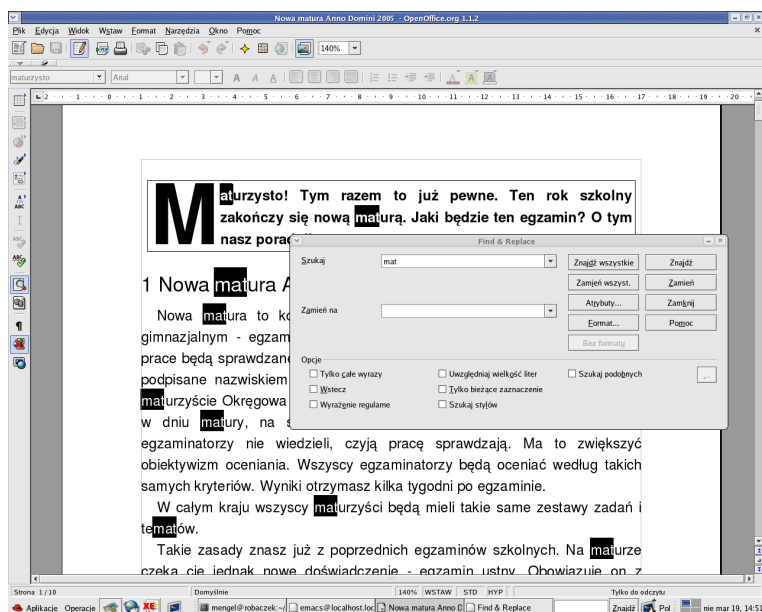


lub wszystkie wystąpienia tego słowa w przypadku wciśnięcia klawisza *Znajdź wszystkie*.



Jeśli wyszukiwane słowo nie występuje w dalszej części tekstu, to edytor wyświetli komunikat pytający, czy kontynuować wyszukiwanie słowa od początku dokumentu.

Warto zwrócić uwagę, że szukać możemy nie tylko pełnych słów, ale także dowolnych ich fragmentów. Jeśli spróbujemy szukać wzorca „mat”, to edytor znajdzie nie tylko słowo „mat”, ale także słowa „matura”, „maturalna”, „maturalny”, „matematyka”, „matematyczny”, w każdym z nich podświetlając cząstkę „mat”.



Takie zachowanie systemu można wykorzystać chcąc znaleźć wszystkie wystąpienia jakiegoś rzeczownika, np.: „matura” w różnych przypadkach. Wpiszemy wtedy po prostu rdzeń „matur” i znajdziemy „matura”, „maturę”, ale także „Maturzystę” i „maturalny”.

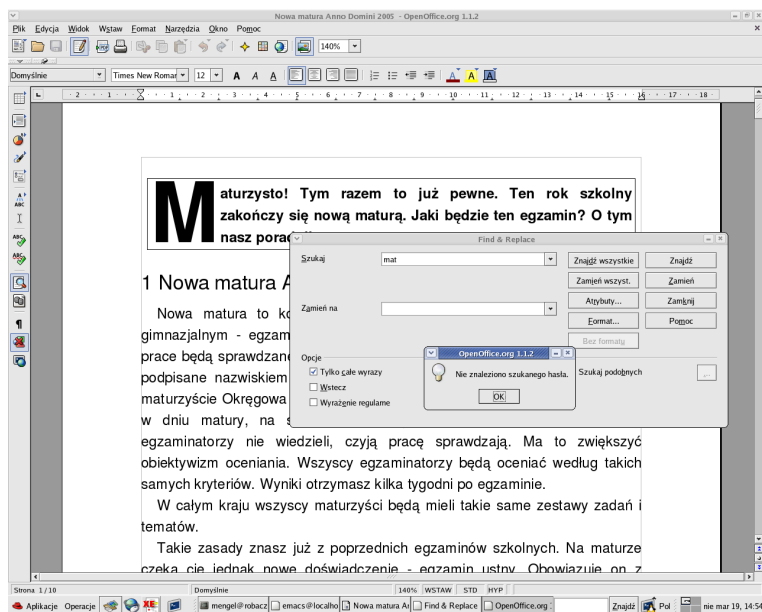
Dialog *Znajdź i zastąp* pozwala na sterowanie procesem przeszukiwania i zmianę opisanego powyżej zachowania domyślnego.

Można na przykład szukać od miejsca kursora wstecz, czyli w stronę początku dokumentu zamiast w przód. Dzięki temu można łatwo znaleźć poprzednie wystąpienie podanego wzorca. Aby spowodować takie zachowanie systemu wystarczy zaznaczyć pole *Wstecz* w dialogu.

Inna modyfikacja zachowania wyszukiwania polega na ograniczeniu jego zakresu. Zamiast przeszukiwać cały dokument możemy przeszukać tylko jego fragment. W tym celu należy najpierw zaznaczyć fragment, w którym będzie się odbywać przeszukiwanie, następnie postępować tak, jak przy zwykłym wyszukiwaniu opisanym powyżej. Jedyna różnica polega na tym, że należy zaznaczyć pole *Tylko bieżące zaznaczenie*.

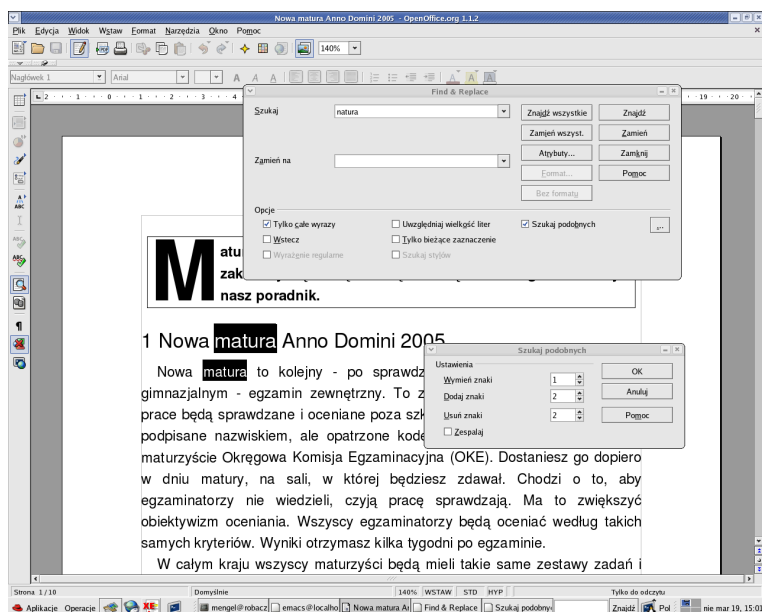
Dialog znajdź i zastąp zapamiętuje ustawienia. Jeśli dokonaliśmy przeszukiwania we fragmencie tekstu i później ponownie uruchomimy dialog znajdź i zastąp, to pole Tylko bieżące zaznaczenie w dalszym ciągu jest zaznaczone. Aby szukać w całym dokumencie należy pamiętać o jego odznaczeniu. To samo dotyczy pozostałych pól.

Można także zaznaczyć pole *Tylko całe wyrazy*. Wtedy szukając wzorca „mat” znajdziemy jedynie wyrazy „mat”, a nie wszystkie wystąpienia „mat” jako części wyrazów.



Z kolei zaznaczenie pola *Uwzględniaj wielkość liter* powoduje zmianę domyślnego zachowania polegającego na ignorowaniu wielkości liter. Domyślnie szukając słowa „matura” znajdziemy także słowo „Matura”, gdyż program ignoruje wielkość poszczególnych liter. Po zaznaczeniu tej opcji wyszukane zostaną jedynie te słowa, które dokładnie pasują do szukanego wzorca.

Ciekawą możliwość oferuje opcja *Szukaj podobnych*. Zaznaczając to pole, a następnie klikając przycisk z wielokropkiem dostajemy możliwość wyszukiwania nie tylko wprowadzonego wzorca, ale także wyrazów „podobnych” do wprowadzonego wzorca. Co to znaczy, że słowo jest „podobne” definiujemy przy tym sami. Możemy uznać za podobne słowo, które różni się od wzorca wskazaną liczbą znaków (np. dwoma jak słowa „kura” i „pula”) albo słowo, które powstaje z wzorca przez usunięcie lub dodanie wskazanej liczby znaków.



Należy także wspomnieć, że wszystkie opisane powyżej opcje można ze sobą łączyć.

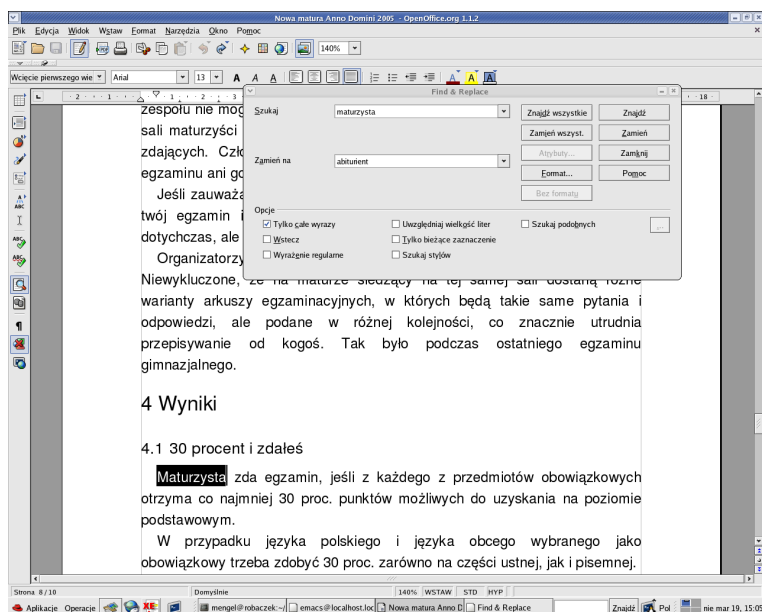
9.15.2 Zastępowanie wzorca w tekście

Czasami zachodzi potrzeba zastąpienia jednego słowa innym w całym tekście.

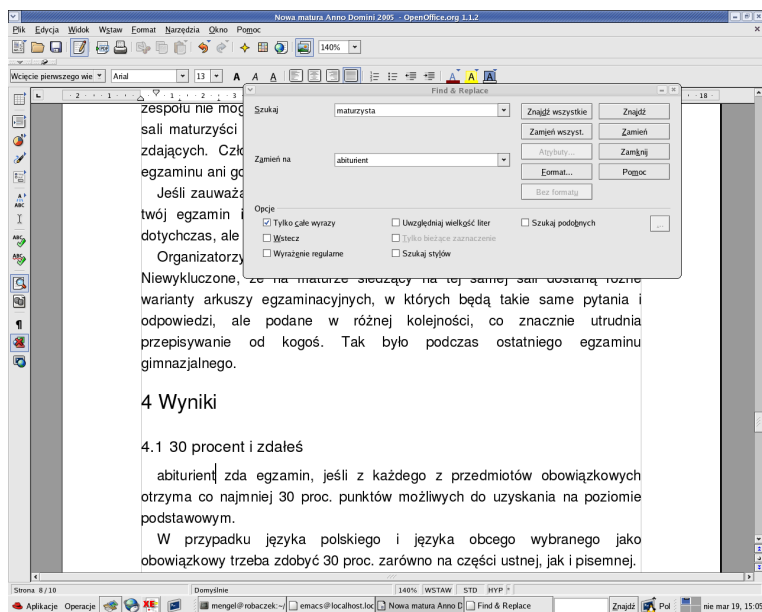
- Dialog *Znajdź i zastąp* umożliwia także zastąpienie znalezionych słów innymi
- Jest możliwa zamiana
 - *wszystkich* wystąpień wzorca
 - *wybranych* wystąpień wzorca

Automatyczną zamianę wszystkich wystąpień należy stosować z dużą ostrożnością.

O ile wyszukiwanie pozwala znaleźć wystąpienia podanego wzorca w tekście, o tyle zamiana umożliwia automatyczne zastąpienie wprowadzonego tekstu innym lub też jego usunięcie. Przypuśćmy, że chcemy zmienić każde wystąpienie słowa „maturzysta” słowem „abiturient”. Wprowadzamy wtedy w dialogu *Znajdź i zastąp* słowo „maturzysta” do pola *Szukaj*, a słowo „abiturient” do pola *Zmień na*:



Następnie klikamy przycisk *Zmień*. Program odszuka pierwsze (za miejscem w którym znajdował się kursor) wystąpienie słowa „maturzysta” i podświetli je nie dokonując na razie zamiany. Jeśli ponownie klikniemy przycisk *Zmień*, to słowo zostanie podmienione słowem abiturient, a system podświetli kolejne wystąpienie słowa maturzysta.



Jeśli chcemy także i to wystąpienie zamienić klikamy klawisz *Zmień*. Jeśli chcemy to wystąpienie zachować w oryginalnej postaci i przejść do kolejnych wystąpień wzorca to zamiast *Zmień* klikamy klawisz *Znajdź*. Zauważmy, że jeśli pole *Zmień na* będzie puste, to tego rodzaju zamiana ma po prostu efekt usunięcia wyszukiwanego słowa.

Istnieje również możliwość zamiany wszystkich pozostałych wystąpień szukanego słowa nowym. Stanie się tak, jeśli zamiast klawisza *Zmień* klikniemy

my przycisk *Zamień wszystko*. Komputer dokona wówczas zamiany wszystkich wystąpień wskazanego słowa i wyświetli informację o tym ile ich było. Taka możliwość jest jednak dość niebezpieczna i należy ją stosować z dużą ostrożnością, pamiętając o tym, że nie zawsze efekt takiej automatycznej zamiany może być zgodny z naszymi zamierzeniami. Na szczęście zawsze pozostaje możliwość wycofanie tych zamian za pomocą opcji *Cofnij* z menu *Edycja*.

Rozdział 10

Strukturalizacja dokumentu i złożone operacje na tekście

10.1 Strukturalizacja dokumentu

Podczas pisania dłuższych dokumentów przygotowywanych w długim okresie czasu, trudno jest zachować spójne zasady składu poszczególnych elementów. Często zdarza się, że występujące w dokumencie jednostki takie jak przykłady, raz są złożone czcionką pochyłą, a innym razem są wyróżniane w inny sposób. Może się też okazać, że tytuły podrozdziałów raz są złożone czcionką rozmiaru 16pt, a innym razem czcionką rozmiaru 14pt. Pracując nad tekstem przez długi czas, trudno jest zapamiętać wszystkie przyjęte konwencje typograficzne i jednolicie je stosować.

W długich dokumentach, których struktura ulega częstym zmianom, dodatkowym problemem jest także spójna numeracja rozdziałów, podrozdziałów, punktów i podpunktów. Gdy struktura pisanej pracy zmieni się w trakcie jej powstawania i na przykład po rozdziale 3. a przed dotychczasowym 4. wstawimy nowy rozdział, to zmieni się numeracja wszystkich rozdziałów od czwartego począwszy. Jeśli gdzieś w tekście umieszczono odsyłacz do punktu znajdującego się w rozdziale 5 (typu *patrz punkt 5.3*), to teraz trzeba pamiętać o zmianie tego odsyłacza na *6.3*. Podobne problemy występują, gdy numerujemy przykłady, zadania, definicje i odwołujemy się do nich w innych fragmentach tekstu.

Aby poradzić sobie z tego typu problemami, warto zapoznać się z bardziej złożonymi narzędziami do tworzenia dokumentów elektronicznych: stylami, konspektem, odsyłaczami i bibliograficzną bazą danych.

- Motywacja
 - ujednoczenie poszczególnych elementów dużych dokumentów
 - skupienie uwagi na strukturze, a nie formatowaniu
 - łatwiejsze przygotowywanie prac zbiorowych
 - łatwiejsze modyfikacje wyglądu dokumentu

- Narzędzia do tworzenia struktury
 - konspekt
 - style
 - baza bibliograficzna
- Dodatkowe zalety strukturalnego dokumentu:
 - automatyczne spisy
 - automatyczna numeracja rozdziałów

W kolejnych częściach tego rozdziału przedstawimy poszczególne mechanizmy i pokażemy na przykładzie, jak zastosować je w tworzonej dokumentacji.

10.2 Style i ich stosowanie

Podstawowym narzędziem ułatwiającym tworzenie spójnych dokumentów są *style*. Styl to po prostu zbiór ustawień różnych atrybutów, który można zastosować do wybranego fragmentu tekstu. W zależności od tego, czy fragmentem tym jest grupa pojedynczych znaków, akapit, strona czy numeracja mówimy o stylu znaków, akapitu, strony lub numeracji.

- styl definiuje atrybuty dotyczące fragmentu tekstu
- styl może dotyczyć:
 - pojedynczego znaku lub ciągu znaków — *styl znaków*
 - akapitu — *styl akapitu*
 - sposobu numeracji przy wyliczeniach, w nagłówkach rozdziałów itp — *styl numeracji*
 - strony — *styl strony*
 - ramki — *styl ramki*
- niektóre style (standardowe) są wstępnie zdefiniowane
- użytkownik może tworzyć własne style i modyfikować niektóre style standardowe

Formatując określony fragment tekstu możemy zlecić złożenie go określonym stylem, zamiast pracować ustawiając za każdym razem rozmiar czcionki, jej krój, kolor, wielkości wcięć itp. Taka metoda postępowania daje dużą oszczędność czasu — można wtedy jednym kliknięciem ustawić wiele parametrów. Ponadto, mamy gwarancję, że nie zapomnimy ustawić żadnego parametru i każdy fragment złożony tym samym stylem będzie miał identyczny wygląd. Nie bez znaczenia jest także duża elastyczność takiej metody. Wyjaśnijmy to na przykładzie. Powiedzmy, że w dokumencie chcemy

stosować wyróżnienia pewnych kluczowych wyrazów. Możemy oczywiście za każdym razem składać te wyrazy kursywą. Jeśli jednak później dojdziemy do wniosku, że lepiej byłoby zamiast stosować kursywę zmienić rodzaj czcionki, to musimy przejrzeć cały dokument i pracowicie zamienić sposób wyróżnienia każdego wyrazu. Lepsza metoda postępowania polega na zdefiniowaniu stylu o nazwie, na przykład *wyróżnienie*. Styl ten może nakazywać złożenie określonego fragmentu tekstu na przykład pochylą czcionką *arial* koloru czerwonego na białym tle.

Przygotowując dokument składamy słowa, które chcemy wyróżnić stosując do nich styl *wyróżnienie*. Spowoduje to odpowiednie ustawienie atrybutów tego słowa (a więc czerwonej pochylonej czcionki *arial*). Oczywiście wszystkie fragmenty tekstu, do których zastosujemy styl *wyróżnienie* zostaną złożone dokładnie w ten sam sposób. Jeśli później zmienimy definicję stylu *wyróżnienie*, rezygnując na przykład z czerwonego koloru, to nie musimy w ogóle zmieniać dokumentu! Wystarczy zmiana definicji stylu, a wszystkie wyróżnione miejsca zmienią automatycznie swój wygląd.

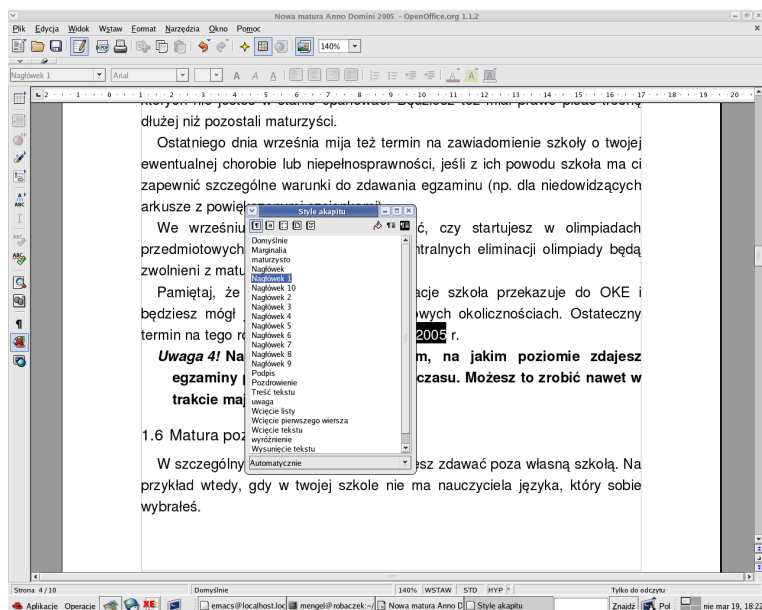
Przygotowanie dokumentu możemy zatem rozpocząć od zdefiniowania pewnej liczby stylów. Dobieramy ich nazwy tak, aby przypominały nam, do czego zostały zaprojektowane. Definiujemy na przykład styl *wyróżnienie* do wyróżniania fragmentów tekstu, styl *przykład* do składania treści przykładów, styl *cytat* do składania cytatów itd. Następnie pisząc dokument skupiamy się już wyłącznie na jego strukturze. Gdy wprowadzamy tekst przykładu nie zastanawiamy się jakiego formatowania użyć, a jedynie informujemy edytor, że właśnie tworzymy przykład, czyli że do jego formatowania należy użyć stylu o nazwie *przykład*.

Naszkiecowana metoda postępowania szczególnie dobrze sprawdza się przy pisaniu pracy zbiorowej. Autorzy po prostu ustalają wspólnie style i ich nazwy, a następnie konsekwentnie je stosują pisząc swoje fragmenty pracy. Powstający dokument będzie miał dzięki zastosowaniu stylów jednolitą postać graficzną.

Podstawowym narzędziem pracy ze stylami jest *stylista*.

- Stylistę wywołujemy i chowamy klawiszem *F11*.
- Początkowo stylista zawiera jedynie zdefiniowane wstępnie style.
- Za pomocą stylisty można tworzyć nowe style, edytować je i usuwać oraz stosować wybrane style do zaznaczonych fragmentów dokumentu.

Kolejne wciśnięcia klawisza *F11* naprzemiennie włączają i ukrywają okienko stylisty. Przyjrzyjmy się bliżej poszczególnym jego elementom:



Po lewej stronie paska znajdującego się na górze okienka stylisty znajdują się przyciski, za pomocą których możemy wyświetlać odpowiednio: style akapitów, znaków, ramek, stron i numeracji. Najczęściej korzysta się ze stylów znaków i akapitów i nimi właśnie zajmiemy się w dalszej części podręcznika.

Po prawej stronie na górze okienka stylisty znajdują się trzy przyciski, które ułatwiają stosowanie stylów w dokumencie i definiowanie nowych stylów. Zajmiemy się nimi nieco później, na razie upewnijmy się, że żaden z tych przycisków nie jest wciśnięty.

Na dole okienka stylisty znajduje się lista wyboru, za pomocą którego decydujemy, jakie style są wyświetlane w styliszcie. Możemy wyświetlić na przykład wszystkie dostępne style, jedynie style już użyte w bieżącym dokumencie lub też jedynie style niestandardowe, czyli zmodyfikowane lub zdefiniowane przez użytkownika.

W głównej części okienka stylisty są wyświetlane nazwy istniejących stylów. Klikając na dowolny z nich prawym klawiszem myszy mamy do wyboru dwie czynności: *Modyfikuj* lub *Nowy*. W przypadku stylów zdefiniowanych przez użytkownika dochodzi jeszcze jedna możliwość: *Usuń*. Modyfikacja to po prostu edycja danego stylu, czyli zmiana ustawień atrybutów, które ten styl definiuje. Wybór opcji *Nowy* daje nam możliwość stworzenia nowego stylu bazującego na zaznaczonym. Tymi możliwościami zajmiemy się za chwilę, najpierw jednak zobaczymy jak stosować style do określonych fragmentów tekstu.

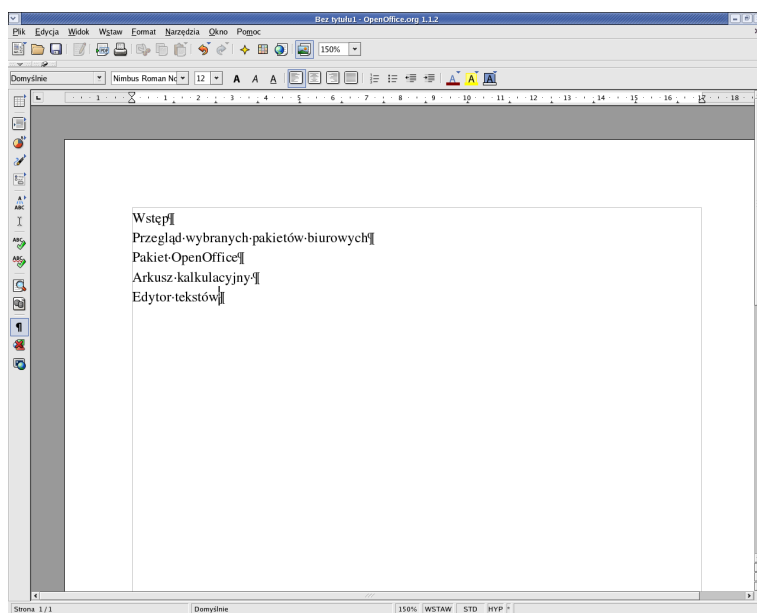
Sposób zastosowania stylu do wybranego fragmentu dokumentu zależy od tego, czy używamy stylu znaków czy stylu akapitu. W każdym przypadku należy jednak najpierw włączyć okienko stylisty. Dobrze jest też upewnić się, że stylista automatycznie decyduje, jakie style wyświetlać (na dole okienka widnieje napis *Automatycznie*).

Sposób zastosowania stylu w dokumencie zależy od jego rodzaju:

- Styl znaków
 - zaznacz fragment, do którego chcesz zastosować styl
 - za pomocą przycisków w stylisście wybierz styl znaków
 - podwójnie kliknij na żądany styl znaku w stylisście
- Styl akapitu
 - ustaw się w akapicie, którego styl chcesz zmienić
 - za pomocą przycisków w stylisście wybierz styl akapitu
 - podwójnie kliknij na żądany styl akapitu w stylisście

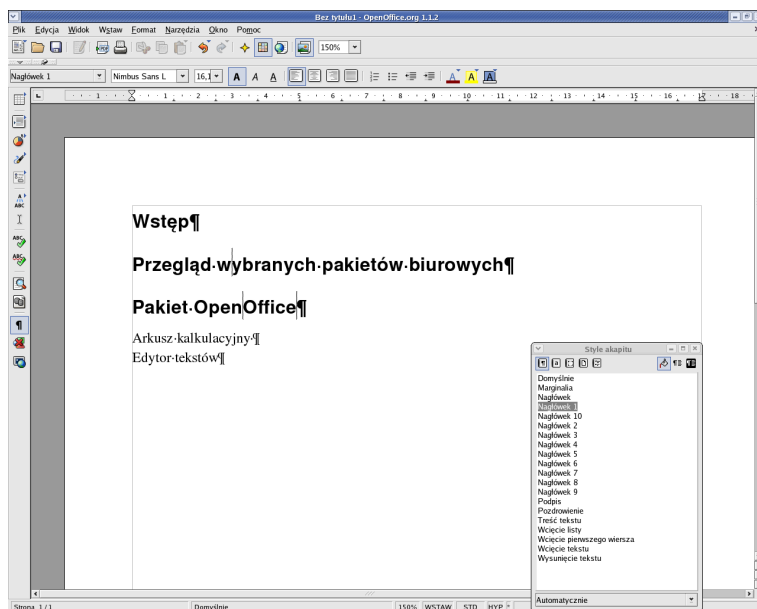
Popatrzmy, jak w praktyce wygląda praca ze stylami akapitów. Przypuśćmy, że przygotowujemy dokument o różnych pakietach biurowych. Powiedzmy, że dokument rozpocznie się od rozdziału „Wstęp”, następny rozdział będzie zatytułowany „Przegląd wybranych pakietów biurowych”, a kolejny będzie dotyczył pakietu *OpenOffice* i będzie składał się z dwóch podrozdziałów: „Arkusz kalkulacyjny” oraz „Edytor tekstów”.

Przygotujmy tekst zawierający tytuły tych części jako pojedyncze akapity. Nie wyróżniamy nic w tym dokumencie, nie modyfikujemy rozmiaru czcionki. Przy włączonym podglądzie znaków niedrukowalnych dokument powinien wyglądać następująco:



Zastosujmy teraz styl o nazwie *Nagłówek 1* do akapitu zawierającego tytuł pierwszego rozdziału. Zauważmy, że rozmiar czcionki został powiększony i jednocześnie czcionka została wytłuszczzona. Zastosujmy ten sam styl do tytułów rozdziałów. Tym razem jednak wykorzystajmy inną metodę, tzw. *tryb wypełniania formatem*. W tym celu klikamy przycisk z wiaderkiem u góry stylisty i upewniamy się, że w stylisście podświetlony jest styl o nazwie *Nagłówek 1*.

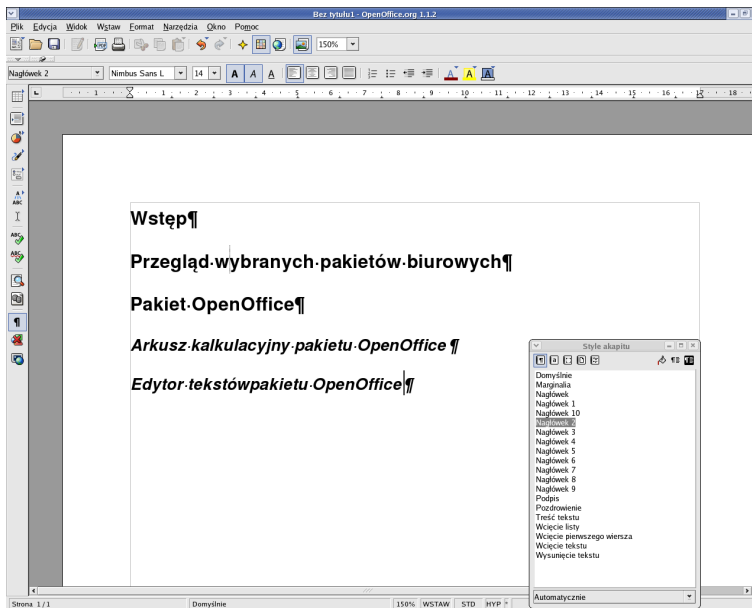
Następnie przemieszczamy kursor myszy do akapitu z tekstem „Przegląd wybranych pakietów biurowych”. Zauważmy że kursor ma postać wiaderka. Oznacza to, że kliknięcie w dowolnym miejscu akapitu spowoduje zastosowanie do niego podświetlonego w styluście stylu. Kliknijmy więc na wszystkie tytuły rozdziałów (ale nie podrozdziałów!). Powinniśmy otrzymać dokument o następującym wyglądzie:



Na koniec wyłączmy tryb wypełniania formatem, gdyż przeszkadza on przy innych czynnościach edycyjnych. Przy okazji zwróćmy uwagę na jeszcze jeden element. Nazwa stylu zastosowana do akapitu, w którym aktualnie znajduje się kursor jest wyświetlana na pasku obiektów w głównym okienku edytora.

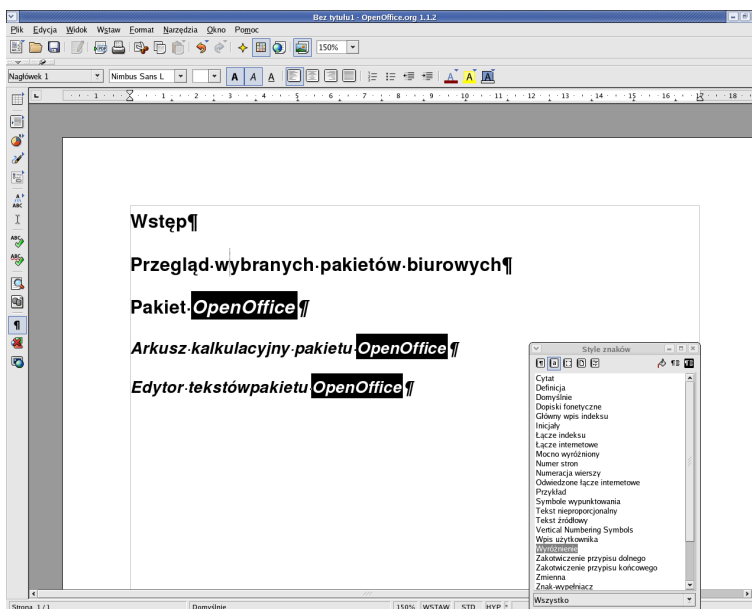
Ćwiczenie 10.2.1. Zastosuj teraz style o nazwie *Nagłówek 2* do tytułów podrozdziałów: „Arkusz kalkulacyjny” i „Edytor tekstów”. W razie kłopotów pamiętaj o poleceniu *Edycja* ► *Cofnij*.

Teraz zajmijmy się stylami znaków. Zmieńmy tytuły podrozdziałów na „Arkusz kalkulacyjny pakietu OpenOffice” oraz „Edytor tekstów pakietu OpenOffice”. Dokument wygląda następująco:



Przypuśćmy teraz, że każde wystąpienie słowa „OpenOffice” chcemy składać w ten sam sposób stosując styl znaków o nazwie *Wyróżnienie* — jest to styl wstępnie zdefiniowany, formatujący tekst w postaci kursywy.

W tym celu wyświetlamy stylistę, wybieramy w jego nagłówku przycisk wyświetlający style znaków i znajdujemy na liście styl *wyróżnienie*. Zaznaczamy w tekście wszystkie wystąpienia napisu „OpenOffice” i podwójnie klikamy na nazwę stylu *Wyróżnienie*.



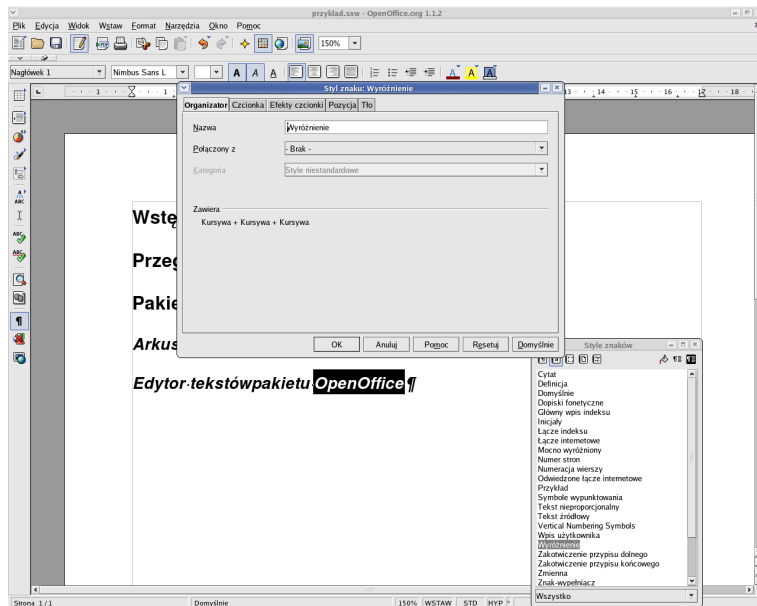
Zauważmy, że napis zmienił się jedynie w nagłówku rozdziału, gdyż w nagłówkach podrozdziału był już złożony kursywą. Niemniej jednak, jak się niebawem przekonamy, edytor zapamiętał, że zastosowaliśmy styl także w nagłówkach podrozdziałów.

10.3 Modyfikacja stylu znaków

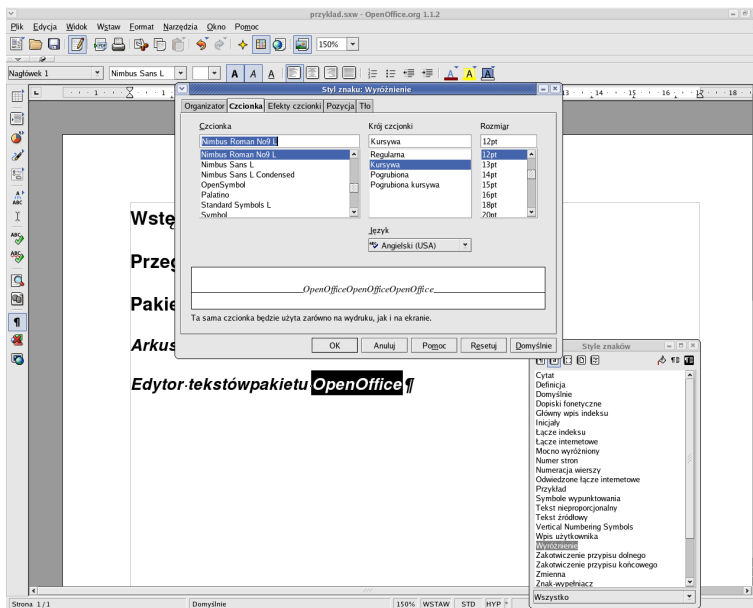
Style znaków można modyfikować. Dzięki temu możemy decydować, jak będą wyglądać fragmenty dokumentów złożone poszczególnymi stylami, np. stylem *Wyróżnienie*. W tym celu:

- Otwórz Stylistę
- Wybierz style znaków
- Wybierz styl, który chcesz modyfikować i kliknij na niego prawym klawiszem myszki
- Wybierz opcję *Modyfikuj*
- Ustaw atrybuty w poszczególnych zakładkach
- Zakończ edycję klikając myszką klawisz *OK*

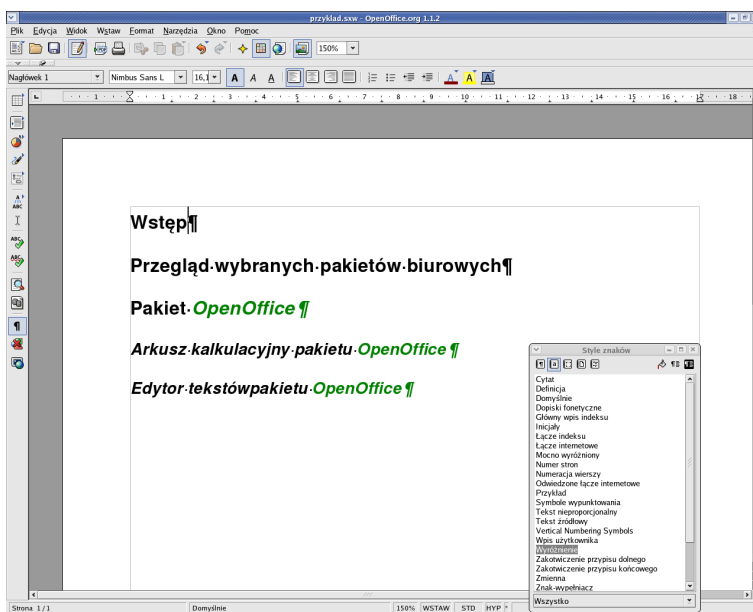
Aby zmodyfikować istniejący styl klikamy na niego w stylistyce prawym klawiszem myszy i wybieramy opcję *Modyfikuj*. Zmodyfikujemy dla przykładu styl *Wyróżnienie*. W tym celu wyświetlamy stylistę, wybieramy style znaków i klikamy prawym klawiszem myszy na styl *Wyróżnienie*. Z menu podręcznego wybieramy opcję *Modyfikuj*. Pojawia się okienko z pięcioma zakładkami, z których aktywna jest na początku zakładka *Organizator*.



Zakładka *Organizator* służy do organizacji stylów. Nie przejmujemy się nią na razie — do chwili tworzenia nowego stylu i przejdźmy do zakładki *Czcionka*:



Umożliwia ona ustawienie parametrów związanych z wyglądem czcionki. Zmieńmy dla przykładu krój czcionki na pogrubioną kursywę pozostawiając rozmiar bez zmian. Następnie za pomocą kolejnej zakładki: *Efekty czcionki* ustawmy kolor czcionki na zielony. Po kliknięciu na klawisz *OK*. Powinniśmy zobaczyć, że wszystkie fragmenty złożone uprzednio stylem znaków *Wyróżnienie* zmieniły swój wygląd:



Zauważmy, że nie musieliśmy przy tym pracować wyszukiwać ich w tekście! Jeśli w przyszłości zapagniemy ponownie zmienić sposób formatowania wyróżnionych fragmentów tekstu, to wystarczy ponowna zmiana atrybutów stylu *Wyróżnienie*.

10.4 Definiowanie nowego stylu znaków

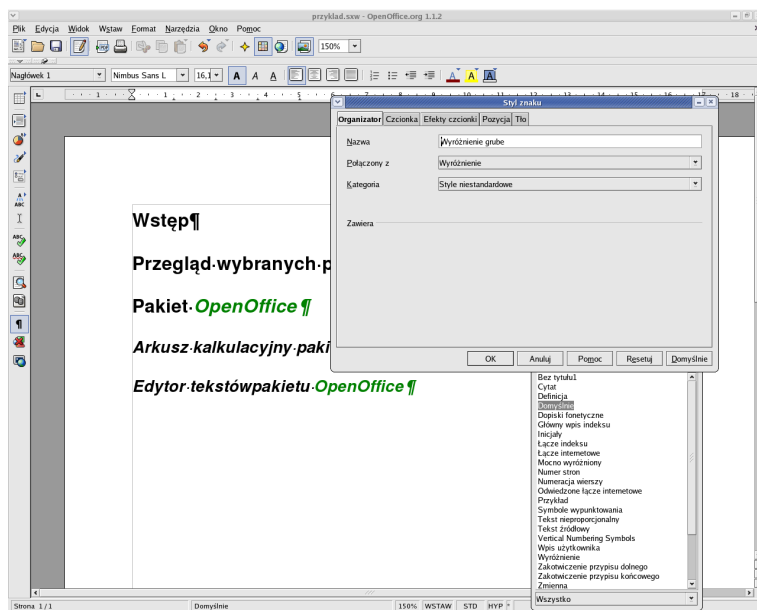
Definiowanie nowego stylu przebiega w sposób bardzo podobny do modyfikacji stylu już istniejącego. Do tworzenia nowego stylu należy się jednak przygotować. Po pierwsze każdy styl musi mieć nazwę. Dobrze jest jeśli nazwa ta mówi coś o przeznaczeniu stylu — wtedy nie trzeba się zastanawiać jakiego stylu użyć w konkretnym kontekście. Po drugie, nowy styl może bazować na stylu już istniejącym (inaczej: być z nim *połączony*). Wtedy w chwili rozpoczęcia definiowania nowego stylu ma on takie same atrybuty, jak styl, z którym jest połączony. Co więcej jakakolwiek zmiana tych atrybutów stylu bazowego, które nie są zmodyfikowane w definiowanym stylu, spowoduje także zmiany w wyglądzie fragmentów składanych definiowanym stylem.

Definiowanie nowego stylu znaku przebiega następująco:

- Otwórz Stylistę
- Wybierz style znaków
- Wybierz styl bazowy dla nowego stylu
- Kliknij na niego prawym klawiszem myszki
- Wybierz opcję *Nowy*
- W zakładce *Organizator* wprowadź nazwę nowego stylu
- Ustaw atrybuty w poszczególnych zakładkach
- Zakończ edycję klikając myszką klawisz *OK*

Popatrzmy jak wygląda to w praktyce. Przypuśćmy, że chcemy zdefiniować nowy styl, który również będzie służył do wyróżniania elementów tekstu, ale tym razem jedynie czcionką wytłuszczoną i powiększoną w stosunku do zwykłej czcionki w tekście. Pozostałe atrybuty stylu powinny być takie jak atrybuty stylu *Wyróżnienie*. W tym celu otwieramy stylistę, wybieramy style znaków i wybieramy styl bazowy dla nowo tworzonego stylu. W tym przypadku będzie to oczywiście styl *Wyróżnienie*. Klikamy na styl *Wyróżnienie* prawym klawiszem myszy i z menu podręcznego wybieramy opcję *Nowy*. Na ekranie pojawi się ten sam dialog, co przy modyfikacji stylu. Tym razem skupmy się jednak najpierw na zakładce *Organizator*.

Najpierw w polu *Nazwa* wprowadzamy nazwę nowego stylu. Wpiszmy na przykład *Wyróżnienie grube*. Sprawdźmy w polu *Połączony z*, czy styl jest połączony ze stylem *Wyróżnienie*. Jeśli tak nie jest, to znaczy, że kliknęliśmy w stylisicie niewłaściwy styl. Nie musimy jednak powtarzać wszystkich kroków wystarczy, że zmienimy wartość w polu *Połączony z*. Wartość *Kategoria* pozostawmy bez zmian — nie będziemy się zajmować tym polem. Zanim przejdziemy do kolejnej zakładki zwróćmy jeszcze uwagę, że część okienka pod napisem *Zawiera* jest pusta:



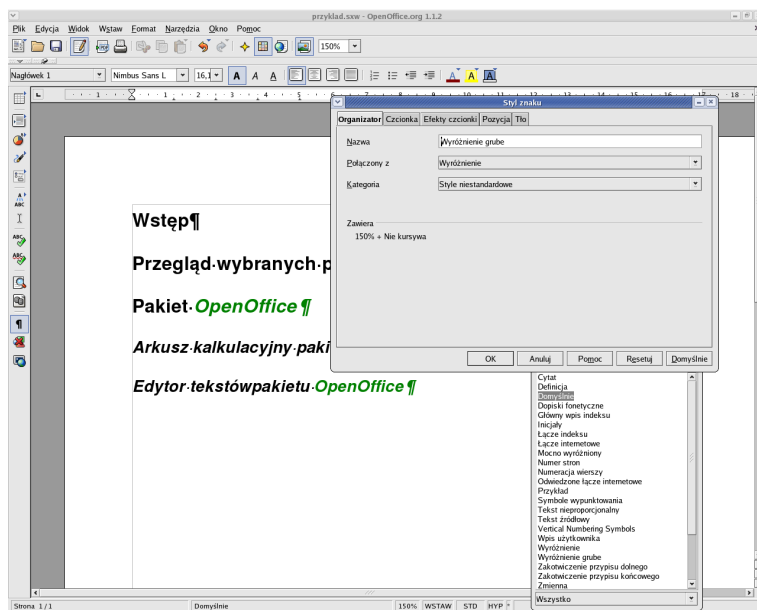
Teraz wybierzmy zakładkę *Czcionka* i zmienmy rozmiar czcionki. Warto wiedzieć, że pole *Rozmiar* można wypełnić na dwa sposoby. Pierwszy to podanie wielkości np.: *16pt*. Wtedy wszystkie fragmenty tekstu składane takim stylem będą miały rozmiar czcionki 16 punktów. Często lepszy efekt można jednak uzyskać uzależniając rozmiar czcionki od rozmiaru czcionki bazowej. W tym celu podajemy rozmiar czcionki jako wartość w procentach, np.: *150%* — rozmiar czcionki zostanie wówczas ustawiony jako 150% rozmiaru czcionki ze stylu bazowego. Jeśli więc nasz styl ma jako styl bazy styl *Wyróżnienie*, a w stylu *Wyróżnienie* wielkość czcionki zdefiniowano na 12pt, to rozmiar czcionki wyniesie 18pt. Co więcej, jeśli później w stylu bazowym zmienimy rozmiar czcionki na 10pt, to automatycznie wszystkie fragmenty składane stylem *Wyróżnienie* zmienią rozmiar na 15pt!

Taki sposób postępowania jest szczególnie uzasadniony przy definiowaniu stylu przeznaczonych do składania na przykład różnego rodzaju nagłówków. Zamiast ustawiać w nich na stałe wielkość rozmiaru czcionki możemy uzależnić ją od stylu, którego używamy do składania zwykłych akapitów.

Wróćmy jednak do przykładu. Po ustawieniu rozmiaru czcionki na 150% przechodzimy do kolejnej zakładki *Efekty czcionki*. Tu możemy ustawić różne dodatkowe parametry, takie jak sposób podkreślenia, przekreślenia, kolor (jak robiliśmy to już poprzednio), efekty, cienie, kontury itp. Z kolei w zakładce *Pozycja* możemy zdecydować, czy napis składany tym stylem będzie obrócony, czy może podniesiony względem normalnie składanego wiersza (pole *Indeks górny*) czy też będzie złożony poniżej normalnego wiersza (pole *Indeks dolny*). Możemy przy tym sami dobrać wielkość przesunięcia oraz powiększenie/pomniejszenie czcionki. Jest także możliwość zwiększenia/zmniejszenia odstępu między poszczególnymi literami poprzez wybór odpowiedniej wartości w polu *Odstęp*. Ostatnia zakładka *Tło* służy do ustalenia koloru tła.

Po zakończeniu edycji powróćmy jeszcze raz do zakładki *Organizator*. Tym

razem część okienka pod napisem *Zawiera* zawiera wszystkie zmiany, jakie dokonaliśmy w stosunku do stylu bazowego:



Gdy jesteśmy zadowoleni z dokonanych ustawień klikamy klawisz *OK*. Nowy styl powinien pojawić się w stylisście. Na razie jednak nie został on zastosowany do żadnego fragmentu tekstu. Aby to zrobić postępujemy w sposób opisany wcześniej: zaznaczamy fragment tekstu myszką i podwójnie klikamy na nazwę nowego stylu w stylisście.

Ćwiczenie 10.4.1. Zastosuj nowy styl do wybranych fragmentów dokumentu.

Warto wiedzieć, że definicję nowego stylu można kończyć na różne sposoby. Klawisz *OK* oczywiście zatwierdza wszystkie wprowadzone zmiany, a klawisz *Anuluj* je anuluje. W przypadku niektórych stylów możemy ponadto przywrócić ich standardową definicję klikając klawisz *Domyślne*. Przydaje się to, jeśli zmodyfikowany został któryś ze standardowych stylów i chcemy przywrócić jego oryginalną postać.

Ćwiczenie 10.4.2. Co zmieni się w dokumencie, jeśli zmodyfikujemy teraz styl *Wyróżnienie* zwiększając rozmiar czcionki. Czy zmianie ulegną jedynie fragmenty złożone tym stylem?

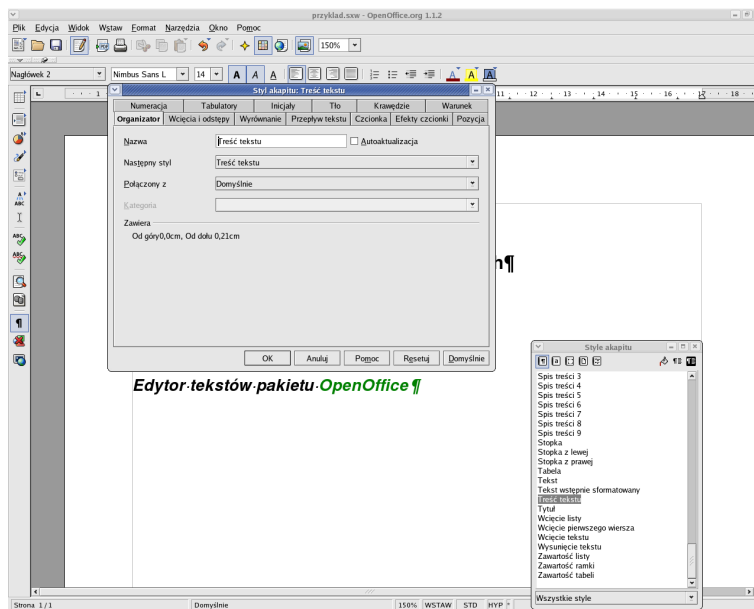
10.5 Style akapitów: tworzenie i modyfikacja

Do edycji i tworzenia stylów akapitów stosują się te same zasady co w przypadku omówionych w poprzednim punkcie dotyczącym stylów znaków.

- Otwórz Stylistę
- Wybierz style akapitu

- Wybierz styl bazowy i kliknij na niego prawym klawiszem myszki
- Wybierz opcję *Nowy* lub *Modyfikuj*
- W zakładce *Organizator* wprowadź nazwę stylu w przypadku definiowania nowego stylu
- Ustaw nazwę następnego stylu i zdecyduj, czy styl ma być automatycznie aktualizowany
- Ustaw atrybuty w pozostałych zakładkach
- Zakończ edycję klikając myszką klawisz *OK*

Tym razem jednak okienko określające parametry stylu jest dużo bardziej rozbudowane. Już sama liczba zakładek jest większa: 12 zamiast 5, ale niektóre zostały już omówione w poprzednim rozdziale i punkcie o stylach znaków.



Zakładka *Organizator* pełni tę samą funkcję, co w przypadku stylu znaku. Mamy tutaj jednak dodatkowe pole *Następny styl*.

Znaczenie tego pola najlepiej omówić na przykładzie. Przypuśćmy, że zdefiniowaliśmy styl o nazwie *Zwykły* służący do składania zwykłego akapitu w tekście oraz styl o nazwie *Przykład*, którym będziemy formatować przykłady. Gdy piszemy tekst i chcemy złożyć przykład zmieniamy styl nowego akapitu na styl *Przykład*. Następnie wprowadzamy tekst przykładu. Po jego wprowadzeniu rozpoczynamy nowy akapit wciskając klawisz *Enter*. Ten nowy akapit najprawdopodobniej będzie jednak składany stylem zwykłym, więc wygodnie byłoby, gdyby po zakończeniu akapitu z przykładem edytor sam zmienił styl na *Zwykły*. Do tego celu służy właśnie pole *Następny styl* — jest to nazwa stylu akapitu, która ma być automatycznie użyta do składania akapitu następującego po zdefiniowanym stylu. Definiując styl

Przykład podamy w polu *Następny styl* wartość *Zwykły*. W polu *Następny styl* stylu *Zwykły* wpisujemy także wartość *Zwykły*, bo po zwykłym akapicie najczęściej następuje również zwykły akapit.

W zakładce *Organizator* występuje też pole *Autoaktualizacja*. Jeśli pole to jest zaznaczone, to każda zmiana atrybutów wykonana w dowolnym akapicie złożonym definiowanym stylem powoduje automatyczne dokonanie analogicznych zmian we wszystkich akapitach złożonych tym stylem. Zaletą takiego rozwiązania jest zachowanie spójnego wyglądu wszystkich akapitów korzystających z ustalonego stylu. Jeśli jednak chcemy zmienić jakiś drobiazg w wyglądzie konkretnego akapitu, musimy pamiętać o wyłączeniu tej opcji!

Pozostałe zakładki służą do ustawiania poszczególnych atrybutów akapitu takich jak rozmiar czcionki, wcięcia i odstępy, wyrównanie tekstu itd. Zatrzymajmy się jedynie dłużej przy zakładce *Numeracja*. Pełni ona ważną rolę przy definiowaniu akapitów, które powinny być automatycznie numerowane — z takiej możliwości często korzystamy definiując na przykład styl do formatowania przykładów, zadań, uwag, twierdzeń itp.

10.6 Style z automatyczną numeracją

Automatyczna numeracja pewnych akapitów umożliwia rzucenie na edytor zadania polegającego na utrzymaniu ciągłości i spójności numeracji przykładów, definicji itp. Jest to szczególnie istotne wówczas, gdy tworzony dokument podlega zmianom — dodajemy nowe przykłady, usuwamy istniejące albo zmieniamy ich kolejność. Dzięki włączonej autonumeracji nie musimy sami zamieniać numerów przy każdej zmianie, która tego wymaga — robi to za nas edytor.

Motywacja:

- automatyczne zachowanie właściwej kolejności
- ciągłość i spójność numeracji
- brak konieczności żmudnych modyfikacji wielu fragmentów tekstu

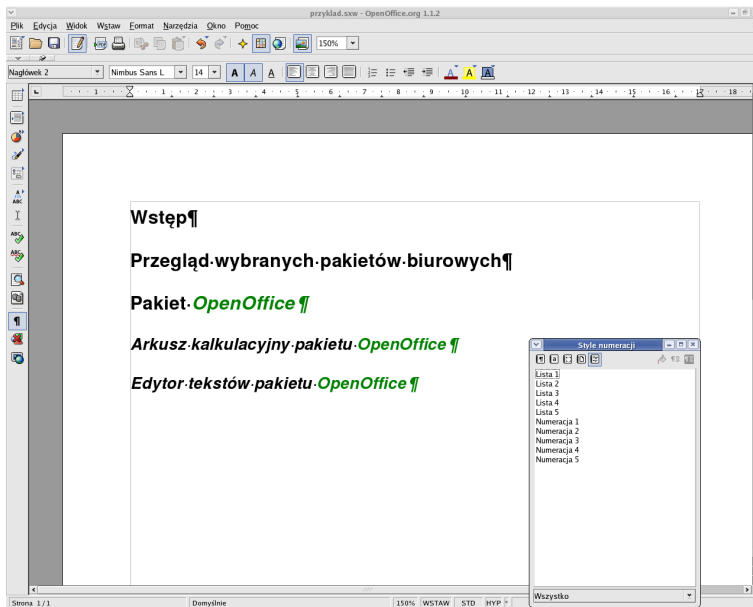
Sposób postępowania:

- Zdefiniuj *styl numeracji*. Może on na przykład numerować przykłady liczbami arabskimi umieszczając dodatkowo przed numerem słowo **Przykład**.
- Zdefiniuj *styl akapitu* podając jako styl numeracji styl zdefiniowany w poprzednim punkcie

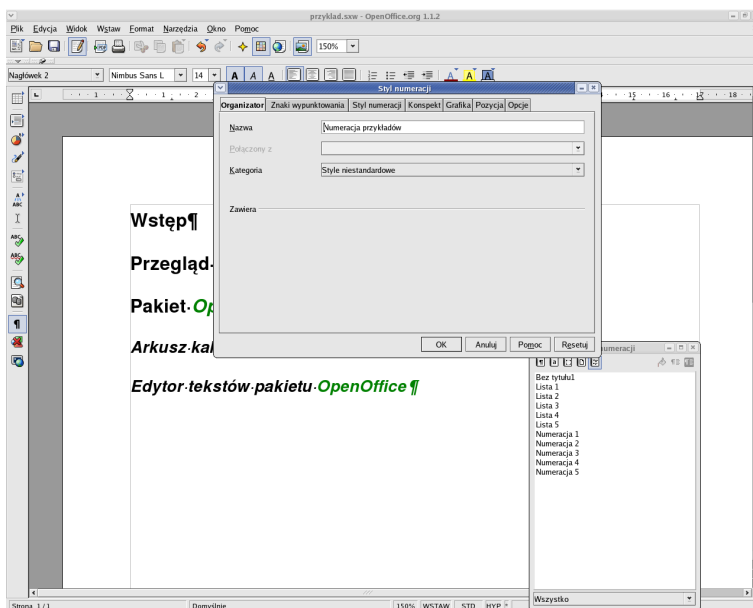
Sposób postępowania ze stylami z automatyczną numeracją zilustrujemy na przykładzie stylu do formatowania przykładów. Dobrze byłoby, aby taki

styl uwzględniał automatyczną numerację przykładów tak, aby każdy przykład otrzymywał automatycznie numer i aby numeracja ta była spójna w całym dokumencie. W tym celu musimy zdefiniować *dwa* style: jeden to styl numeracji, drugi — styl akapitu.

Zacznijmy od stylu numeracji. Otwieramy stylistę i w jego pasku narzędzi wybieramy *styl numeracji*:



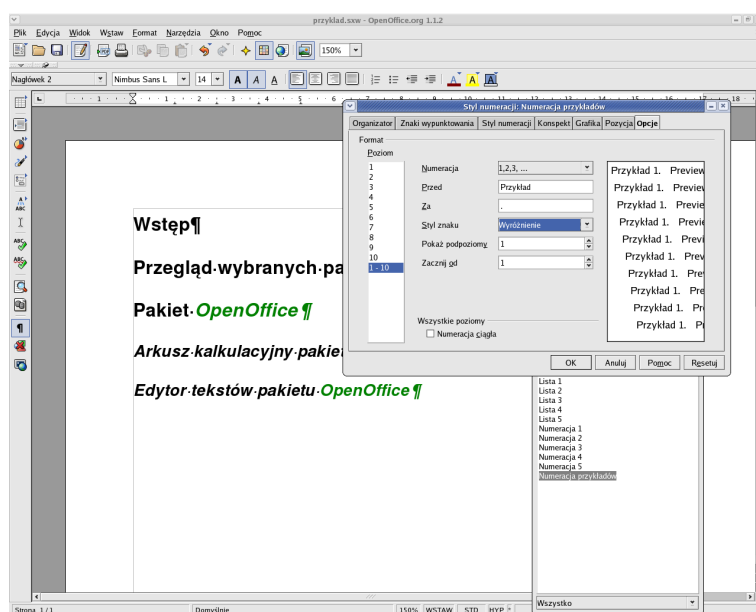
Następnie tworzymy nowy styl numeracji. Obowiązują przy tym te same zasady, co przy tworzeniu stylu znaków. Najpierw klikamy prawym klawiszem myszy na istniejący styl, na przykład *Numeracja 1* i wybieramy opcję *Nowy*. To właśnie na tym stylu będzie bazował tworzony przez nas nowy styl. Ustawiamy w zakładce *Organizator* nazwę stylu na *Numeracja przykładów*:



Następnie przechodzimy do zakładki *Opcje*. Zakładka ta umożliwia ustawienie sposobu numeracji poprzez ustawienie m.in. następujących pól:

1. Pole *Numeracja* — sposób numerowania. Możemy wybrać tu na przykład liczby arabskie. Zauważmy, że w okienku po prawej stronie na bieżąco oglądamy efekt dokonywanych zmian.
2. Pole *Przed* — tekst przed numerem. Wpisujemy tu tekst, który ma pojawiać się przed numerem przykładu. Wpiszmy tam słowo „Przykład”, a po nim spację.
3. Pole *Za* — tekst po numerze. Wpisujemy tu tekst, który będzie następował po numerze przykładu. Wpiszmy kropkę.
4. Pole *Styl znaku* — styl jakim składany jest numer. Umożliwia zdefiniowanie sposobu składania numeru. Możemy zastosować do niego dowolny zdefiniowany styl znaków, np.: *Wyróżnienie*.

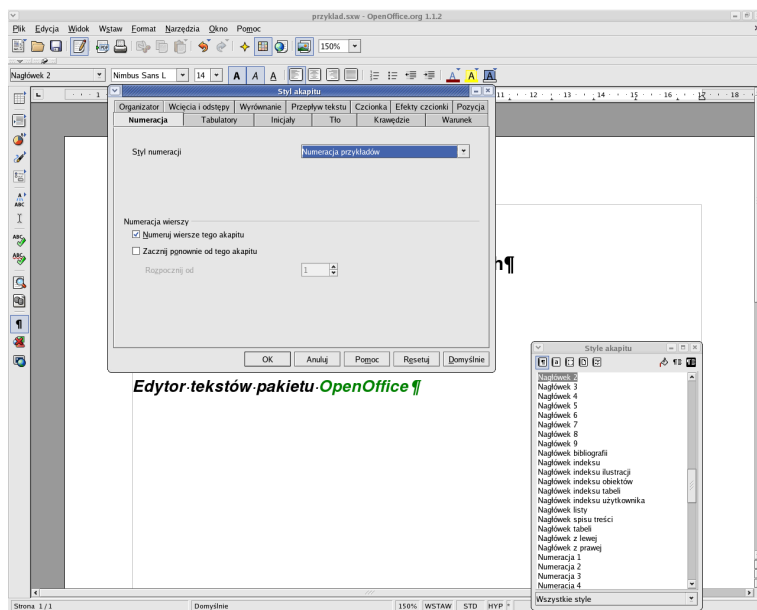
Powinniśmy otrzymać następujący dialog:



Kliknięcie klawisza *OK* kończy tworzenie stylu numeracji.

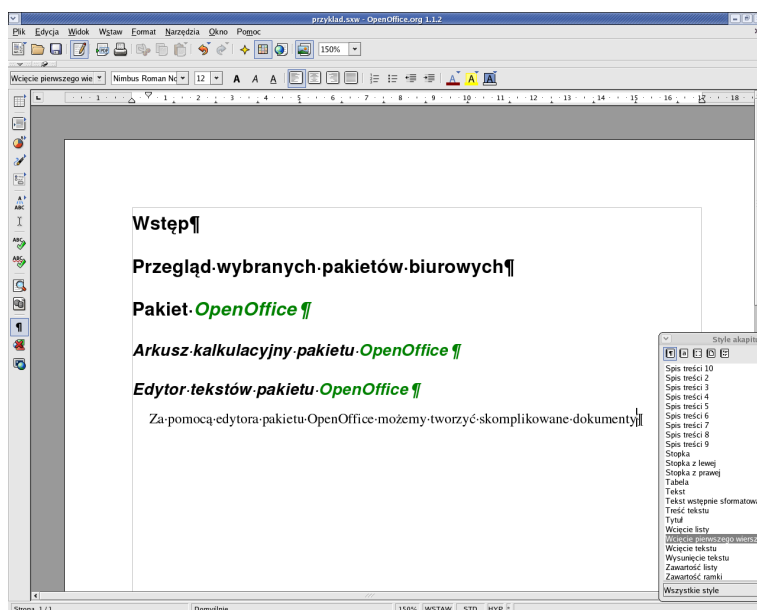
W tym momencie mamy styl, którego użyjemy do *numerowania* przykładów. Potrzebujemy jeszcze styl do *składania* przykładów.

Wyświetlamy zatem stylistę, wybieramy style akapitów, a następnie tworzymy nowy styl, który nazwiemy *Przykład*. Ustawiamy potrzebne parametry tego stylu (możemy na przykład zwiększyć wcięcie tak, aby przykłady rzucały się w oczy przy czytaniu tekstu), po czym przechodzimy do zakładki *Numeracja*. Tam jako *Styl numeracji* wybieramy zdefiniowany uprzednio styl *Numeracja przykładów* i zaznaczamy pole *Numeracja wierszy*:

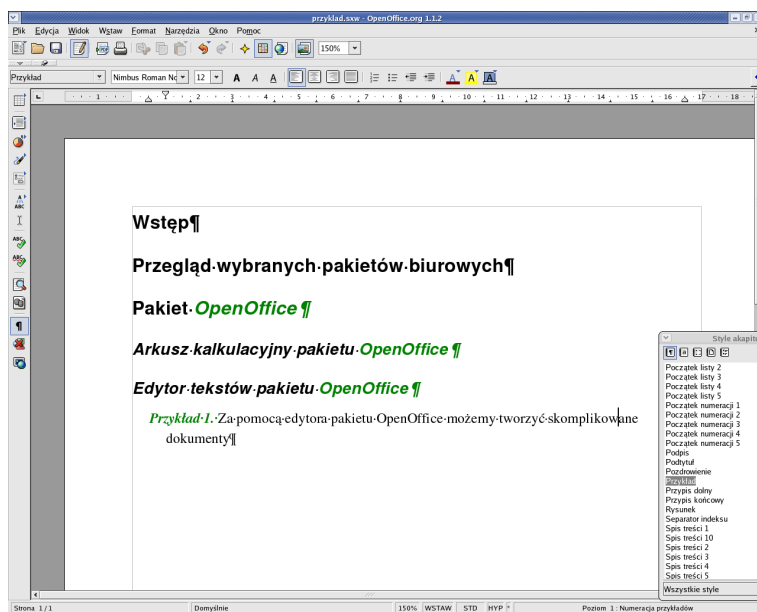


Po zakończeniu edycji klawiszem *OK* możemy przygotować akapit, który posłuży jako przykład.

Wprowadzając tekst pomijamy słowo „Przykład” oraz numer otrzymując dokument wyglądający mniej więcej w następujący sposób:



Następnie stosujemy zdefiniowany styl do akapitu:



10.7 Szybka zmiana stylu akapitu

Edytor tekstu zapamiętuje style, które zostały zastosowane w edytowanym dokumencie. W lewym górnym rogu paska obiektów znajduje się pole zawierające nazwę stylu, którym złożono bieżący akapit. Jeśli chcemy zmienić ten styl na inny (ale już zastosowany w innym fragmencie tekstu), nie musimy odwoływać się do stylisty. Po prostu wybieramy odpowiedni styl posługując się listą wyboru.

10.8 Szybkie definiowanie nowego stylu akapitu

Zamiast pracować ustawiać wszystkie atrybuty stylu za pomocą dialogu definiującego nowy styl możemy postąpić w inny sposób. Najpierw formatujemy akapit w odpowiadający nam sposób. Następnie otwieramy stylistę i klikamy na przycisk *Kopiuj z akapitu*. Pojawi się dialog, za pomocą którego wprowadzimy nazwę nowego stylu, a jego atrybuty zostaną automatycznie ustawione tak, jak w przygotowanym akapicie.

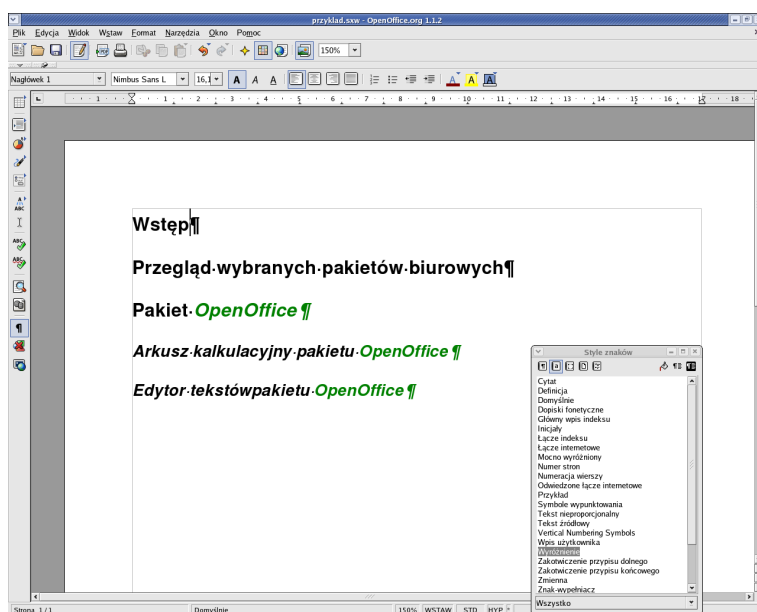
10.9 Konspekt i automatyczna numeracja rozdziałów

Jak już wspomniano, ważną rzeczą jest automatyczna numeracja rozdziałów w dokumencie. Zadanie to realizuje się za pomocą tzw. konspektu. Konspekt definiuje strukturę pracy: jej rozdziały, podrozdziały itd.

Edytor tekstów ma zdefiniowane specjalne style przeznaczone do składania tytułów rozdziałów. Style te noszą nazwy *Nagłówek 1*, *Nagłówek 2*, ...,

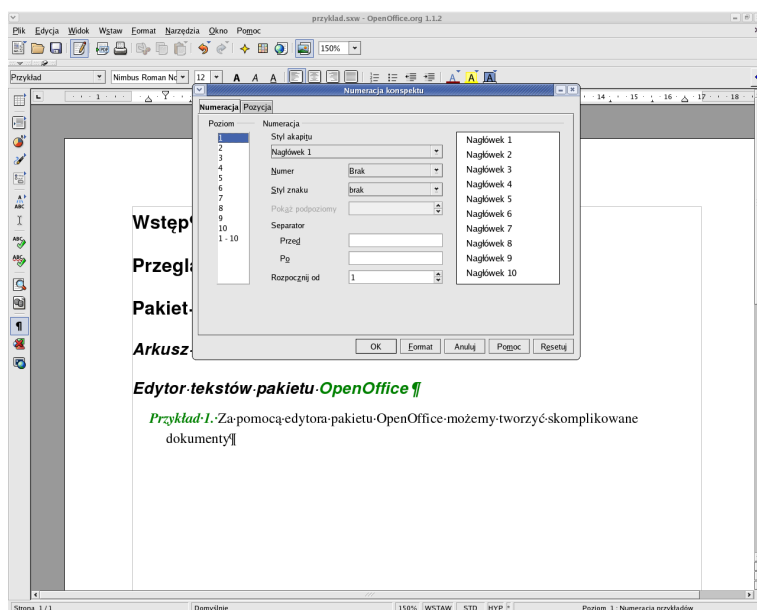
Nagłówek 10. Każdy z nich służy do składania tytułów znajdujących się na odpowiednim poziomie. I tak, jeśli dokument składa się z 4 części to do złożenia tytułu każdej z nich użyjemy *Nagłówka 1*. Jeśli część pierwsza ma 6 rozdziałów, to każdy z nich należy złożyć posługując się *Nagłówkiem 2*. Podrozdziały tych rozdziałów będą miały tytuły złożone *Nagłówkiem 3* itd. Oczywiście każdy z tych stylów można zmodyfikować, jeśli nie odpowiada nam domyślny wygląd. Do formatowania nagłówków poszczególnych składowych struktury można też użyć stylów o innych nazwach.

Sformatowanie tytułu części, rozdziału, podrozdziału nie powoduje jeszcze włączenia automatycznej numeracji, a jedynie nadaje strukturę dokumentowi. Aby zobaczyć jak włączyć automatyczną numerację przygotujemy najpierw dokument zawierający kilka rozdziałów, podrozdziałów i punktów. Możemy też wykorzystać w tym celu wcześniej przygotowany dokument o pakietach biurowych:



Aby włączyć numerację należy posłużyć się dialogiem dostępnym poprzez *Narzędzia* ► *Numeracja konspektu*.

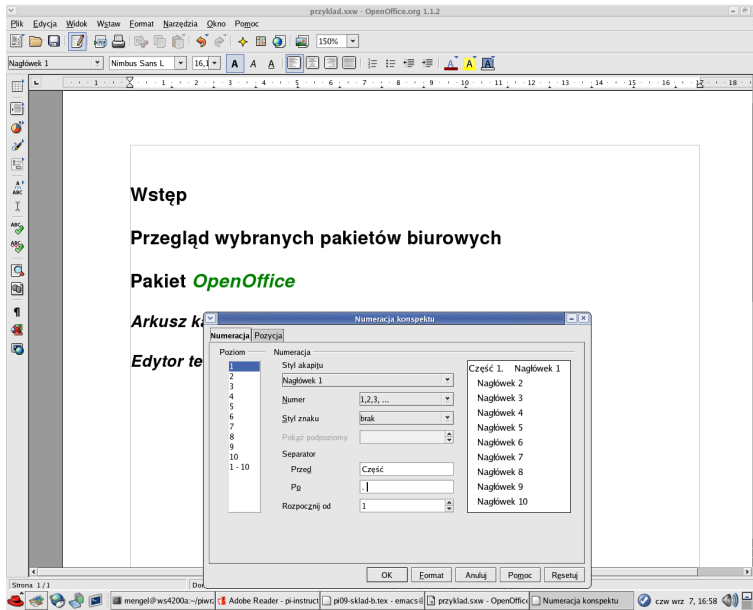
Na ekranie pojawi się wówczas okno przypominające to, którym posługiwaliśmy się definiując style numeracji. Omówimy teraz dokładnie poszczególne jego elementy.



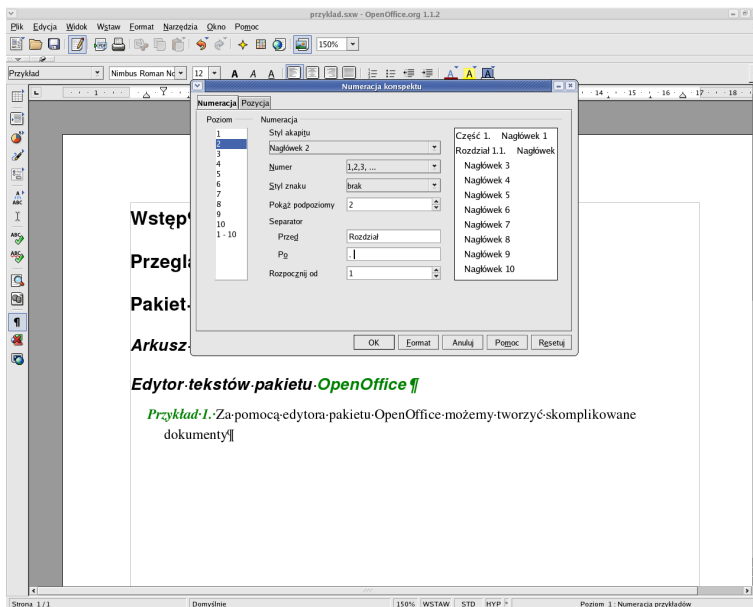
Po lewej stronie znajdują się numery poziomów, których parametry chcemy ustawić. Jeśli chcemy zdefiniować sposób numeracji części zaznaczamy poziom 1, jeśli rozdziałów w poszczególnych częściach — poziom 2 itd. Po upewnieniu się, że jest podświetlony odpowiedni numer poziomu możemy ustawić wartości w poszczególnych polach. I tak, pole *Numer* określa sposób numeracji nagłówków na danym poziomie. Numerami mogą być m.in. liczby arabskie lub rzymskie, a także małe lub wielkie litery alfabetu. Pole *Styl znaku* określa w jaki sposób (czyli jakim stylem znaków) zostanie złożony numer. Pole *Pokaż podpoziomy* określa ile podpoziomów ma zostać uwzględnionych przy numerowaniu nagłówków danego poziomu. Na przykład, jeśli definiujemy sposób numeracji poziomu 3 i określimy, że ma być pokazywany jeden podpoziom, to tytuły podrozdziałów składających się na rozdział 2 części 3 będą numerowane po prostu kolejnymi liczbami. Jeśli jednak zażądamy pokazania 3 podpoziomów, to podrozdział 4 w rozdziale 2 i części 3 będzie miał numer 3.2.4.

Pola *Przed* i *Po* określają tekst, który ma wystąpić odpowiednio przed numerem rozdziału i po nim, a *Rozpocznij od* umożliwia ustawienie pierwszego numeru na danym poziomie (dzięki temu możemy na przykład rozpocząć numerację rozdziałów od 0 a nie od 1).

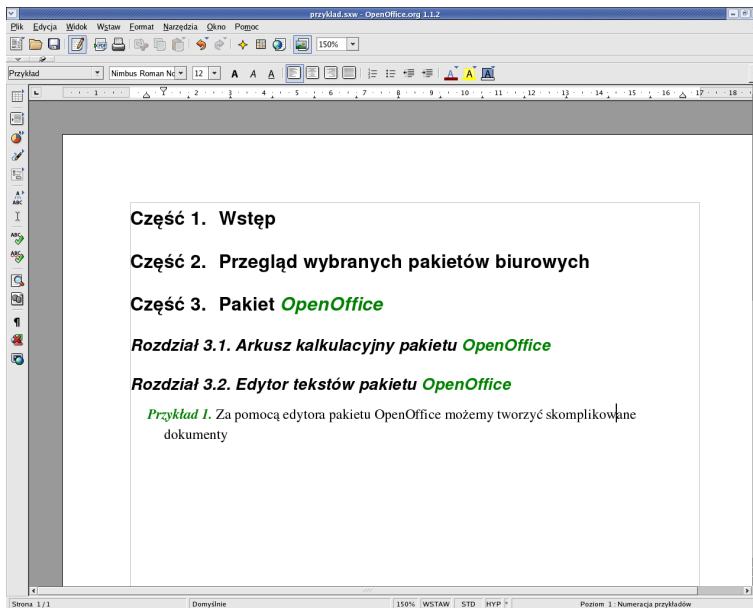
Spróbujmy teraz wydefiniować wszystkie parametry tak, aby uzyskać automatyczną numerację części. Sposób formatowania numeracji pierwszego poziomu możemy ustawić na przykład tak:



a numeracji nagłówków drugiego poziomu tak:



Po dokonaniu takich ustawień dokument będzie wyglądał następująco:



W drugiej zakładce dialogu *Numeracja konspektu*, zatytułowanej *Pozycja* można ustawiać sposób rozmieszczenia tekstu względem numery. Zasada działania tego zakładki jest taka jak poprzedniej, tzn. w okienku po lewej stronie należy zaznaczyć poziom, którego dotyczą ustawiane parametry, a następnie dobrać stosowanie do potrzeb wartości poszczególnych parametrów. Na bieżąco w dolnej części okienka wyświetlany jest podgląd układu tekstu.

Ćwiczenie 10.9.1. Dodaj na początku dokumentu nagłówek nowej części. Co stało się z numerami pozostałych części? Czy widzisz już dlaczego przedstawiony tu sposób jest lepszy niż ręczna numeracja?

Definicja konspektu w opisany powyżej sposób umożliwia nam ponadto łatwe i automatyczne utworzenie spisu treści oraz tworzenie odnośników.

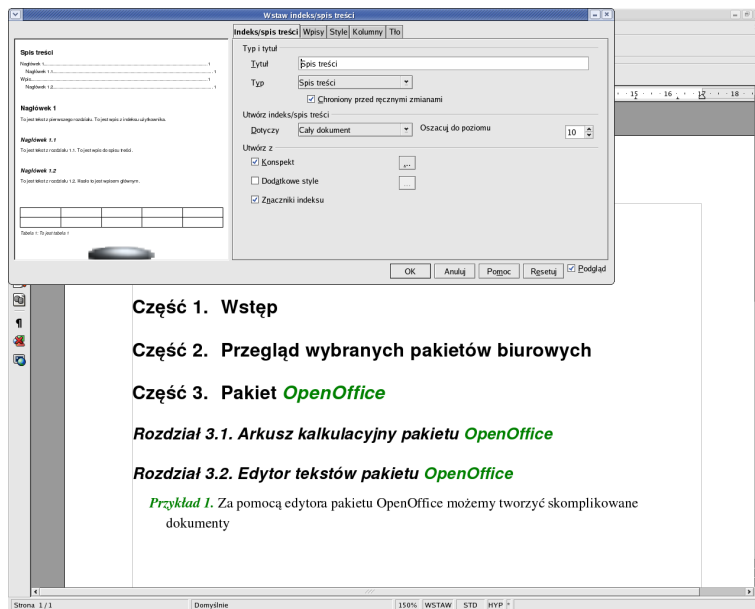
10.10 Spis treści i inne spisy

Aby wstawić w dowolne miejsce dokumentu spis treści należy:

1. Upewnić się, że zdefiniowaliśmy konspekt w opisany w poprzednim punkcie sposób.
2. Ustawić kursor w tym miejscu tekstu, w którym ma być wstawiony spis treści.
3. Wybrać z menu *Wstaw* opcję *Indeksy i spisy*.
4. Ustawić tytuł (domyślnie jest to po prostu *Spis treści*).
5. Ustawić *typ spisu* na *Spis treści* (mamy też inne możliwości, na przykład *Indeks alfabetyczny*).

6. Upewnić się, że zaznaczone jest pole wyboru *Konspekt*.
7. Kliknąć klawisz *OK*.

Pojawi się wówczas dialog:



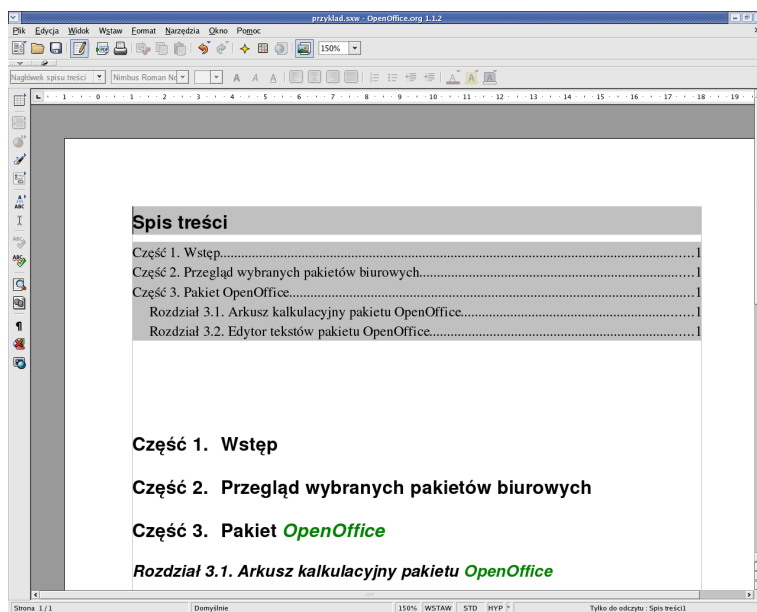
Warto wiedzieć, że pozostawienie zaznaczonego pola **Chroniony przed ręcznymi zmianami** uniemożliwi ręczną edycję spisu treści. Oczywiście w miarę dodawania nowych rozdziałów do tworzonego tekstu, spis treści będzie aktualizowany bez konieczności ręcznej edycji znajdujących się w nim pozycji.

Czasami zdarza się, że chcemy zamieścić w tekście spisy treści ograniczone do konkretnych rozdziałów. Możemy na przykład na początku każdej części książki umieszczać spisy treści zawierające jedynie rozdziały, które składają się na daną część. Zadanie to zrealizujemy ustawiając odpowiednie wartości pola *Dotyczy*.

Pozostałe zakładki okienka *Wstaw indeks/spis treści* dotyczą sposobu edycji poszczególnych pozycji w spisie treści oraz sposobu złożenia całego spisu. Nie będziemy się nimi szczegółowo zajmować.

□ **Ćwiczenie 10.10.1.** Stwórz spis treści w opisany powyżej sposób. Spróbuj w nim dokonać zmian. Co się dzieje? Jak zmienić to zachowanie? □

Po wstawieniu do dokumentu spis treści jest podświetlany na szaro. Przypomnijmy, że w ten sposób edytor oznacza elementy wyliczane automatycznie:



Wstawiony do dokumentu spis treści możemy usunąć klikając gdzieś w jego obrębie prawym klawiszem myszki i wybierając opcję *Usuń indeks*. Inne opcje to *Edytuj indeks*, która pozwala na zmianę ustawień spisu treści poprzez ponowne wywołanie opisanego dialogu oraz *Aktualizuj indeks*. Ta ostatnia opcja służy do wymuszenia aktualizacji spisu treści, gdy zmieni się struktura dokumentu (dojdą nowe rozdziały lub części, zmieni się ich kolejność lub też niektóre z nich zostaną usunięte). Zdarza się, że edytor tekstu sam nie uaktualni w takich sytuacjach spisu treści i wtedy właśnie jest konieczne jego zaktualizowanie w opisany sposób.

10.11 Wyszukiwanie różnych atrybutów tekstu

Oprócz standardowego wyszukiwania wzorca jest możliwe także bardziej zaawansowane przeszukiwanie dokumentu.

- Jest możliwe wyszukiwanie konkretnych atrybutów tekstu w dokumencie, na przykład:
 - czcionki pogrubionej
 - czcionki pochyłej
 - zmiany interlinii
 - miejsc, w których czcionka różni się od domyślnej
 - ...
- Umiejętne stosowanie tego rodzaju przeszukiwania pozwala na ujednolicenie szaty graficznej dokumentu
- Pełna lista atrybutów wraz z wyjaśnieniem ich znaczenia znajduje się w systemie pomocy

Dialog *Znajdź i zastąp* daje możliwość wyszukania określonych atrybutów tekstu. Aby wykonać tego rodzaju przeszukiwanie dokumentu należy kliknąć przycisk *Atrybuty*. Otwiera się wówczas okienko pozwalające zaznaczyć te atrybuty, które chcemy znaleźć w tekście.

I tak na przykład zaznaczenie pola *Czcionka* powoduje znalezienie tych miejsc w tekście, w którym została zmieniona czcionka. Dokładniej mówiąc chodzi o te fragmenty tekstu, które zostały złożone czcionką inną niż domyślna czcionka związana ze stylem, w którym złożono dany akapit. Podobnie zaznaczenie pola *Grubość czcionki* spowoduje odnalezienie tych miejsc, w których tekst został pogrubiony.

Tego rodzaju przeszukiwanie może być przydatne, gdy zależy nam na ujednoliceniu postaci graficznej dokumentu. Kluczowe jest wtedy znalezienie wszystkich miejsc, w których czcionka, jej rozmiar, kolor lub inne atrybuty różnią się od wartości domyślnych ustalanych za pomocą stylu, którym złożono akapit.

Atrybutów jest bardzo dużo, a ich znaczenie jest opisane w systemie pomocy do edytora. Pomoc można uzyskać zawsze wciskając klawisz *Pomoc* w aktualnie wyświetlanym okienku lub dialogu.

Po wyborze interesujących nas atrybutów i wciśnięciu klawisza *OK*, powracamy do głównego okienka *Znajdź i zastąp*. Dalszy sposób postępowania jest taki sam, jak przy zwykłym wyszukiwaniu.

Uwaga! Jak już o tym wspominaliśmy dialog *Znajdź i zastąp* zapamiętuje ustawienia. Zaznaczone atrybuty pozostają więc zaznaczone dopóki ich nie odznaczymy. W szczególności, jeśli ponownie wywołamy dialog *Znajdź i zastąp* próbując znaleźć wszystkie wystąpienia danego słowa, a przy poprzednim jego użyciu zaznaczyliśmy pewne atrybuty, to przeszukiwanie będzie dotyczyć tylko tych wystąpień szukanego słowa, które mają wskazane atrybuty.

10.12 Wyrażenia regularne

Jeszcze więcej możliwości stwarza wyszukiwanie wyrażeń regularnych.

- Zwykle wyszukiwanie wymaga podania dokładnie tekstu, który chcemy znaleźć
- Wyrażenie regularne jest *wzorem*, który może pasować do wielu słów, np.:
 - możemy szukać słów złożonych z czterech liter rozpoczynających się literą *A*
 - możemy szukać słów, których pierwsze pięć liter to „matur”, a ostatnia szósta to *a*, *e*, *q* lub *y*
 - możemy szukać wszystkich liczb

- możemy ograniczyć wyszukiwanie do tych słów, które występują na początku akapitu
- Wyszukiwanie wyrażeń regularnych jest elastyczniejszą formą przeszukiwania, ale wymaga opanowania zasad budowania wyrażeń regularnych
- Aby szukać wyrażeń regularnych zaznaczamy pole **Wyrażenia regularne**

Wyszukiwanie wyrażeń regularnych jest najbardziej złożoną metodą przeszukiwania, oferującą jednak wiele użytecznych możliwości. Jej działanie wyjaśnimy na przykładzie. Przypuśćmy, że chcemy znaleźć wszystkie wystąpienia słowa „matura”. Jeśli wykonamy zwykłe wyszukiwanie znajdziemy te miejsca w dokumencie, w których występuje po prostu słowo „matura”, ale pominiemy miejsca, w których szukane słowo występuje w innych przypadkach niż mianownik. Możemy próbować szukać słowa (a dokładniej fragmenty słowa — przy odznaczonym polu **Tylko całe wyrazy**) „matur”. Wtedy jednak odnajdziemy także napisy takie jak „maturalny”, itp.

Wyrażenia regularne umożliwiają podanie nie tyle słowa, które chcemy znaleźć co *wzoru* opisującego te słowa. Ten *wzór* nosi właśnie nazwę wyrażenia regularnego. Co może zawierać taki wzór? W najprostszym przypadku zwykle znaki: litery, cyfry. Wtedy wyszukiwanie wyrażenia regularnego niczym nie różni się od zwykłego wyszukiwania. Możemy jednak w wyrażeniu regularnym zawrzeć pewne symbole, które mają specjalne znaczenie. I tak na przykład kropka w wyrażeniu regularnym nie oznacza po prostu kropki lecz dowolny pojedynczy znak. Zatem szukając wyrażenia „matur.” znajdziemy wszystkie wystąpienia słów „matura”, „matury”, „maturą” itd. Dzieje się tak dlatego, że ostatnim znakiem wyrażenia regularnego jest właśnie kropka, która oznacza dowolny znak.

Symbole występujące w wyrażeniach regularnych to m.in.:

- . (kropka) oznacza dowolny jeden znak
- \$ oznacza koniec akapitu
- ^ oznacza początek akapitu
- [agheo] oznacza dowolny (ale jeden!) znak spośród znaków w nawiasach
- [a-e] oznacza dowolny (pojedynczy) znak z zakresu od *a* do *e*
- * za wyrażeniem oznacza dowolną (także zerową) liczbę jego powtórzeń
- + za wyrażeniem oznacza dowolną (ale niezerową) liczbę jego powtórzeń
- \. oznacza kropkę
- *<* i *>* oznaczają odpowiednio początek i koniec słowa

Innym przydatnym symbolem mogą okazać się „^” oraz „\$”. Pierwszy z nich oznacza początek akapitu, drugi — koniec akapitu. Zatem wyrażenie „^matura” spowoduje wyszukanie wszystkich wystąpień słowa „matura” znajdujących się na początku akapitu. W szczególności „^\$” znajdzie wszystkie puste akapity.

Jeśli pewne pojedyncze znaki ujmiemy w nawiasy kwadratowe, to znajdziemy wszystkie wystąpienia dowolnego z nich. Na przykład $[aąęęioóu]$ odnajdzie wszystkie samogłoski w tekście. Oczywiście możemy łączyć wszystkie opisane sposoby, podając na przykład wzorzec $^{[nm]}atur[ayęo]$ spowoduje wyszukanie wszystkich wystąpień słów *matura*, *matury*, *maturę*, *maturzo*, ale także *natura*, *natury*, *naturę*, *naturzo* na początku akapitu.

Zamiast podawać w nawiasach kwadratowych poszczególne znaki możemy posłużyć się zakresem. Na przykład $[0-9]$ znajdzie wszystkie cyfry, a $[a-e]$ wszystkie wystąpienia liter od a do e.

Jeszcze więcej możliwości dają nam operacje $+$ oraz $*$. Gwiazdka oznacza powtórzenie poprzedzającego ją elementu dowolną (także zerową) liczbę razy, zaś $+$ oznacza powtórzenie tego elementu dowolną niezerową liczbę razy. I tak na przykład: Ab^*a odnajdzie wszystkie słowa: *Aa*, *Aba*, *Abba* itd, a wyrażenie $[0-9]^+$ odnajdzie wszystkie liczby występujące w tekście.

Aby skorzystać z wyszukiwania wyrażeń regularnych należy w dialogu *Znajdź i zastąp* zaznaczyć pole *Wyrażenia regularne* i wprowadzić w polu *Szukaj* wyrażenie, które poszukujemy w tekście. Dalsze postępowanie jest takie same jak w przypadku zwykłego przeszukiwania.

A oto kilka przykładów wyrażeń regularnych:

- Ala — słowo *Ala* na początku akapitu
- $A.^*a$ — dowolny ciąg symboli rozpoczynający się do *A* i kończący również literą *a*
- $[0-9]$ — jedna cyfra
- $[0-9]^+$ — liczba złożona z dowolnej liczby cyfr
- $\.$ — kropka

□ **Ćwiczenie 10.12.1.** Napisz wyrażenie regularne, które pasuje do słowa „matura” we wszystkich przypadkach (*matury*, *maturze*, ...). □

10.13 Indeks

W dużych dokumentach zachodzi często potrzeba przygotowania indeksu, zawierającego spis haseł wraz ze stronami, na których się one pojawiają. Edytor pakietu *OpenOffice* umożliwia automatyczne stworzenie takiego indeksu.

Przygotowanie indeksu przebiega dwuetapowo. Pierwszy etap to definiowanie haseł (wpisów) do indeksu, drugi etap to wstawienie w wybranym miejscu dokumentu samego indeksu i jego sformatowanie.

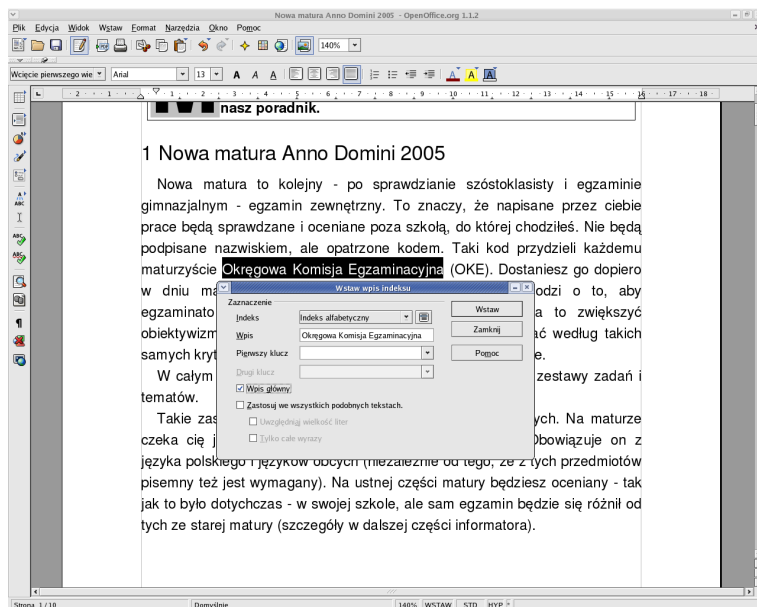
1. Określenie haseł, które mają pojawić się w indeksie, w tym:

- wpisu głównego
- wpisów zwykłych
- wpisów z dodatkowymi kluczami

2. Wygenerowanie właściwego indeksu.

Wpis do indeksu może być główny, zwykły lub może zawierać dodatkowe klucze. Wpis główny to wpis, który chcemy wyróżnić — na przykład numer strony z definicją pojęcia, podczas gdy pozostałe wpisy dotyczą odwołań do tego pojęcia. Wpisy z dodatkowymi kluczami umożliwiają tworzenie wpisów złożonych. Możemy na przykład stworzyć wpis zawierający odsyłacze do wszystkich wystąpień nazwiska Euler, a wśród tych spisów umieścić odsyłacz do *twierdzenia* Eulera.

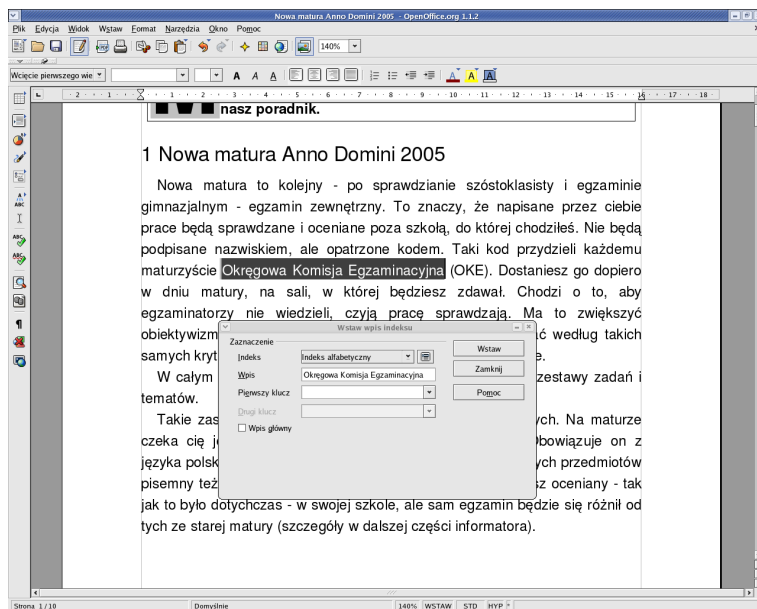
Określanie haseł, które mają się pojawić w indeksie wyjaśnimy na przykładzie. Przypuśćmy, że chcemy wstawić do indeksu hasło „Okręgowa Komisja Egzaminacyjna”. Chcemy przy tym, aby pierwsze wystąpienie tego hasła było wpisem głównym, a pozostałe — zwykłymi. W tym celu rozpoczynamy od znalezienia w dokumencie pierwszego wystąpienia wzorca „Okręgowa Komisja Egzaminacyjna”. Edytor podświetla znaleziony napis. Następnie wybieramy *Wstaw* ► *Indeksy i spis treści* ► *Wpis*. Pojawia się dialog *Wstaw wpis indeksu ze znalezionym hasłem* wpisanym już do pola *Wpis*:



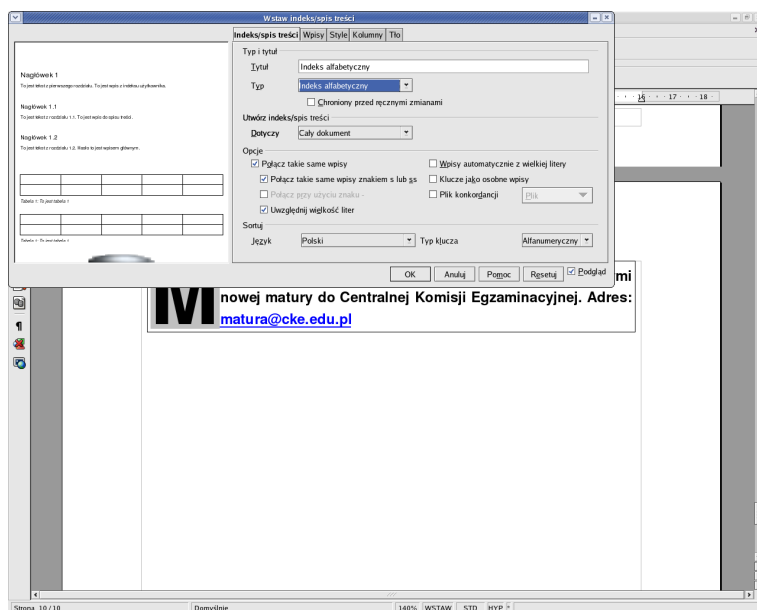
Ponieważ chcemy, aby to wystąpienie było wpisem głównym zaznaczamy pole *Wpis główny* i klikamy klawisz *Wstaw*, po czym zamykamy dialog.

Tekst „Okręgowa Komisja Egzaminacyjna” został podświetlony na szaro, co oznacza, że jest on hasłem w indeksie.

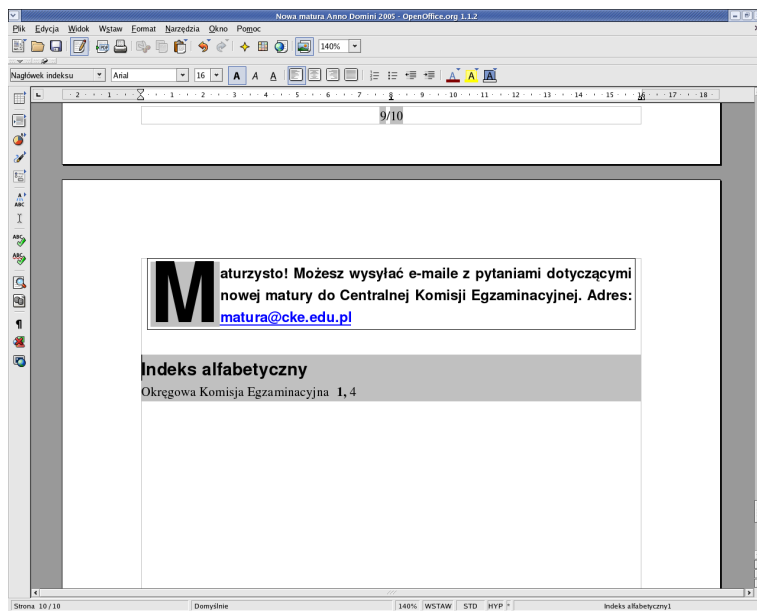
Jeśli chcemy teraz wstawić do indeksu wszystkie pozostałe wystąpienia napisu „Okręgowa Komisja Egzaminacyjna” ponawiamy wyszukiwanie, tym razem jednak wybierając opcję *Znajdź wszystkie*. Ponownie wybieramy *Wstaw* ► *Indeksy i spis treści* ► *Wpis*, tym razem jednak nie zaznaczając pola *Wpis główny* i klikamy klawisz *Wstaw*, po czym zamykamy dialog.



Zobaczmy jak wyglądałby indeks z tak dodanymi hasłami. W tym celu ustawiamy kursor w tym miejscu dokumentu, w którym chcemy wstawić indeks i wybieramy *Wstaw* ► *Indeksy i spis treści* ► *Indeksy i spis treści*. W polu *Typ* wybieramy *Indeks alfabetyczny*.



Po kliknięciu klawisza *OK* w dokumencie pojawia się podświetlony na szaro indeks:

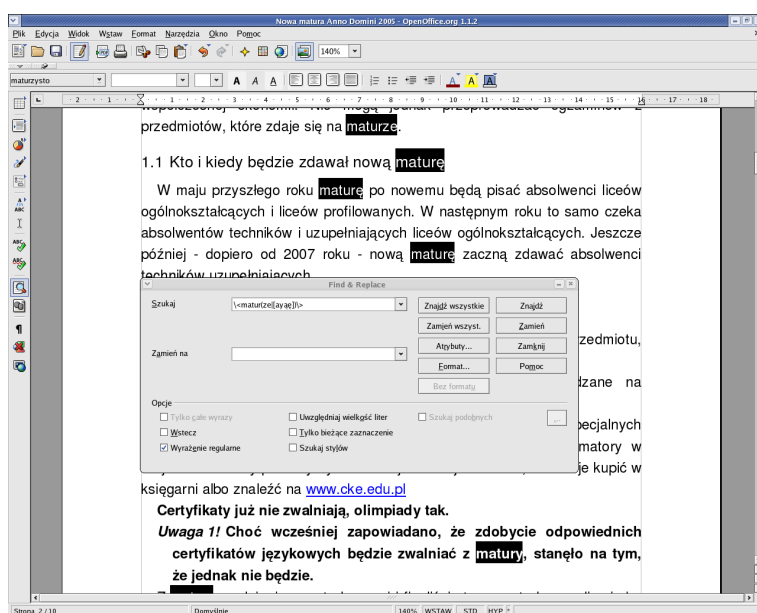


Numer strony dotyczącej wpisu głównego jest przy tym wyróżniony pogrubioną czcionką, pozostałe strony są składane czcionką zwykłą.

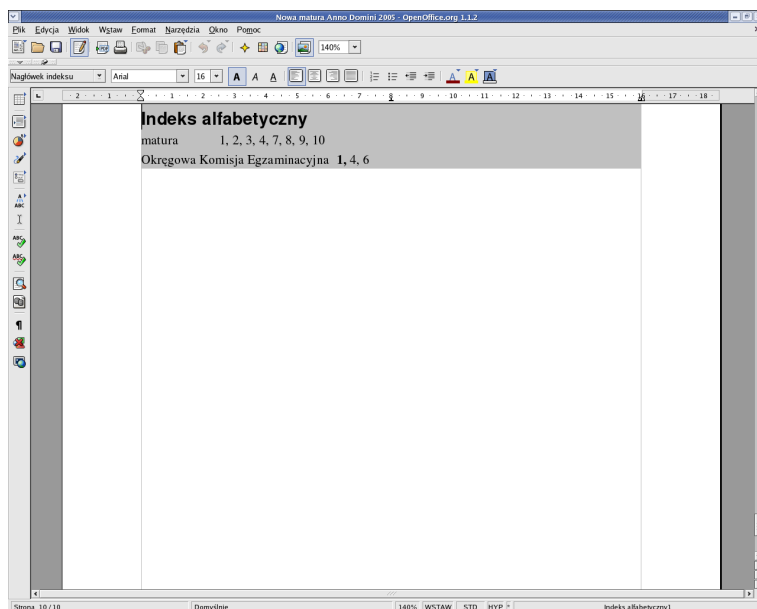
Indeks można usunąć zaznaczając myszką dowolny jego fragment i klikając w niego prawym klawiszem myszy. Pojawi się wówczas menu podręczne, z którego wybieramy opcję *Usuń* (tak jak w przypadku spisu treści).

Spróbujmy teraz trudniejszej rzeczy. Wstawmy do indeksu wszystkie wystąpienia słowa „matura” we wszystkich przypadkach pod hasło „matura”.

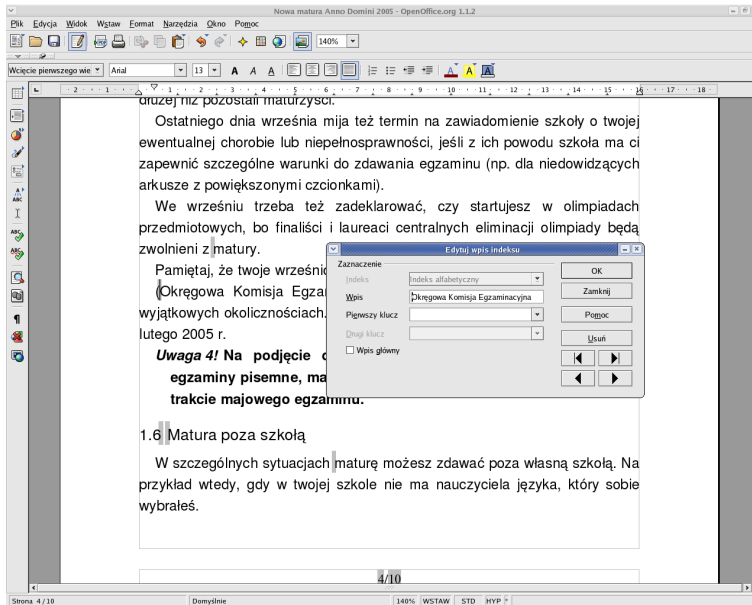
Posłużymy się w tym celu wyszukiwaniem wyrażenia regularnego. Wyszukamy wszystkie wystąpienia wyrażenia regularnego „\ < matur([ayęą]—ze\ >”.



Następnie wywołajmy dialog *Wstaw ► Indeksy i spis treści ► Wpis*, wpisując w polu *Wpis* słowo „matura” (chcemy, aby wszystkie zaznaczone słowa pojawiły się pod tym samym hasłem „matura”). Klikamy *Wstaw* i *Zamknij*, po czym ponownie wstawiamy indeks do dokumentu.



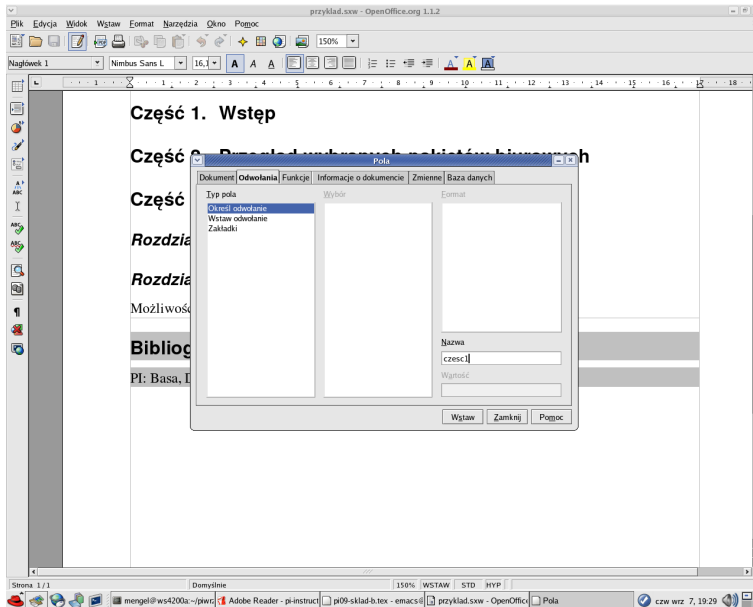
Wpisy do indeksu można poddawać edycji, a także usuwać. Jeśli chcemy usunąć wpis z indeksu, musimy znaleźć miejsce jego wystąpienia w tekście. Widząc w indeksie hasło „Okręgowa Komisja Egzaminacyjna” ze stroną numer 4, udajemy się na stronę czwartą szukając podświetlonego na szaro napisu „Okręgowa Komisja Egzaminacyjna” (czasem podświetlenie występuje **przed** napisem), ustawiamy kursor przed nim i klikamy prawym klawiszem myszy wybierając opcję *Wpis z indeksu*. Za pomocą wyświetlonego dialogu możemy edytować wpis lub też usunąć go. Strzałki w dole okienka służą do przemieszczania się między kolejnymi wpisami do indeksu.



10.14 Odsyłacze

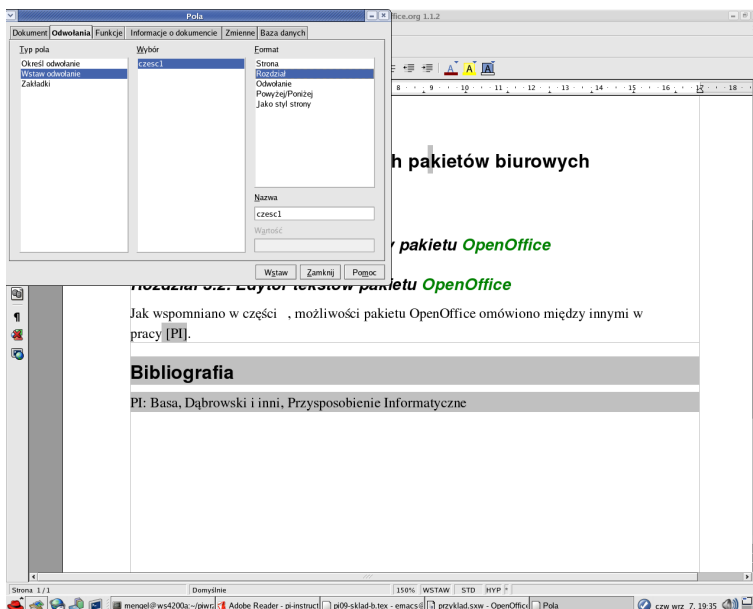
Mając ponumerowane rozdziały możemy odwoływać się do nich w tekście, pisząc na przykład coś w rodzaju „patrz rozdział 2”. Nie jest to jednak dobra praktyka. Wstawiając w tekście konkretne numery rozdziałów ryzykujemy, że po późniejszych ewentualnych zmianach struktury pracy i dodaniu nowych rozdziałów, trzeba będzie przejrzeć cały dokument sprawdzając, że odsyłacze są poprawne. Lepszym sposobem jest nazwanie fragmentów tekstu, do których chcemy się odwołać i użycie w odnośniku nazwy tego fragmentu. Jak to zrobić?

Przypuśćmy, że w rozdziale 3.2 chcemy odwołać się do części 2 naszego dokumentu. Zaczynamy od nazwania fragmentu, *do którego* chcemy się odwołać, czyli części 2. W tym celu ustawiamy kursor na nagłówku tej części i wybieramy *Wstaw ► Odsyłacz*. W dialogu, który się ukaże, zaznaczamy opcję *Określ odwołanie* i w polu *Nazwa* nadajemy mu dowolnie wybraną nazwę, np. *czesc1*:

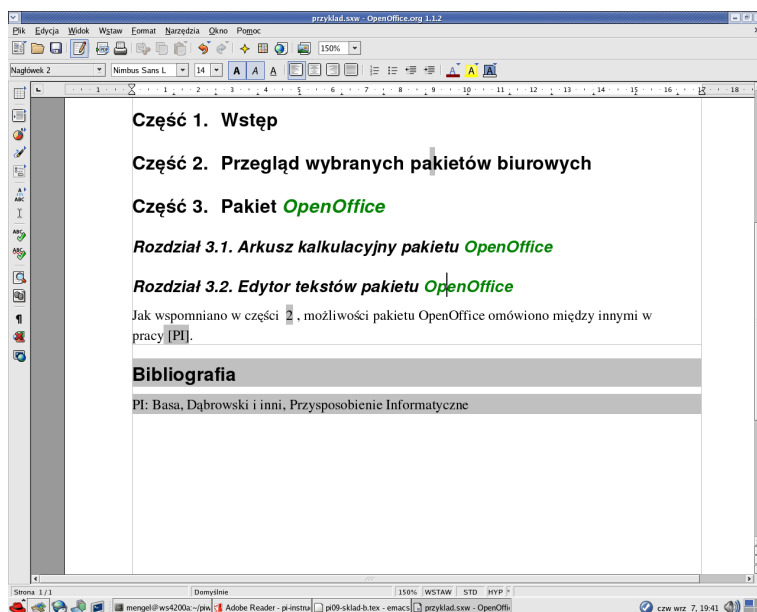


Następnie wciskamy myszką klawisz *Wstaw* i zamykamy dialog. W nagłówku części pierwszej, w miejscu gdzie znajdował się kursor pojawia się szary prostokąt (jeśli jest włączone cieniowanie pól).

Teraz ustawiamy kursor w miejscu, w którym chcemy wstawić odsyłacz. Ponownie wywołujemy ten sam dialog za pomocą *Wstaw ► Odsyłacz*, ale tym razem zaznaczamy opcję *Wstaw odwołanie*. W okienku *Wybór* odnajdujemy nazwę miejsca, do którego chcemy się odwołać, a w polu *Format* określamy sposób odwołania. Jeśli na przykład wybierzemy *Strona*, to w dokumencie pojawi się numer strony, na której znajduje się rozdział, do którego się odwołujemy. Wartość *Rozdział* oznacza numer rozdziału:



Znów wciskamy klawisz *Wstaw* i zamykamy dialog. W tekście pojawiła się podświetlona na szaro liczba 2 oznaczająca numer rozdziału:



Ćwiczenie 10.14.1. Wstaw na początku pracy nową część. Wybierz następnie *Narzędzia* ► *Aktualizuj*. Zaobserwuj, co stanie się z odsyłaczem. Wiesz już, dlaczego taki sposób jest lepszy niż wstawianie konkretnego numeru rozdziału?

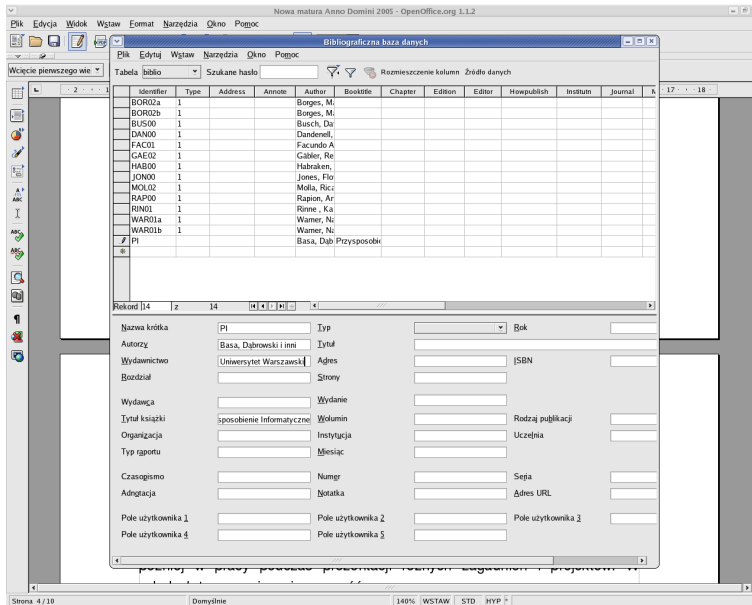
10.15 Bibliografia

Edytor pakietu *OpenOffice* stwarza możliwości wygodnego przygotowania bibliografii. Użytkownik może wprowadzić literaturę, z której korzysta w swoich pracach do bibliograficznej bazy danych. Zawiera ona wszystkie informacje o poszczególnych pozycjach takie jak: tytuł, autorów, rodzaj (książka, artykuł, raport itd), rok wydania, liczba stron itp. Baza ta zawiera nie tylko pozycje, do których chcemy się odwoływać w jednym konkretnym dokumencie. Zazwyczaj tworzy się jedną bazę danych, z której korzysta się w wielu dokumentach.

Chcąc odwołać się do pozycji z bazy danych wstawia się w tekście odsyłacze. Pod koniec dokumentu umieszcza się bibliografię, do której system automatycznie dopisuje te i tylko te prace z bazy danych, do których wystąpiło odwołanie w tekście.

Widać więc, że przygotowanie bibliografii składa się z trzech etapów: przygotowania i utrzymywania bazy danych, wstawienia odsyłaczy w tekście oraz wygenerowania bibliografii. Przyjrzyjmy się tym etapom po kolei.

Dostęp do bazy danych uzyskujemy wybierając opcję *Narzędzia* ► *Bibliograficzna baza danych*. Pojawia się wtedy okienko:

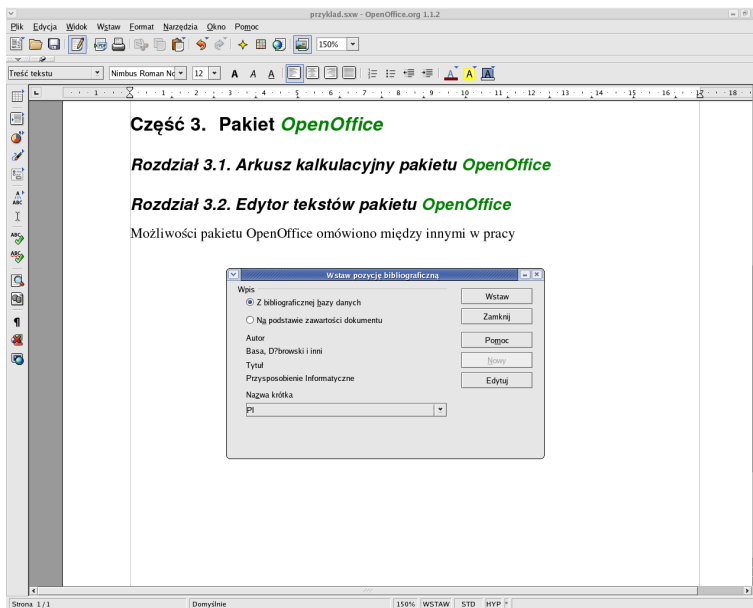


W górnej jego części są wyświetlane prace znajdujące się już w bazie danych. W poszczególnych kolumnach są różne informacje o pracach takie jak autorzy, tytuł, wydawnictwo itp. W pierwszej kolumnie jest wyświetlany identyfikator pracy, czyli nadana przez użytkownika nazwa, którą będziemy się posługiwać odwołując się do tej pracy w dokumencie.

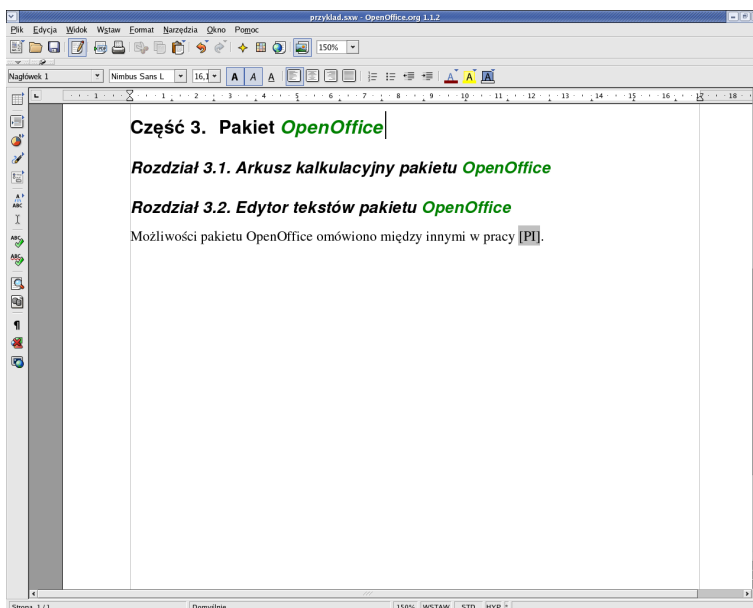
W dolnej części okienka znajdują się szczegóły dotyczące aktualnie podświetlanej pracy, czyli pracy przy której w górnej części okienka znajduje się trójkącik.

Dokładne omówienie operacji możliwych na bazie danych przekracza ramy tego podręcznika. Spróbujemy jedynie utworzyć nową pozycję. W tym celu klikamy na ostatni (pusty) wiersz w górnej części okienka, a w dolnej wypełniamy poszczególne pola. Dodajmy na przykład odsyłacz, do niniejszego podręcznika. Nadajmy mu nazwę „PI” i ustawmy typ na „Książka”. Następnie wypełnijmy pola *Autorzy*, *Tytuł*, *Wydawnictwo* i zamknijmy okienko. Dane o pracy zostały zapamiętane.

Przypuśćmy teraz, że chcemy odwołać się w dokumencie do wprowadzonej pracy. W tym celu ustawiamy kursor w miejscu, gdzie chcemy wstawić odsyłacz i wybieramy opcję *Wstaw ► Indeksy i spis treści ► Pozycja bibliograficzna*. Pojawia się dialog:

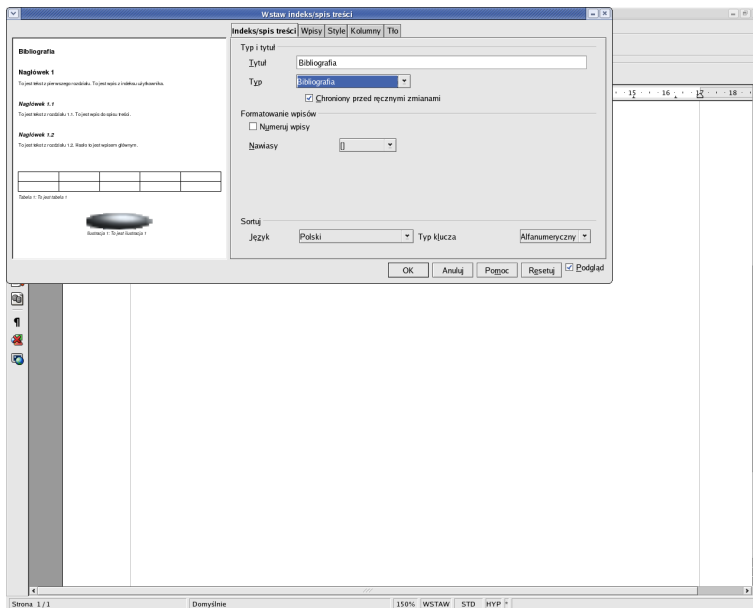


W polu wyboru wybieramy nazwę pracy, która pojawia się w dokumencie, co skutkuje umieszczeniem w nim odnośnika:

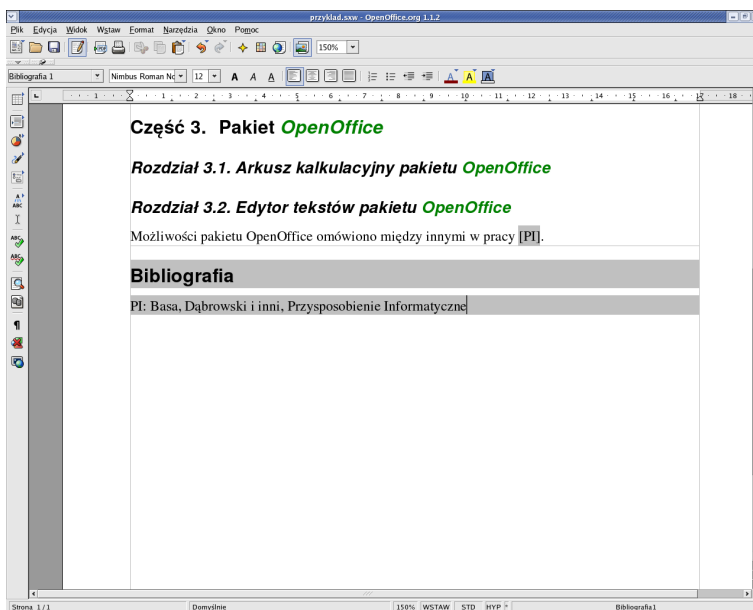


Czynność tę powtarzamy, przy każdym odwołaniu do pracy z bibliograficznej bazy danych.

Ostatnim etapem jest umieszczenie w dokumencie spisu literatury. W tym celu ustawiamy kursor w miejscu, gdzie ma się pojawić bibliografia i wybieramy *Wstaw ► Indeksy i spis treści ► Indeksy i spis treści*. Pojawia się znany nam już dialog, w którym w polu *Typ* wybieramy wartość *Bibliografia*.



W dokumencie zostaje umieszczony spis bibliograficzny zawierający te i tylko te pozycje z bazy danych, do których nastąpiło odwołanie w tekście.



Możemy decydować, w jaki sposób jest składana bibliografia. Wystarczy kliknąć prawym klawiszem myszy na bibliografię i z menu podręcznego wybrać opcję *Edytuj indeks*. Eksperymenty z różnymi ustawieniami pozostawiamy już jednak Czytelnikowi.

Rozdział 11

System składu dokumentów \LaTeX

11.1 Wprowadzenie do systemu \LaTeX

Pierwsza powszechnie dostępna wersja systemu \LaTeX pojawiła się w 1985 roku. W odróżnieniu od wielu pakietów służących do składania dokumentów jest to system bezpłatny, a do tego dostępny na różne platformy sprzętowe i różne systemy operacyjne.

Od swojego powstania system \LaTeX zyskiwał coraz większą popularność głównie w środowisku naukowym, ale także wydawniczym. Powodem tego jest zaszyta w niego wiedza o zasadach składu tekstu z reguły obca przeciętnemu użytkownikowi. \LaTeX potrafi sam dobrać rozmiary czcionki, rozmiar odstępu między wierszami i między akapitami tak, aby dokument miał możliwie najlepszą postać graficzną.

System charakteryzuje się przy tym dużą elastycznością. Można w miarę łatwo dostosować go do konkretnych potrzeb oraz rozszerzać jego funkcjonalność. W połączeniu z jego dostępnością spowodowało to powstanie dużej liczby tzw. pakietów przeznaczonych do przygotowywania różnorodnych tekstów: muzycznych, publikacji szachowych, brydżowych, chemicznych, matematycznych, krzyżówek i wielu innych. System jest też wykorzystywany w wielu wydawnictwach na całym świecie.

Podstawowa filozofia systemu polega na tym, aby zwolnić autora tekstu z konieczności myślenia o tym, *jak* dokument ma wyglądać. Wielkość czcionki, odstępy między wierszami, znaki poprzedzające poszczególne punkty wyliczenia, numeracja list i rozdziałów, podział na strony, a nawet łamanie wyrazów na końcu wiersza — tym wszystkim zajmuje się system. Oczywiście użytkownik musi dostarczyć mu do tego niezbędnych informacji, ale nie są to polecenia typu „zastosuj czcionkę 12pt”. Użytkownik informuje system, jaka jest struktura rozdziału mówiąc na przykład: „ten akapit jest tytułem kolejnego rozdziału”. Informacje te są przy tym wpisywane w sam dokument w postaci poleceń. Podobne zasady obowiązują w języku HTML stosowanym przy tworzeniu stron WWW, o czym będziemy mówić

w dalszej części kursu.

Aby system mógł sprostać takim wyzwaniom musi być jednak spełniony jeden warunek. \LaTeX musi znać cały tekst (a przynajmniej odpowiednio duży jego fragment) *zanim* przystąpi do jego składania. Z tego powodu przygotowanie dokumentu w \LaTeX u wygląda zupełnie inaczej niż w pakiecie *OpenOffice*. Przede wszystkim użytkownik nie widzi dokumentu w jego ostatecznej postaci. To nie jest system WYSIWYG, co początkowo sprawia pewne trudności. Autor przygotowuje najpierw plik źródłowy, który jest następnie przetwarzany (kompilowany) przez system \LaTeX . Kompilację trzeba przy tym przeprowadzić po każdej zmianie wprowadzonej w tekście. Dopiero skompilowany tekst można obejrzeć na ekranie lub wydrukować.

11.2 Schemat pracy z systemem \LaTeX

Praca z systemem \LaTeX jest zupełnie inna niż w przypadku edytorów typu WYSIWYG. Dużym utrudnieniem, przynajmniej początkowo, jest to, że autor nie widzi końcowej postaci dokumentu.

Przygotowanie dokumentu rozpoczyna się od przygotowania pliku źródłowego. Jest to zwykły plik tekstowy, więc do jego tworzenia można wykorzystać zwykły edytor plików tekstowych, taki jak na przykład *Notatnik* w systemie *Microsoft Windows*, czy *Emacs* w systemie *Linux*. Plik źródłowy zawiera tekst dokumentu, w którym dodatkowo znajdują się specjalne polecenia dla systemu \LaTeX — wprowadzane również w postaci tekstowej. Polecenia te informują system, jak ma być składany tekst. Na przykład polecenie `\emph` nakazuje wyróżnienie wskazanego fragmentu.

```
\begin{document}
\maketitle

\section{Wstęp}
Dynamiczny rozwój technik komputerowych, obserwowany ostatnimi laty,
spowodował upowszechnienie się w wielu dziedzinach życia urządzeń
zawierających mikroprocesory. Ich prawidłowe działanie, od którego
często zależy nasze zdrowie i życie, jest uzależnione nie tylko od
niezawodności sprzętu, ale także --- a może nawet przede wszystkim ---
od poprawności i niezawodności sterującego nim oprogramowania.

Z konieczności produkcji niezawodnego oprogramowania zdają sobie
sprawę ludzie zaangażowani w jego tworzenie. Nie jest to jednak łatwe
zadanie. Uważa się dość powszechnie, że programy komputerowe są najbardziej
skomplikowanym wytworem myśli ludzkiej. Nic więc dziwnego, że są
prowadzone intensywne badania mające na celu wypracowanie metod
projektowania i wytwarzania poprawnego oprogramowania. Referowana
rozprawa porusza zagadnienia związane ze {\em specyfikowaniem}
systemów komputerowych i badaniem ich {\em poprawności}, a także
poprawności specyfikacji.

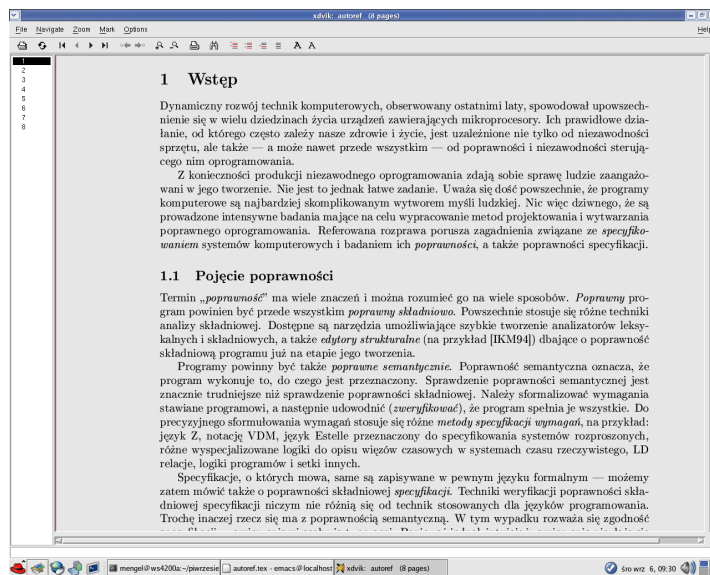
\subsection{Pojęcie poprawności}
Termin „{\em poprawność}” ma wiele znaczeń i można rozumieć go na
wiele sposobów. {\em Poprawny} program powinien być przede wszystkim
{\em poprawny składniowo}. Powszechnie stosuje się różne techniki
analizy składniowej. Dostępne są narzędzia umożliwiające szybkie
tworzenie analizatorów leksykalnych i składniowych, a także {\em
edytory strukturalne} (na przykład {\code{\texttt{EM:trace-editor}}}) dbające o
poprawność składniową programu już na etapie jego tworzenia.

Programy powinny być także {\em poprawne semantycznie}. Poprawność
semantyczna oznacza, że program wykonuje to, do czego jest
przeznaczony. Sprawdzenie poprawności semantycznej jest znacznie
trudniejsze niż sprawdzenie poprawności składniowej. Należy
sformalizować wymagania stawiane programowi, a następnie udowodnić
({\em zweryfikować}), że program spełnia je wszystkie. Do
precyzyjnego sformułowania wymagań stosuje się różne {\em metody
specyfikacji wymagań}, na przykład: język
Z, notacja
VDM, język Estelle przeznaczony do
specyfikowania systemów rozproszonych,
różne wyspecjalizowane logiki do opisu wzmów czasowych w systemach
czasu rzeczywistego, ID relacje,
logiki programów i setki innych.
```

Nazwa pliku źródłowego powinna kończyć się rozszerzeniem `.tex`. Po przygotowaniu pliku źródłowego i jego zapisaniu na dysk, należy go *skompilować*.

Sposób kompilacji zależy od systemu operacyjnego, pod którym uruchamiamy \LaTeX oraz jego dystrybucji. Pod systemem Linux uruchamia się na przykład program *latex*, którego argumentem jest nazwa pliku źródłowego. System \LaTeX w tym momencie rozpoczyna przetwarzanie tekstu i tworzenie dokumentu, który zapisuje w postaci pliku o tej samej nazwie, co plik źródłowy lecz na końcu zamiast *.tex* umieszcza przyrostek *.dvi*.

Plik ten można następnie obejrzeć na ekranie za pomocą przeglądarki. Jest to specjalny program, o nazwie zależnej od konkretnego systemu operacyjnego i dystrybucji systemu \LaTeX , który potrafi wyświetlić dokument w końcowej postaci. Teraz dopiero widać, jak będzie wyglądać końcowy dokument, gdzie będą przejścia do nowych wierszy, łamania stron itp.



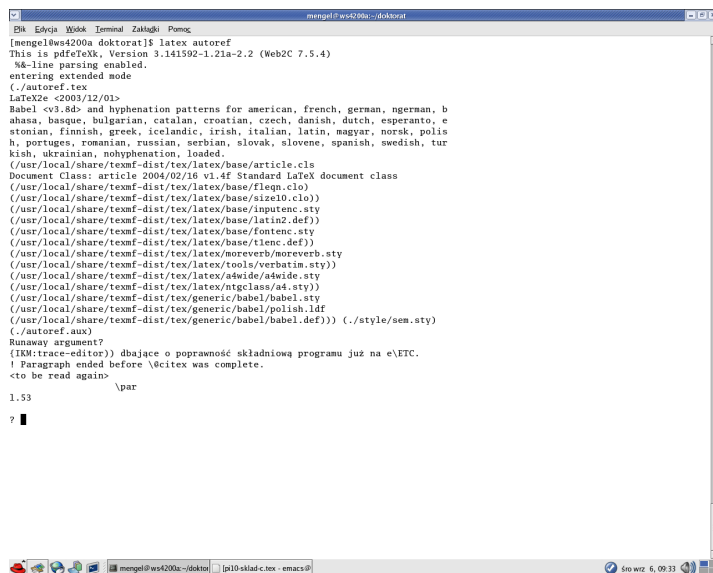
Najczęściej na tym etapie okazuje się, że nie wszystko wygląda tak, jak powinno. Niestety nie możemy dokonywać zmian za pomocą przeglądarki. Trzeba powrócić do pliku źródłowego i w nim dokonać niezbędnych korekt. Następnie jest konieczne ponowne skompilowanie pliku źródłowego i uruchomienie przeglądarki. Proces ten powtarza się tak długo, aż będziemy zadowoleni z końcowej postaci dokumentu. Wtedy można go wydrukować.

Niektóre przeglądarki pozwalają przesłać dokument od razu na drukarkę poprzez wybranie odpowiedniej opcji z menu. Czasem jednak dokument należy przekształcić na dokument w formacie Postscript za pomocą specjalnego konwertera (*dvips*) i dopiero taki plik można wydrukować.

1. Utwórz plik źródłowy
2. Skompiluj go. Jeśli wystąpią błędy kompilacji wróć do punktu 1.
3. Uruchom przeglądarkę i obejrzyj dokument. Jeśli trzeba coś poprawić wróć do punktu 1.
4. Wydrukuj dokument.

Warto także wspomnieć, że za pomocą odpowiednio skonfigurowanego systemu \LaTeX można uzyskać także dokumenty w formacie *PDF*.

Przedstawiony proces przygotowywania dokumentu wydaje się dość żmudny i niewygodny. Sprawę dodatkowo komplikuje fakt, że źle przygotowany plik źródłowy powoduje błędy podczas kompilacji, a kompilator nie wyjaśnia ich w przyjazny dla użytkownika sposób:



```
mmjgf@ws200a-~$ latex autoref
This is pdfTeX, Version 3.141592-1.21a-2.2 (Web2C 7.5.4)
%&-line parsing enabled.
entering extended mode
{./autoref.tex
LaTeX2e <2003/12/01>
Babel <3.84> and hyphenation patterns for american, french, german, ngerman, b
ahasa, basque, bulgarian, catalan, croatian, czech, danish, dutch, esperanto, e
stonian, finnish, greek, icelandic, irish, italian, latin, magyar, norsk, polis
h, portugese, romanian, russian, serbian, slovak, slovene, spanish, swedish, tur
kish, ukrainian, nonhyphenation. loaded.
(/usr/local/share/texmf-dist/tex/latex/base/article.cls
Document Class: article 2004/02/16 v1.4f Standard LaTeX document class
(/usr/local/share/texmf-dist/tex/latex/base/floats.clo)
(/usr/local/share/texmf-dist/tex/latex/base/size10.clo)
(/usr/local/share/texmf-dist/tex/latex/base/inputenc.sty
(/usr/local/share/texmf-dist/tex/latex/base/latin2.def)
(/usr/local/share/texmf-dist/tex/latex/base/fontenc.sty
(/usr/local/share/texmf-dist/tex/latex/base/tlenc.def)
(/usr/local/share/texmf-dist/tex/latex/moreverb/moreverb.sty
(/usr/local/share/texmf-dist/tex/latex/tools/verbatim.sty)
(/usr/local/share/texmf-dist/tex/latex/adwide/adwide.sty
(/usr/local/share/texmf-dist/tex/latex/ntgclass/a4.sty)
(/usr/local/share/texmf-dist/tex/generic/babel/babel.sty
(/usr/local/share/texmf-dist/tex/generic/babel/polish.ldf
(/usr/local/share/texmf-dist/tex/generic/babel/babel.def)) (.style/sen.sty)
{./autoref.aux)
Runaway argument?
{TKM:trace=editor} dbajace o poprawnośc składniową programu juz na e/ETC.
! Paragraph ended before \catex was complete.
<to be read again>
\par
1.53
? █
```

Jednak w miarę używania systemu dochodzi się do wprawy, a końcowy wygląd dokumentu i możliwości systemu w pełni rekompensują niedogodności związane ze skomplikowanym procesem tworzenia dokumentu.

System \LaTeX jest chętnie stosowany przez matematyków, informatyków, fizyków ze względu na możliwości składania złożonych wzorów i formuł matematycznych. Często okazuje się, że czas potrzebny na przygotowanie porządnie wyglądającej pracy matematycznej za pomocą systemu \LaTeX jest krótszy niż przy użyciu jakiegokolwiek innego edytora typu WYSIWYG.

11.3 Struktura pliku źródłowego

Jak wspomnieliśmy wcześniej pracę nad dokumentem zaczyna się od przygotowania pliku źródłowego. Jest to zwykły plik tekstowy, który można utworzyć za pomocą dowolnego edytora plików tekstowych. W pliku tym umieszczamy tekst dokumentu, w którym znajdują się także polecenia systemu \LaTeX . Polecenia rozpoczynają się od wstecznego ukośnika (ang. *backslash*), po którym następuje jedno z rozumianych przez system słów kluczowych. Polecenia mogą mieć argumenty, które ujmujemy w nawiasy $\{$ i $\}$.

Każdy dokument rozpoczyna się od polecenia `\documentclass`. Argumentem tego polecenia jest nazwa klasy dokumentu, który przygotowujemy. \LaTeX oferuje kilka standardowych klas dokumentów. Są to m.in.:

- *article*
 - stosowana do przygotowywania krótkich dokumentów
- *report*
 - stosowana dla dłuższych tekstów o bardziej rozbudowanej strukturze
- *book*
 - używana do przygotowywania książek
- *letter*
 - przydatna do składania listów

Wybór określonej klasy determinuje wygląd dokumentu oraz takie jego parametry jak: jednostki wchodzące w skład dokumentu (części, rozdziały, podrozdziały itd), sposób składania nagłówka i stopki strony, wymuszanie nowych stron przed każdym nowym rozdziałem i wiele innych. W dalszej części podręcznika będziemy korzystać jedynie z klasy `article`.

Klasy dokumentów mają jeszcze pewne opcje. Na przykład decydując się na klasę `article` można jeszcze określić (między innymi) następujące parametry:

- *11pt*
 - dokument ma być składany czcionką 11 punktów zamiast standardowej 10-punktowej
- *12pt*
 - dokument ma być składany czcionką 12 punktów zamiast standardowej 10-punktowej
- *twoside*
 - przygotowuje dokument tak, aby nadawał się do druku dwustronnego
- *twocolumn*
 - składa dokument w dwóch kolumnach

Opcje umieszcza się, rozdzielając je przecinkami bez żadnych spacji, w nawiasach kwadratowych między nazwą polecenia a jego parametrem. Dokument może zaczynać się od polecenia: `\documentclass[12pt]{article}`.

Oprócz poleceń standardowych dla danej klasy dokumentu, przygotowując tekst można skorzystać z tzw. *pakietów*. Wprowadzają one dodatkowe polecenia ułatwiające składanie specyficznych dokumentów. W standardowej dystrybucji L^AT_EXa jest wiele różnych pakietów. Oto niektóre przykłady:

- *babel*
 - umożliwia składanie wielojęzycznych dokumentów
- *latsym*
 - wprowadza dodatkowe symbole matematyczne
- *seminar*
 - umożliwia przygotowywanie slajdów i prezentacji elektronicznych
- *musictex*
 - daje możliwości składania utworów muzycznych, definiując wszystkie niezbędne obiekty potrzebne do tego: pięciolinie, klucze, nuty itd.
- *chess*
 - umożliwia składanie diagramów szachowych
- i wiele, wiele innych do składania krzyżówek, publikacji brydżowych, chemicznych, grafów, diagramów,

Gdy chcemy skorzystać z pewnego pakietu, to po nagłówku dokumentu umieszczamy polecenia `\usepackage`, którego argumentem jest nazwa pakietu.

We wszystkich przykładach omawianych w tym podręczniku będziemy korzystać z następujących pakietów:

- *babel*
 - z opcją *polish*, aby wygodnie składać dokumenty w języku polskim
- *fontenc*
 - z opcją *T1*, który oferuje polską czcionkę
- *inputenc*
 - z opcją *latin2*, jeśli pracujemy w środowisku Linux lub opcją *cp850* w środowisku Windows, który umożliwia korzystanie z polskich liter wprowadzanych w standardowy sposób, czyli za pomocą lewego klawisza *Alt*
- *a4wide*
 - ustawiający rozmiar strony na A4

Każdy dokument przygotowywany przez nas dla systemu \LaTeX będzie zatem rozpoczynał się w ten sam sposób:

```

\documentclass[12pt]{article}

\usepackage[latin2]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[polish]{babel}
\usepackage{a4wide}

```

Właściwy tekst dokumentu rozpoczyna się od polecenia `\begin{document}` a kończy na `\end{document}`. Fragment pliku źródłowego rozpoczynający się od nagłówka (deklaracji klasy dokumentu), a kończący się na poleceniu `\begin{document}` będziemy nazywać *preambułą*. W preambule umieszczamy m.in. deklaracje używanych pakietów.

Teraz możemy już przygotować pierwszy poprawny dokument.

```

\documentclass[12pt]{article}
\usepackage[latin2]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[polish]{babel}
\usepackage{a4wide}
\begin{document}
\end{document}

```

Zawiera on wszystkie niezbędne elementy: preambułę oraz właściwą treść dokumentu, która jest jednak pusta. Przez to dokument jest mało ciekawy — zawiera po prostu jedną pustą stronę. W takiej sytuacji \LaTeX nie generuje nawet pliku wynikowego.

Spróbujmy przygotować teraz krótką notatkę. Jej treść wpisujemy między polecenia `\begin{document}` a `\end{document}`. Przygotowując tekst używamy wielkich i małych liter alfabetu polskiego, cyfr, znaków przestankowych i symboli: `. : ; , ? ! () [] - / * .` Pozostałe symbole uzyskuje się w inny, specjalny sposób.

Sposób, w jaki rozmieszcza się tekst w pliku źródłowym (prawie) nie ma znaczenia dla wyglądu ostatecznego dokumentu. Oznacza to, że możesz wciskać klawisz *Enter* w dogodnych dla siebie miejscach, dzieląc tekst źródłowy na wiersze, jak Ci się podoba, możesz oddzielać wyrazy za pomocą pojedynczych spacji lub wstawiać ich dowolnie wiele — \LaTeX i tak przeanalizuje cały dokument i dokona podziału na wiersze tak, aby wyglądał jak najlepiej.

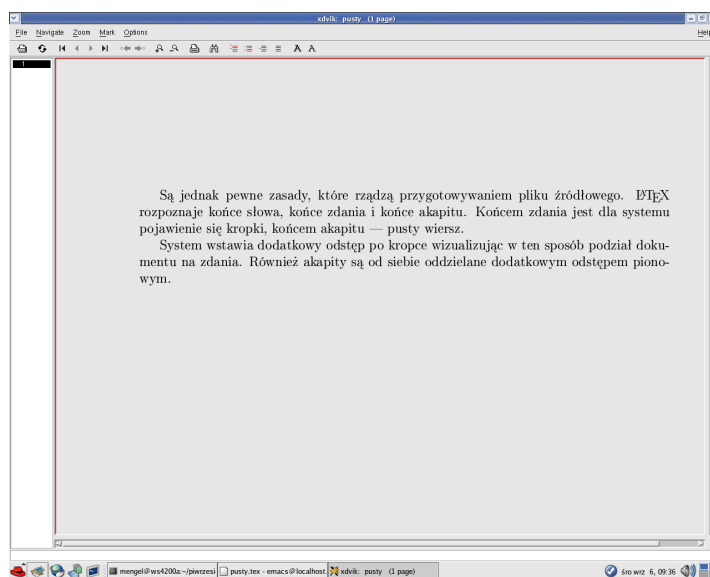
Są jednak pewne zasady, które rządzą przygotowaniem pliku źródłowego. \LaTeX rozpoznaje końce słowa, końce zdania i końce akapitu. Końcem zdania jest dla systemu pojawienie się kropki, końcem akapitu — pusty wiersz. System wstawia dodatkowy odstęp po kropce wizualizując w ten sposób podział dokumentu na zdania. Również akapity są od siebie oddzielane dodatkowych odstępem pionowym.

Popatrzmy na przykład. Plik źródłowy zawierający tekst:

Są jednak pewne zasady, które rządzą przygotowaniem pliku źródłowego. \LaTeX rozpoznaje końce słowa, końce zdania i końce akapitu. Końcem zdania jest dla systemu pojawienie się kropki, końcem akapitu --- pusty wiersz.

System wstawia dodatkowy odstęp po kropce wizualizując w ten sposób podział dokumentu na zdania. Również akapity są od siebie oddzielane dodatkowym odstępem pionowym.

po skompilowaniu wygląda tak:



Jak pokazuje powyższy przykład \LaTeX nie tylko dzieli tekst na wiersze, ale potrafi też przenosić wyrazy. W tekście wystąpiły ponadto trzy interesujące konstrukcje. Pierwsza z nich to polecenie \LaTeX , które w gotowym dokumencie wygląda tak: \LaTeX .

Drugie polecenie to --- . Składając dokumenty stosuje się cztery rodzaje kresczek:

- *łącznik*
 - używamy w wyrazach złożonych, jak w czarno-biały. W tekście źródłowym zapisujemy go jako -.
- *półpauza*
 - stosowana do zakresów liczb, np.:strony 3–5. W tekście źródłowym jest zapisywany jako --.
- *myślnik*

– jak w poprzednim przykładzie zapisywany jako ---.

- *minus*

– występujący w formułach matematycznych, np $4 - 7$.

Trzecia konstrukcja występująca w omawianym przykładzie to `\` (ukośnik ze spacją). Nakazuje on pozostawienie spacji w dokumencie. Gdybyśmy napisali `\LaTeX` a zamiast `\LaTeX\` a, to w składanym dokumencie pojawiłby się tekst `LATEXa`. Spacja wymusza pozostawienie dodatkowego miejsca po tzw. *makrach*. Innym częstym użyciem tego polecenia jest zapobieganie wstawianiu dodatkowego miejsca po kropce. `LATEX` uznaje, że każda kropka nie występująca po dużej literze kończy zdanie i pozostawia za nią większy odstęp. Czasem jednak kropka jest używana w innym kontekście, na przykład po skrócie. Wtedy należy wymusić po niej normalny odstęp za pomocą omawianego polecenia.

Jeśli w tekście chcemy umieścić cudzysłów używamy dwóch przecinków jako cudzysłowu otwierającego oraz dwóch apostrofów jako cudzysłowu zamykającego:

```
Autorem „Pana Tadeusza” jest Adam Mickiewicz.
```

Autorem „Pana Tadeusza” jest Adam Mickiewicz.

Ze spacjami w tekście związane jest jeszcze jedno ważne zagadnienie. Ponieważ `LATEX` sam rozstrzyga, gdzie złamać wiersz, a robi to nie rozumiejąc tekstu, który przetwarza, czasem może podzielić wiersz w niewłaściwym w punktu widzenia logiki dokumentu miejscu. Przykładowo przejście do nowego wiersza nie powinno występować m.in.:

- w wyrażeniach typu „rozdział 10” przed numerem
- między tytułem a nazwiskiem jak w „dr Kowalski”
- między inicjałami, np.: C. A. R. Hoare

Jeśli chcemy zapobiec łamaniu wiersza w danym miejscu, to zamiast zwykłej spacji umieszczamy w dokumencie źródłowych znak `~`, pisząc na przykład `dr~Kowalski`.

Czasami także zdarza się, że `LATEX` w niewłaściwy sposób dzieli wyraz między wiersze. Można wtedy zasugerować systemowi właściwe miejsce podziału wstawiając w słowo polecenia `\-`. Jeśli na przykład napiszemy w pliku źródłowym wyrażenie `prze\-śli\ -czny`, to słowo to w dokumencie będzie wyglądać zwyczajnie: `prześliczny` (kreski nie będą widoczne). Jeśli jednak słowo to znajdzie się pod koniec wiersza, to system będzie wiedział, że może je podzielić na sylaby we wskazanych miejscach. Algorytm podziału słów wbudowany w system z reguły dobrze radzi sobie z podziałem wyrazów na sylaby, nie ma więc potrzeby wpisywania takich punktów podziału w każdy

wyraz w pliku źródłowym. Można to zrobić dopiero po przygotowaniu całego dokumentu, gdy okaże się, że występują problemy z podziałem pewnych słów.

Za pomocą systemu \LaTeX można składać dokumenty zawierające nie tylko polskie litery, ale także znaki z różnymi znakami diakrytycznymi (akcentami, ogonkami, kropkami itp). Idea konstrukcji takich znaków jest bardzo prosta. Najpierw wpisujemy polecenie, które produkuje potrzebny znak diakrytyczny (np.: polecenie `\"` dodaje dwie kropki nad literą), następnie w nawiasach `{ }` umieszczamy literkę, której ten znak ma dotyczyć. Ponadto za pomocą specjalnych poleceń można otrzymać specjalne litery występujące w niektórych językach.

A oto przykłady poleceń i efektów ich działania:

```
\‘{o} ò \’{o} ó \~{o} ôx
\"{o} ö \~{n} ñ \={o} õ
\.{o} ó \u{o} õ \v{o} õ
\H{o} ó \c{o} q \d{o} q
\c{o} o
```

Są ponadto dostępne znaki specjalne:

```
\i   produkuje i bez kropki: i
\j   produkuje j bez kropki: j
\aa  produkuje å
\AA  produkuje Å
\o   produkuje ø
\O   produkuje Ø
\ss  produkuje ß
\oe  produkuje œ
\ae  produkuje æ
\OE  produkuje Œ
\AE  produkuje Æ
```

11.4 Strona tytułowa

System \LaTeX potrafi sam przygotować stronę tytułową. Jedyne czynnościami, jakie musimy zrobić jest podanie autora, tytułu, daty oraz ewentualnych innych danych.

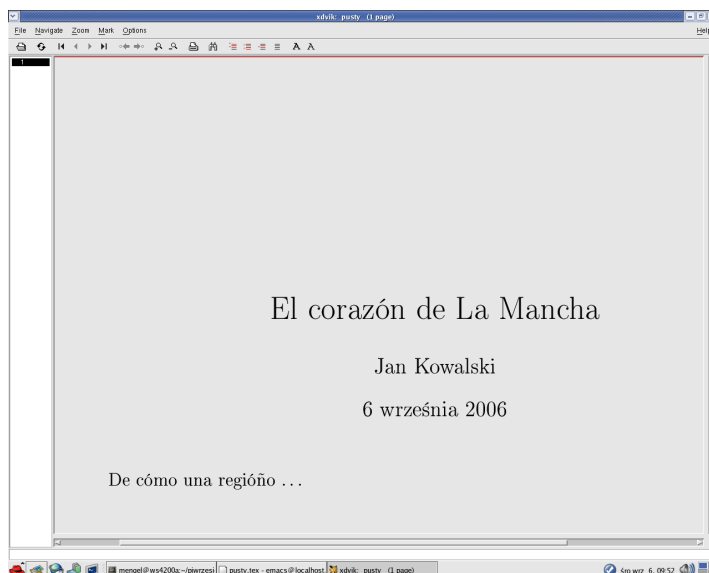
Postać strony tytułowej zależy od zadeklarowanej klasy dokumentów. Jest możliwość zmiany tej postaci graficznej poprzez wykorzystanie odpowiednich pakietów, można też samemu przygotować stronę tytułową, nie jest to jednak tak proste jak w edytorach typu WYSIWYG. W tym podręczniku poprzestaniemy na domyślnej, oferowanej przez system postaci strony tytułowej.

Przygotowanie strony tytułowej polega na umieszczeniu w dokumencie źródłowym polecenia `\maketitle` w miejscu, gdzie ma się pojawić tytuł. Przedtem jednak — najczęściej w preambule — należy umieścić polecenia `\author`,

`\title` z parametrami informującymi odpowiednio, kto jest autorem dokumentu i jaki jest jego tytuł. Oto pełny przykład:

```
\documentclass[12pt]{article}
\usepackage[latin2]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[polish]{babel}
\usepackage{a4wide}
\author{Jan Kowalski}
\title{El coraz\`o de La Mancha}
\begin{document}
\maketitle
De c\`o mo una regi\`o \~{n}o \ldots
\end{document}
```

Po skompilowaniu dokument wygląda następująco:



Jak widzimy system automatycznie dołącza do strony tytułowej aktualną datę. Możemy temu zapobiec umieszczając w preambule polecenie `\date{}`. Możemy również wpisać dowolną datę podając ją jako argument polecenia `\date`, np.: `\date{31 maja 2001}`.

11.5 Podział dokumentu na rozdziały

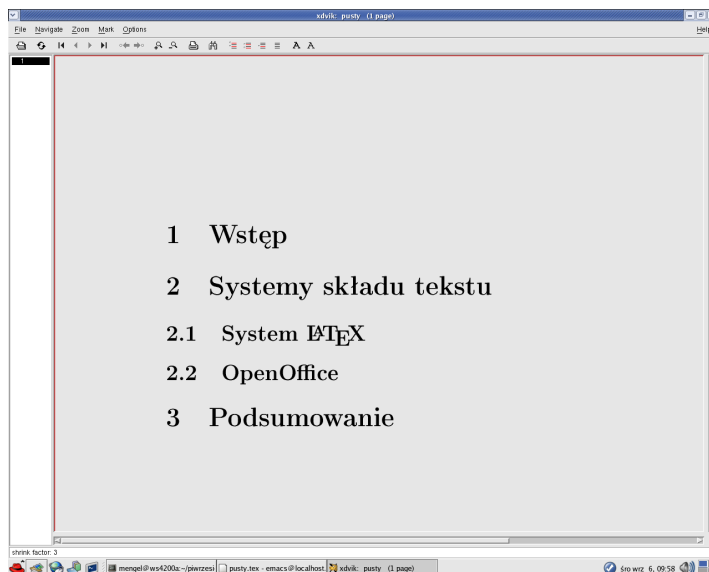
Przygotowując dokument można mu nadać hierarchiczną strukturę dzieląc go na rozdziały, podrozdziały itd. Dokonuje się tego za pomocą poleceń podziału. W klasie `article`, z której korzystamy mamy możliwość utworzenia struktury o pięciu poziomach głębokości. Na najwyższym poziomie jest część uzyskiwana poleceniem `\part`.

Części nie muszą jednak występować. Następnie są rozdziały (polecenie `\section`), podrozdziały (polecenie `\subsection`), punkty (`\subsubsection`), i dalej jeszcze `\paragraph` oraz `\subparagraph`. W praktyce najczęściej korzysta się z rozdziałów, podrozdziałów i punktów.

Argumentem każdego z tych poleceń jest tytuł danej jednostki. \LaTeX sam numeruje wszystkie jednostki dokumentu, sam też dobiera odpowiednią wielkość nagłówka w zależności od klasy dokumentu i użytych opcji (głównie rozmiaru czcionki). Dokument składający się z trzech rozdziałów, z których drugi zawiera dwa podrozdziały wyglądałby więc na przykład tak:

```
\begin{document}
  \section{Wstęp}
  \section{Systemy składu tekstu}
  \subsection{System  $\text{\LaTeX}$ }
  \subsection{OpenOffice}
  \section{Podsumowanie}
\end{document}
```

co po skompilowaniu daje:



11.6 Wyróżnianie, zmiana czcionki

Czasem chcemy wyróżnić pewne elementy dokumentu składając je na przykład kursywą. System \LaTeX udostępnia specjalne polecenie `\emph` służące do wyróżniania fragmentu tekstu. Nie jest jednak powiedziane, jaką czcionką zostanie złożony wyróżniany tekst. \LaTeX sam o tym zdecyduje dbając o to, żeby wskazany fragment dokumentu wyróżniał się w danym kontekście. Najczęściej wyróżniany fragment tekstu będzie składany kursywą. Jeśli jednak zapagniemy dokonać wyróżnienia we fragmencie, który już jest skła-

dany kursywą, to \LaTeX zastosuje czcionkę prostą. Można również samemu zdecydować o rodzaju zastosowanej czcionki. Popatrzmy na przykłady:

<i>kursywa</i>	<code>\textit{kursywa}</code>
<i>pochyła</i>	<code>\textsl{pochyła}</code>
prosta	<code>\textit{prosta}</code>
KAPITALIKI	<code>\textsc{kapitaliki}</code>
wytłuszczenie	<code>\textbf{wytłuszczenie}</code>
antykwą	<code>\textrm{antykwą}</code>
bezszerzefowa	<code>\textsf{bezszerzefowa}</code>
maszynowa	<code>\texttt{maszynowa}</code>

Można również tworzyć kombinacje, np:

```
\textbf{\textit{wytłuszczona kursywa}}
```

daje napis *wytłuszczona kursywa*

11.7 Wyliczenia

Do tworzenia wyliczeń w systemie \LaTeX służą tzw. *otoczenia*. Otoczenie o nazwie *nazwa* zapisuje się zawsze w postaci

```
\begin{nazwa}  
\end{nazwa}
```

wstawiając składany tekst między `\begin{nazwa}` a `\end{nazwa}`. Z otoczeniem zetknęliśmy się już uprzednio: treść dokumentu zapisywaliśmy wewnątrz otoczenia o nazwie `document`.

W \LaTeX u mamy trzy otoczenia do tworzenia wyliczeń. Są to otoczenia: `itemize`, `enumerate` oraz `description`. Działają one bardzo podobnie do siebie tworząc listy wypunktowane. Elementy listy zapisujemy między początkiem otoczenia a jego końcem, rozpoczynając każdy element od polecenia `\item`.

- Otoczenie *itemize* tworzy wyliczenie, którego poszczególne elementy są wypunktowywane, np:
 - Poniedziałek
 - Wtorek
 - ...
- Otoczenie *enumerate* tworzy wyliczenie, którego poszczególne elementy są numerowane, np:
 1. Poniedziałek
 2. Wtorek
 3. ...

- Otoczenie *description* tworzy wyliczenie, którego poszczególne elementy mają podany przez użytkownika opis, np:

pon Poniedziałek
wt Wtorek
reszta ...

Tworzenie wyliczeń przebiega w następujący sposób:

Plik źródłowy:

```
\begin{itemize}
  \item Poniedziałek
  \item Wtorek
  \item \ldots
\end{itemize}
```

daje efekt końcowy:

- Poniedziałek
- Wtorek
- ...

Sposób składania takich wyliczeń, w tym znak jaki pojawia się przed każdym elementem listy określa sam system, choć jest możliwość zmiany tych ustawień. Wyliczenia można zagnieżdżać. Wtedy znaki rozpoczynające wyliczenia wewnętrzne będą różnić się od znaków w wyliczeniach zewnętrznych. Plik źródłowy:

```
\begin{itemize}
  \item Pakiety do składu tekstu:
    \begin{itemize}
      \item OpenOffice
      \item Microsoft Office
      \item \LaTeX
    \end{itemized}
  \item Arkusze kalkulacyjne:
    \begin{itemize}
      \item OpenOffice
      \item Microsoft Office
    \end{itemize}
\end{itemize}
```

powoduje wygenerowanie tekstu:

- Pakiety do składu tekstu:

- OpenOffice
 - Microsoft Office
 - L^AT_EX
- Arkusze kalkulacyjne:
 - OpenOffice
 - Microsoft Office

Przygotowanie listy numerowanej różni się jedynie użyciem otoczenia `enumerate` zamiast `itemize`. Również teraz system w inny sposób numeruje elementy list zagnieżdżonych.

1. Pakiety do składu tekstu:

- (a) OpenOffice
- (b) Microsoft Office
- (c) L^AT_EX

2. Arkusze kalkulacyjne:

- (a) OpenOffice
- (b) Microsoft Office

I wreszcie listy z opisami, w których stosujemy otoczenie `description`, a ponadto przy każdym poleceniu `\item` wpisujemy w nawiasach kwadratowych wykorzystywany w tym elemencie opis:

Tekst źródłowy:

```
\begin{description}
  \item[pon] Poniedziałek
  \item[wt] Wtorek
  \item[reszta] \ldots
\end{description}
```

Efekt:

```
pon Poniedziałek
wt Wtorek
reszta ...
```

11.8 Tabele

Do składania tabel służy otoczenie `tabular`. Ma ono jeden argument, za pomocą którego opisuje się liczbę kolumn i sposób wyrównywania tekstu w kolumnach. Każdą kolumnę opisuje się za pomocą jednej litery, którą może być `l`, `c` lub `r`. I tak litera `l` oznacza kolumnę z wyrównaniem do lewej strony, litera `c` — kolumnę, w której materiał jest wyśrodkowany, a litera `r` — wyrównanie do strony prawej. Polecenie `\begin{tabular}{lcl}` rozpoczyna zatem tabelę z trzema kolumnami, z których środkowa zawiera materiał wycentrowany, a pozostałe są wyrównywane do strony lewej.

Wewnątrz otoczenia zapisuje się poszczególne wiersze tabeli, oddzielając je od siebie znakami `\\`. Każdy wiersz z kolei składa się z poszczególnych komórek oddzielonych od siebie znakami `&`. Wyjaśnijmy to na przykładzie:

Tekst źródłowy:

```
\begin{tabular}{lcl}
  styczeń & 31 & \\
  luty    & 28 & a czasem 29 \\
  marzec  & 31 & 
\end{tabular}
```

Efekt: [3mm]

styczeń	31	
luty	28	a czasem 29
marzec	31	

W powyższym przykładzie w pierwszym i trzecim wierszu ostatniej kolumny nie było żadnego napisu. Lepiej można to zaobserwować, jeśli doda się pionowe linie oddzielające kolumny. Uzyskuje się je wstawiając w opisie argumentu symbole `||`, w odpowiednie miejsca:

Tekst źródłowy:

```
\begin{tabular}{||lcl||}
  styczeń & 31 & \\
  luty    & 28 & a czasem 29 \\
  marzec  & 31 & 
\end{tabular}
```

Efekt: [3mm]

styczeń	31	
luty	28	a czasem 29
marzec	31	

Między wierszami można również umieścić poziome kreski za pomocą polecenia `\hline`:

Tekst źródłowy:

```

\begin{tabular}{||lc|l||} \hline
  styczeń & 31 & \\ \hline
  luty    & 28 & a czasem 29 \\ \hline
  marzec  & 31 & \\ \hline
\end{tabular}

```

Efekt: [3mm]

styczeń	31	
luty	28	a czasem 29
marzec	31	

Linie poziomą można wytyczyć nie tylko przez całą długość tabeli. Można spowodować, żeby rozciągała się między wskazanymi kolumnami. Zamiast `\hline` stosuje się wtedy polecenie `\cline`, którego argumentem jest zakres, w jakim ma się rozciągać linia:

Tekst źródłowy:

```

\begin{tabular}{||lc|l||} \hline
  styczeń & 31 & \\ \cline{1-2}
  luty    & 28 & a czasem 29 \\ \cline{1-2}
  marzec  & 31 & \\ \hline
\end{tabular}

```

Efekt: [3mm]

styczeń	31	
luty	28	a czasem 29
marzec	31	

Jeszcze bardziej złożone tabele można tworzyć scalając wybrane komórki. Służy do tego polecenie `\multicolumn` z trzema argumentami, z których pierwszy to liczba komórek do scalenia, drugi — sposób wyrównania tekstu w scalonej komórce, a trzeci — tekst zamieszczany w scalonej komórce. Spójrzmy na przykład:

Tekst źródłowy:

```

\begin{tabular}{|l|l|r|} \hline
  \multicolumn{2}{|c|}{OKRES} & STRATA \\ \hline
  styczeń & 2001 & 300.50 \\ \hline
  luty    & 2002 & 1254.00 \\ \hline
  marzec  & 2002 & 0.23 \\ \hline
\end{tabular}

```

Efekt: [3mm]

	OKRES	STRATA
styczeń	2001	300.50
luty	2002	1254.00
marzec	2002	0.23

Umiejętnie łącząc powyższe polecenia oraz zagnieżdżając otoczenia `tabular` w sobie można tworzyć nawet skomplikowane tabele.

11.9 Spis treści i odsyłacze

System \LaTeX pozwala na szybkie utworzenie spisu treści poprzez umieszczenie w dokumencie polecenia `\tableofcontents`. Tworzenie spisu treści przebiega dwuetapowo. Gdy w dokumencie źródłowym znajduje się polecenie `\tableofcontents` \LaTeX tworzy w trakcie kompilacji plik o tej samej nazwie co dokument i rozszerzeniu `toc` i zapisuje w nim informacje potrzebne do utworzenia spisu treści. Jednocześnie informacja z poprzedniej wersji pliku `toc` jest wykorzystywana do utworzenia spisu treści. Skutkiem takiego rozwiązania jest konieczność co najmniej dwukrotnego przekompilowania pliku po każdej zmianie struktury dokumentu.

Informacje do tworzenia spisu treści są tworzone na podstawie wszystkich poleceń podziału dokumentu na rozdziały: `\section`, `\subsection` itd. Domyślnie każdy tytuł rozdziału i podrozdziału wraz z numerem strony jest wstawiany do spisu treści.

Jeśli zachodzi potrzeba zróżnicowania tytułu rozdziału i odpowiadającego mu wpisu w spisie treści, to można wykorzystać opcjonalny argument poleceń podziału. Pisząc na przykład `\section[Wstęp]{Wstęp do zagadnień podziału dokumentu na strony}` powodujemy wstawienie do dokumentu rozdziału zatytułowanego „Wstęp do zagadnień podziału dokumentu na strony”. W spisie treści rozdział ten będzie jednak reprezentowany za pomocą krótkiego wpisu „Wstęp”.

- Spis treści jest tworzony automatycznie na podstawie informacji zebranych w czasie kompilacji dokumentu.
- Po każdej zmianie struktury są niezbędne co najmniej dwukrotna kompilacja.
- Wpisy są pobierane z poleceń podziału i domyślnie są nimi tytuły rozdziałów i podrozdziałów.
- Jest możliwe określenie własnego wpisu różnego od tytułu rozdziału.

Typowy wygląd spisu treści jest przedstawiony poniżej:

Spis treści	
1 Wstęp	4
1.1 Poprawność specyfikacji	4
1.2 Specyfikacje tropowe	6
1.3 Systemy wspomagające dowodzenie twierdzeń	8
1.4 Teza pracy	9
1.5 Struktura pracy	10
2 Metoda tropów	12
2.1 Metoda tropów w wersji TAM'97	12
2.1.1 Ogólne założenia metody	12
2.1.2 Przykładowa specyfikacja	16
2.1.3 Specyfikacje niedeterministyczne	19
2.2 Jądro metody tropów	21
2.2.1 Niedeterminizm	21
2.2.2 Zmienne wejściowe i zmienne wyjściowe	21
2.2.3 Żetonny stan	22
2.2.4 Wzajemna rekurencja	22
2.2.5 Parametryzacja modułów	23
2.2.6 Składnia	23
2.3 Statyczna poprawność typów specyfikacji tropowej	25
2.3.1 Metajęzyk	26
2.3.2 Przykładowe reguły analizy statycznej specyfikacji tropowych	28
2.4 Warunki wewnętrznej spójności specyfikacji	37
2.4.1 Przykładowe warunki wewnętrznej spójności	37

W podobny sposób można utworzyć listę wszystkich ilustracji i listę wszystkich tabel. Zamiast polecenia `\tableofcontents` używa się wówczas poleceń `\listoffigures` oraz `\listoftables`.

W długim dokumencie często występują *odsyłacze*, czyli odwołania do innych rozdziałów dokumentu, do równania, elementu wyliczenia, tabeli, ilustracji itp. W odsyłaczu pojawia się numer obiektu, do którego chcemy się odwołać. Nie jest jednak dobrą praktyką umieszczanie tego numeru w pliku źródłowym i zapisanie czegoś w rodzaju: „porównaj rysunek 12”. Zawsze można przecież dodać nowy rysunek lub usunąć któryś z już obecnych w dokumencie i numeracja ulegnie zmianie. To samo dotyczy odsyłaczy do rozdziałów i innych elementów dokumentu.

System \LaTeX oferuje mechanizm polegający na nazywaniu obiektów, do których chcemy się odwoływać i późniejszym wykorzystaniu tych nazw w odsyłaczach. Aby mechanizm ten należycie zadziałał są potrzebne dwa polecenia: jedno do nazywania obiektów, drugie do odwoływania się do nich.

Do nazywania obiektów używamy w pliku źródłowym polecenia `\label`. Argumentem tego polecenia jest nazwa, jaką chcemy nadać obiektowi. Natomiast to, którego obiektu dotyczy nazwa, zależy od kontekstu wystąpienia polecenia `\label`.

Jeśli polecenie `\label` występuje zwyczajnie w tekście, to identyfikator dotyczy bieżącego rozdziału, podrozdziału itp.

A oto przykład:

- polecenie `\label{nazwa}` nadaje nazwę `nazwa` rozdziałowi, podrozdziałowi, ..., w którym się znajduje
- polecenie `\label` nie ma wpływu na postać końcowego dokumentu

Tekst źródłowy:


```

\section{Wstęp}
\label{wstep}
\ldots
\subsection{Teza pracy}
\label{teza}
\ldots
\subsection{Struktura pracy}
\label{struktura}
\ldots

```

definiuje identyfikatory: `wstep` związany z rozdziałem 1
`teza` związany z podrozdziałem 1.1
`struktura` związany z podrozdziałem 1.2

Polecenie `\label` może jednak pojawić się także wewnątrz otoczenia, takiego jak na przykład `enumerate`. Wtedy nazywa element listy, w którym się znalazło. Podobnie jeśli `\label` zapiszemy w otoczeniu `figure`, `table`, `equation`, to wtedy będzie nadawało nazwę odpowiednio: rysunkowi, tabeli czy formule. A oto odpowiednie przykłady:

Element wyliczenia:

```

\begin{enumerate}
  \item Pierwszy element listy
  \item Drugi element listy \label{drugi}
\end{enumerate}

```

Równanie:

```

\noindent A oto poszukiwane rozwiązanie:
\begin{equation}
  x = 2^{\frac{1}{y}} \label{roz}
\end{equation}

```

Aby odwołać się do nazwanego uprzednio obiektu należy wykorzystać polecenie `\ref`, którego argumentem jest nazwa obiektu. Polecenie to generuje po prostu numer tego obiektu. Jeśli zamiast do obiektu chcemy się odwołać do numeru strony, to używamy polecenia `\pageref`, którego argumentem jest nazwa obiektu. Polecenie to generuje numer strony, na której znalazło się odpowiednie polecenie `\label`.

Tekst źródłowy:

```

W rozdziale~\ref{teza} na stronie~\pageref{teza}
przedstawiliśmy tezę pracy. Tezę tę wykazaliśmy
równaniem~\ref{roz}.

```

Efekt końcowy: [2cm]

W rozdziale 1.1 na stronie 2 przedstawiliśmy tezę pracy. Tezę tę wykazaliśmy równaniem 3.1.

11.10 Bibliografia

L^AT_EX umożliwia także stworzenie bibliografii. Można to zrobić na dwa sposoby: samodzielnie lub wykorzystując bibliograficzną bazę danych. Przedstawimy teraz tę pierwszą metodą. Druga, choć jest wygodniejsza, wykracza poza ramy tego kursu.

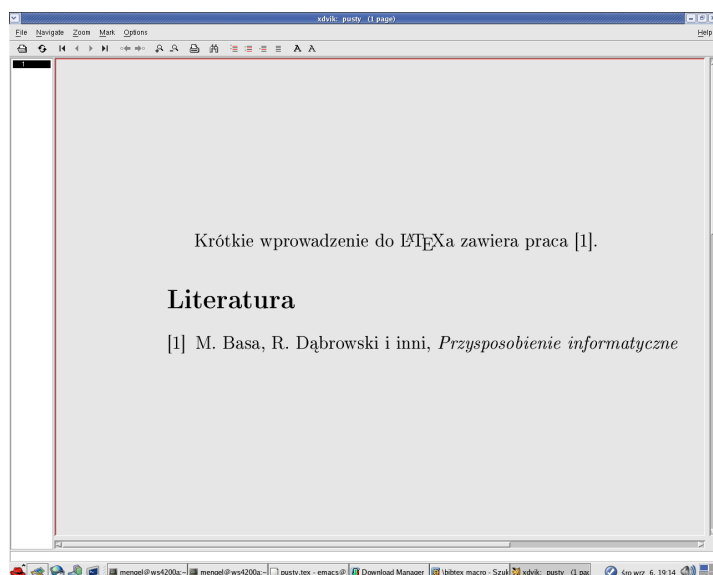
Aby umieścić w tekście odwołanie do pozycji ze spisu literatury (który zwykle zamieszcza się na końcu pracy) korzysta się z polecenia `\cite`. Jego argumentem jest nazwa (nadana przez użytkownika) cytowanej pozycji. Aby na przykład zamieścić odsyłacz do niniejszego podręcznika, można użyć polecenia `\cite{pi}`. Aby jednak taki odsyłacz zadziałał poprawnie, w tym miejscu dokumentu, w którym chcemy zamieścić spis literatury należy zamieścić otoczenie `thebibliography`. Otoczenie to jest bardzo podobne do otoczenia `enumerate`. Poszczególne pozycje na liście zaczynają się tym razem poleceniem `bibitem`, którego argumentem jest nazwa nadana danej pracy. Ponadto otoczenie `thebibliography` ma argument, który jest fragmentem tekstu o tej samej szerokości, co najszersza etykieta w spisie.

Oto przykład. Tekst źródłowy:

```
Krótkie wprowadzenie do \LaTeX a zawiera praca~\cite{pi}.
```

```
\begin{thebibliography}{9}
\bibitem{pi} M.~Basa, R.~Dąbrowski i inni,
  \emph{Przysposobienie informatyczne}
\end{thebibliography}
```

daje efekt:



11.11 Formuły matematyczne

Dużą siłą systemu \LaTeX jest łatwość składania formuł matematycznych. O dziwo, składanie formuł za pomocą poleceń okazuje się łatwiejsze i wygodniejsze niż posługiwanie się wyrafinowanymi edytorami formuł oferowanymi przez edytory typu WYSIWYG. Dowodzi tego fakt, że \LaTeX stał się czymś w rodzaju standardu w środowisku naukowym.

Formuły matematyczne możemy składać na dwa sposoby: w trybie wierszowym, gdy formuła jest umieszczona zwyczajnie w wierszu i otoczona tekstem tak jak: $x^{2y} > 7$, lub też w trybie wystawionym, w którym formuła jest składana w osobnym wierszu, jak tutaj

$$x^{2y} > 7.$$

Formułę składaną w trybie wierszowym ujmujemy w znaki $\$$, a w trybie wystawionym zapisujemy między znakami $\[$ a $\]$. Niezależnie od trybu, który używamy, składanie formuły przestawia system w tryb matematyczny, który w istotny sposób różni się od zwykłego trybu pracy. W trybie tym dostępne są specjalne polecenia, z których można korzystać *wyłącznie* w trybie matematycznym. Ponadto w trybie matematycznym \LaTeX ignoruje spacje w pliku źródłowym.

Składanie formuł w systemie \LaTeX wymaga pewnej wprawy i choć początkowo wydaje się być wyjątkowo trudne, to jednak po pewnym czasie staje się czynnością dość prostą. Dokładne omówienie poleceń dostępnych w trybie matematycznym wykracza poza ramy tego podręcznika, ale przyjrzyjmy się kilku najczęstszym konstrukcjom na wybranych przykładach.

Formuły składane w trybie wierszowym:

Plik źródłowy:	Efekt:
$\$x + y > 8\$$	$x + y > 8$
$\$x_1 / 2\$$	$x_1/2$
$\$x^2\$$	x^2
$\$x^{y+z}\$$	x^{y+z}
$\$x^y_2\$$	x_2^y
$\$x^{y^2}\$$	x^{y^2}

Formuły składane w trybie wystawionym zamiast znaków $\$$ są otoczone znakami $\[$ i $\]$, np.:

$\[x^{y^2}\]$ daje

$$x^{y^2}$$

Zatrzymajmy się na chwilę na przykładach indeksów górnych i dolnych. Przykład drugi i trzeci powyżej pokazuje, jak za pomocą $_$ i $^$ uzyskać indeks dolny i górny. W bardziej skomplikowanym przykładzie czwartym jako indeks górny występuje całe wyrażenie $y + z$, które jest ujęte w nawiasy $\{ i \}$. Pominięcie tych nawiasów spowodowałoby utworzenie formuły

$x^y + z$. Piąty przykład ilustruje jednocześnie użycie indeksu górnego i dolnego, a przykład szósty — wielokrotny indeks górny. Jak wynika z tych przykładów posługując się nawiasami wąsatymi możemy wymuszać traktowanie pewnych wyrażeń jako jednej całości i umieszczenie ich w całości w indeksie dolnym lub górnym.

Równie łatwo można składać ułamki, sumy, całki i inne nawet bardzo skomplikowane wzory matematyczne. Oto kolejne przykłady:

Plik źródłowy:	Efekt:
<code>\frac{x+y}{x^2 + \frac{1}{2}y^2}</code>	$\frac{x+y}{x^2 + \frac{1}{2}y^2}$
<code>x_1, \ldots, x_n</code>	x_1, \dots, x_n
<code>x_1 + \cdots + x_n</code>	$x_1 + \dots + x_n$
<code>\sum_{k=1}^{100} x_k</code>	$\sum_{k=1}^{100} x_k$
<code>\int_{-\infty}^0 f</code>	$\int_{-\infty}^0 f$

Ponieważ matematycy lubią używać wielu różnych dziwnych symboli, \LaTeX dysponuje wieloma z nich. Są dostępne na przykład litery greckie, duże litery kaligraficzne, różnego rodzaju symbole takie jak \leq , \geq , \cap , \cup , \vee , \wedge , strzałki różnego rodzaju: \rightarrow , \leftarrow , \leftrightarrow i setki innych symboli. Dodatkowo można przekreślać symbole, na przykład pisząc `$\$x \not<y\$$` co wyprodukuje napis $x \not< y$.

Skomplikowane wzory można składać korzystając ze specjalnych poleceń rozmieszczających elementy wzoru jako elementy specjalnej tabeli. Przykładem takiego polecenia jest *otoczenie* `array`. Za jego pomocą można składać skomplikowane wzory takie jak:

$$23 - \begin{array}{c} x_1 \\ \vdots \\ x_n \end{array} - \begin{pmatrix} x - y & 100 \\ x + y & 200 \end{pmatrix}$$

Otoczenie `array` rozmieszcza materiał w postaci tabeli. Liczbę kolumn tabeli oraz sposób równania tekstu w każdej z nich podaje się w jako argument tego otoczenia. Na przykład polecenie `\begin{array}{lcr}` rozpoczyna tabelę z trzema kolumnami (bo są trzy litery), z których pierwsza będzie równana do lewej strony, druga — środkowana, a trzecia wyrównana do strony prawej. Następnie wpisujemy elementy poszczególnych wierszy tabeli. Wiersze oddzielamy od siebie znakiem `\\`, a poszczególne komórki w wierszu znakiem `&`. Oto przykład:

- nagłówek, np. `\begin{array}{lcr}` rozpoczyna tabelę z trzema kolumnami: pierwsza wyrównana do lewej, druga do środka, trzecia do prawej
- pierwszy wiersz, np.: `12 & x - y & x^2 \\` składa się z trzech komórek
- drugi wiersz, np.: `13 & x + y & x^{y+x}`
- wierszy może być dowolnie dużo, po ostatnim piszemy `\end{array}`

Plik źródłowy:

```
\[\begin{array}{lcr}
  12 & & x - y & & x^2 \\
  13 & & x + y & & x^{y+x}
\end{array}\]
```

Efekt końcowy:

$$\begin{array}{lcr} 12 & x - y & x^2 \\ 13 & x + y & x^{y+x} \end{array}$$

Oczywiście takie tabele mogą być dowolnie zagnieżdżone.

Inne ciekawe polecenie pozwala umieszczać nawiasy klamrowe pod lub nad wyrażeniem:

Tekst źródłowy: $\$x \cdot y = \underbrace{x + \dots + x}_y\$$

Produkujecie napis:

$$x \cdot y = \underbrace{x + \dots + x}_y$$

Rozdział 12

Hipertekst i HTML

12.1 Wprowadzenie

Czego nie ma w internecie, tego nie ma na świecie — twierdzi jedno z porzekadeł początku XXI wieku. Nikogo chyba nie trzeba dziś przekonywać do tego, że możliwość upowszechniania treści za pomocą stron WWW jest, zwłaszcza w środowisku akademickim, sprawą kapitalnej wagi.

Stąd, jednym z obszerniejszych tematów Przysposobienia Informatycznego jest kurs tworzenia stron WWW z wykorzystaniem *explicite* języka znaczników HTML.

Dlaczego tak dziwnie, skoro większość edytorów tekstowych ma możliwość zapisu opracowywanego dokumentu do formatu HTML (który jest podstawowym formatem stron WWW)? Są na to cztery odpowiedzi:

- Edytory zapisują dokumenty do formatu HTML w nie najlepszym stylu, dając w efekcie strony „ciężkie”, trudne w manipulacji.
- Tylko pisząc samemu w HTML możemy tworzyć strony eleganckie, lekkie, łatwe w utrzymaniu i aktualizacji. Znajomość HTML jest także pierwszym krokiem w stronę tworzenia dynamicznych stron WWW, automatycznie uaktualniających swoją zawartość.
- Znajomość HTML może pomóc nam wygładzić efekt eksportu do HTML z ulubionego edytora;
- Pisanie w HTML to wyzwanie, podobne do logicznej łamigłówki!

Niewątpliwie, kiedyś sytuacja się zmieni i będziemy mogli tworzyć piękne dokumenty internetowe, korzystając ze zwyczajnego edytora, jakiego używamy na co dzień, lub za pomocą uniwersalnego środowiska składu tekstu w rodzaju L^AT_EX’a. Na razie jednak, jeśli chcemy pisać strony WWW, które będą eleganckie i prawidłowo wyświetlane niezależnie od tego, jakim sprzętem lub systemem operacyjnym dysponuje czytelnik — musimy opanować (nietrudną, ale wymagającą dokładności) sztukę zapisywania treści w języku HTML. Osoby, które zetknęły się z narzędziami do składu tekstu w

rodzaju L^AT_EX'a (**DO ZROBIENIA** → [link do kursu](#)) rozpoznają tu z pewnością wiele wspólnych idei i rozwiązań.

Cała filozofia stron WWW polegała — w swej oryginalnej postaci — na tym, by na pierwszym miejscu postawić *treść* dokumentu i jego *sens*, a dopiero dalej — jego formę. Między innymi, chodziło o to, by strony WWW były równie czytelne dla użytkowników najrozmaitszych komputerów i przeglądarek. Było to jednak wbrew powszechnemu trendowi w naszej kulturze, gdzie (niestety!) czołowe znaczenie dla dokumentu ma jego oprawa wizualna (łatwo to zaobserwować na przykładzie tzw. prasy kolorowej). Dlatego w krótkim czasie ten trend (niektórzy powiedzieliby: *tryd*) przeniósł się także na strony WWW...

Dlatego wiele podręczników tworzenia stron WWW ogranicza się do jednego celu: nauczyć, jak tworzyć pięknie wyglądające strony, bajecznie kolorowe, kipiące elementami graficznymi rozmieszczonymi w skomplikowany sposób, a najlepiej jeszcze — tańczące i śpiewające. Zapominają wszakże o jednym: jakiej *treści* ma służyć taka forma!

W niniejszym kursie wracamy do źródeł filozofii WWW. Zgodnie z najnowszymi standardami opracowanymi dla stron WWW, stawiamy (jak na intelektualistów przystało...) na treść i sens. Formę doceniamy, ale jako estetyczny dodatek, podkreślający znaczenie treści.

Wyjaśnijmy to na samym początku: naszym celem *nie jest* nauczyć się, jak *szybko* zapisywać swoje prace w pliku o formacie HTML. Naszym celem *nie jest* także nauczyć się, jak *małym nakładem pracy* stworzyć olśniewające serwisy WWW. Naszym celem *jest* nauczyć się, jak tworzyć poprawne i eleganckie dokumenty w formacie HTML, których pełna treść będzie wystarczającą wskazówką dla przeglądarki czytelnika, jak prawidłowo ją przedstawić. Nie oszukujmy się: jeśli chcemy robić to *dobrze*, nie będzie to ani *szybko*, ani *tanio*...

12.1.1 Cele zajęć

Niniejszy kurs pozwoli *aktywnie studiującej* osobie osiąść umiejętności pozwalające na

- Tworzenie dokumentów internetowych, w których warstwa znaczeniowa jest oddzielona od warstwy wizualnej
- Konstruowanie poprawnych składniowo, eleganckich stron WWW
- Uzupełnianie dokumentów HTML jednolitą, efektowną szatą graficzną
- Twórczą analizę innych stron WWW w celu przeniesienia nowych rozwiązań do swoich dokumentów
- Dalszą, samodzielną naukę bardziej zaawansowanych technik tworzenia stron WWW

Czy zauważyłeś, że ludzie najczęściej mówią o OGLĄDANIU stron internetowych, a nie o ich CZYTANIU?

12.1.2 Materiały dydaktyczne

Do zajęć przewidziane zostały następujące materiały dydaktyczne:

Podręcznik — niniejszy dokument w formacie elektronicznym PDF, zawierający materiał omawiany na zajęciach.

Materiał zawarty w podręczniku zawiera wszystkie informacje wymagane w ramach Przysposobienia Informatycznego, a w wielu miejscach wykracza poza ich zakres. Nie zakłada się, że student będzie korzystał z jakichkolwiek dodatkowych książek. Dla porządku jednak omówimy poniżej kilka dostępnych pozapodręcznikowych źródeł wiedzy.

Dostępne w księgarniach publikacje na temat tworzenia stron WWW, języka HTML i webmasterstwa można liczyć w dziesiątkach kilogramów. Niestety, większość z nich ma znacznie mniejszy ciężar gatunkowy.

W miarę dobrą książką jest dzieło Bartosza Danowskiego „*Tworzenie stron WWW w praktyce*”, wyd. Helion, 2003. Istotną zaletą tej pozycji jest oddzielenie — zgodnie z pierwotną intencją twórców HTML i metodą przyjętą w niniejszym kursie — struktury logicznej dokumentów (rozdziały, podrozdziały, wyróżnienia, itp.) od sposobu ich formatowania (rodzaj użytej czcionki, ramki lub ich brak, itp.). Jej wadą jest, typowy dla tego rodzaju wydawnictw, widoczny pośpiech redakcyjny, sztuczność w doborze przykładów i pewien chaos w prezentacji wiedzy.

Strony internetowe Potencjalnie dobrym źródłem wiedzy jest... analiza kodu *dobrych* stron WWW, a także jakichkolwiek innych, które nam się z jakiegoś powodu spodobały.

Każda przeglądarka stron WWW daje możliwość zobaczenia sposobu, w jaki została spreparowana oglądana przez nas strona. Najczęściej wystarczy polecenie w rodzaju

Widok ► Źródło strony.

Warto więc podejrzeć, jakie ktoś inny zastosował rozwiązania. Przy czym pamiętajmy: korzystajmy tylko z *dobrych* wzorców. Niestety, przeważająca część stron WWW jest źle (lub bardzo źle) skonstruowana!

12.1.3 Przeglądarka

Skoro zamierzamy tworzyć strony WWW, powinniśmy zadać sobie pytanie, dla jakiej przeglądarki internetowej (zob. rozdział 4) to robić. Zapewne wiele razy uważny Czytelnik zauważył na niektórych stronach internetowych groźnie brzmiące dopiski

Zaprojektowano dla Internet Explorera w rozdzielczości
800x600

albo

Best viewed with Netscape Navigator 4.0 or higher,

sugerujące, że strony WWW tworzy się specjalnie dla konkretnej przeglądarki — tak samo, jak dokumenty w formacie *.doc* tworzy się specjalnie dla aplikacji Microsoft Word.

Na szczęście, tak wcale nie jest! Dokumenty HTML (przynajmniej w teorii — i pod warunkiem, że zostaną zaprojektowane według właściwych zasad, o których piszemy w niniejszym podręczniku) powinny równie dobrze¹ wyświetlać się w praktycznie dowolnej przeglądarce wspierającej w wystarczającym zakresie obowiązujące *standardy*.

Właściwie każda z popularnych przeglądarek:

- Microsoft Internet Explorer,
- Mozilla/Firefox,
- Opera

jest dobra — najważniejsze, byśmy dysponowali możliwie *najnowszą* jej wersją. Omówieniem możliwości przeglądarek internetowych zajmuje się rozdział 4, my poniżej przypominamy kilka, z których będziemy tutaj korzystać.

Każda z tych przeglądarek² ma wersję działającą pod dowolnym z systemów: Microsoft Windows, Linux, Mac OS. Osobiście polecałbym na początek Mozillę, gdyż ma od razu wbudowany edytor kodu HTML i jest najmniej uzależniona od systemu. W dalszej części kursu będziemy dość często posługiwać się właśnie Mozillą lub jej młodszym — lecz uboższym przez np. brak edytora HTML — krewnym, Firefoxem.

Wszystkie liczące się przeglądarki HTML mają pod *prawym* klawiszem myszki kontekstowe menu, pozwalające podejrzeć oryginalny kod HTML oglądanej strony (w Mozilli jest to *Widok ► Źródło strony*). Z tej możliwości warto jak najczęściej korzystać podczas niniejszego kursu, aby zobaczyć, jak została zapisana strona, którą w danej chwili widzimy.

12.2 Hipertekst

Korzystając np. z tradycyjnej — drukowanej — wielotomowej encyklopedii, szybko dostrzegamy wady alfabetycznego rozmieszczenia haseł. Przykładowo, czytając jakieś hasło, w którym występują odsyłacze, np:

¹Co wcale nie znaczy, że tak samo!

²Oprócz Microsoft Internet Explorera, który nie ma wersji dla Linuxa.

KOMPUTER Urządzenie składające się z → monitora, → klawiatury, → myszy i pudła z → mikroprocesorem. Do działania potrzebuje → systemu operacyjnego.

wiemy, że odszukanie występujących w nim odsyłaczy, np.

SYSTEM OPERACYJNY Oprogramowanie niezbędne do działania → komputera. Przykładowe systemy operacyjne to

- Microsoft Windows dla komputerów PC,
- Mac OS dla komputerów Apple Macintosh,
- Unix.

Ten ostatni ma całkowicie darmową odmianę na komputery osobiste PC, Linux.

wiąże się z pewną niewygodą. Nie dość, że wymaga od nas uciążliwego wertowania kolejnych stron, to dodatkowo (siłą rzeczy) ogranicza się do wyjaśnień zamieszczonych wyłącznie w ramach tej, a nie innej encyklopedii. O ileż przyjemniej jest *kliknąć* na taki odsyłacz (np. „→ systemu operacyjnego” w przykładzie powyżej) i natychmiast zobaczyć, co się pod tym hasłem kryje!

Hipertekst ma właśnie za zadanie wprowadzenie powyższych udogodnień do klasycznych dokumentów tekstowych (w formie elektronicznej). Odsyłacze w hipertekście nazywa się popularnie z angielska *linkami*. Szybko zorientowano się, że zalety hipertekstu wykraczają poza pierwotne słownikowo-encyklopedyczne korzenie³, nie tylko zastępując standardowe odsyłacze, ramki, notatki na marginesie, referencje do bibliografii, ale także prowadząc do jakościowej zmiany w podejściu do dokumentu.

Wszyscy znamy takie zachowanie się tekstu elektronicznego z codziennych wizyt na stronach WWW. Połączenie hipertekstu i internetu dało bowiem dodatkową możliwość: tworzenia odsyłaczy prowadzących do *innych* dokumentów!

Treść naszego hasła SYSTEM OPERACYJNY może w ten sposób być o wiele bogatsza, gdyż odsyłacze w nim zamieszczone mogą wykraczać poza zawartość naszej encyklopedii: przykładowo, odsyłacz do Microsoft'u mógłby prowadzić wprost na stronę domową tej firmy!

³Jednymi z pierwszych dokumentów hipertekstowych były komputerowe systemy pomocy dla programów użytkowych.

12.2.1 Specyfika hipertekstu

Najważniejszą różnicą w stosunku do zwykłego tekstu jest to, że łamie on tradycyjną strukturę dokumentu.

Autor nie może już zakładać, że czytelnik

- będzie czytać dokument strona po stronie
- w ogóle kiedykolwiek przeczyta początek (np. wstęp) do dokumentu

Raczej należy przypuszczać, że czytelnik znajdzie informację, która go interesuje, być może jeszcze rozejrzy się tu i tam i... przeczuci się na lekturę innego dokumentu.

Więcej na ten temat można znaleźć w artykule B. Siemienieckiego „*Hipertekst jako narzędzie poznawcze*” (zob. <http://www.ped.uni.torun.pl/ztk/a1.htm>).

Tak, zgadza się: musimy liczyć się z powierzchownym odbiorem tekstu. Nie miejsce tutaj na ocenę tego, czy to dobrze, czy to źle. Jak często bywa, kluczowe jest zachowanie właściwych proporcji:

- Autor**
- Nie powinien popadać w przesadę, tworząc dokumenty hipertekstowe zbyt poszatkowane na odseparowane części, ryzykując brak syntetycznego spojrzenia.
 - Powinien, przez odpowiednie olinkowanie i dobrze rozplanowane menu nawigacyjne ułatwić czytelnikowi orientację w dokumencie i wskazać „pożądaną” ścieżkę czytania.

- Czytelnik**
- Nie powinien ulegać złudzeniu, że esencjonalne hasła są równie cenne jak ta sama informacja zanurzona w szerszym kontekście.
 - Powinien starać się podążać za tokiem myślenia autora, umieścić uzyskiwane informacje w szerszym, ale wewnętrznie spójnym kontekście. Powinien skupić się na głównej myśli, a nie tylko uciekać w hiperlinkowe dygresje (skądinąd zazwyczaj nad wyraz ciekawe).

Upowszechniły się dwa standardy (mówimy w żargonie komputerowców o ”formatach”) dokumentów hipertekstowych:

HTML

HTML (ang. *HyperText Markup Language*) — jest to standard dokumentów przeznaczonych do publikacji na stronach WWW. To właśnie dokumenty formatu HTML odczytujemy wprost w popularnych przeglądarkach internetowych, takich jak: Internet Explorer (zob. <http://www.microsoft>).

com/windows/ie), Opera (zob. <http://www.opera.com>), Mozilla (zob. <http://www.mozilla.org>), itp.

Dokument w formacie HTML to w zasadzie zwykły dokument tekstowy uzupełniony znacznikami (ang. *Markup* — stąd nazwa formatu!). To jest wielka zaleta, gdyż takie dokumenty są w ten sposób mało zależne od konkretnej maszyny, a przeglądarka, gdy „nie radzi” sobie z jakimś znacznikiem, zawsze może wyświetlić po prostu zwykły tekst.

Format HTML rozszerza możliwości hipertekstu o multimedia oraz, co bardzo ważne, o możliwość wpływania przez czytelnika (sic!) na ostateczną formę dokumentu. Na przykład, osoby słabo widzące, mogą powiększyć (wystarczy wybrać *Widok ► Rozmiar tekstu*) czcionkę, a nawet zmienić ją na inną, przy czym rozmieszczenie tekstu w okienku dopasuje się automatycznie do nowej sytuacji. Osoby niewidome mogą skorzystać z automatycznego czytania tekstu z wykorzystaniem syntezy mowy. Autor dokumentu HTML w praktyce nie może przewidzieć, w jaki sposób zostanie wyświetlony jego tekst! (Zob. przykłady na str. 296.)

Format HTML jest bardzo dogodny do prezentacji informacji *on-line*, w sieci WWW: właściwie można by rzec, że WWW została stworzona dla HTML (i na odwrót). Strony internetowe w HTML mogą być bardzo atrakcyjne, umożliwiać interaktywne przeglądanie. W naturalny sposób daje się w nich wyszukiwać informacje, zarówno z poziomu przeglądarki (w konkretnym pliku) jak z poziomu całej sieci WWW (korzystając z wyszukiwarek internetowych).

Przyjęło się, że plikom w formacie HTML nadaje się rozszerzenie *.htm* lub *.html*

PDF

PDF (ang. *Portable Document Format*) jest o wiele bardziej niż HTML zbliżony do formy „drukowanej”, zafiksowanej przez autora: tej jedynej i właściwej. Podobnie jak HTML, dokumenty PDF mają możliwości hipertekstowe (klikanie w linki w dokumencie PDF powoduje przeskok do odpowiedniego fragmentu tekstu lub nawet innej strony WWW). Ustalona forma dokumentu i przenośność formatu powoduje, że bez większej przesady można twierdzić, iż każdy, z każdego komputera, będzie mógł go wydrukować w postaci praktycznie identycznej z oryginałem. Dlatego PDF jest bardzo dogodny dla większych materiałów dydaktycznych (skrypty, książki, zestawy zadań). W plikach PDF, podobnie jak w HTML, także jest możliwe wyszukiwanie wyrazów lub fraz, ale tylko w obrębie danego dokumentu. Dobre wyszukiwarki internetowe, np. Google, potrafią zindeksować, oprócz, co oczywiste, dokumentów HTML, także dokumenty w formacie PDF.

Będziemy jeszcze o tym mówić, ale warto już teraz zaznaczyć, że o ile ładne dokumenty w formacie PDF możemy wyprodukować po prostu eksportując pliki z ulubionego (dostatecznie nowoczesnego) edytora tekstu do formatu PDF, o tyle możliwości HTML wykorzystamy najpełniej, tworząc dokumenty w tym formacie niejako „ręcznie”.

W dokumentach HTML mogą także być zanurzone pewne zewnętrzne obiekty: grafika, klipy audio/wideo, itp.

Ma to istotne konsekwencje dla sposobu tworzenia takich dokumentów

Przykład 12.2.1. Elektroniczna wersja naszego podręcznika jest dokumentem PDF! W końcu, niniejszy podręcznik w wersji elektronicznej jest właśnie dokumentem w formacie PDF! Warto zobaczyć, jak są wykorzystane hiperlinki w spisie treści: klikając w nazwę rozdziału, przekakujemy od razu do właściwej strony. Zwróćmy także uwagę na hipertekstowe odsyłacze do innych dokumentów w sieci WWW!

W PDF można także tworzyć samodzielne, *wypełnialne elektronicznie* formularze.

12.2.2 Porównanie wybranych charakterystyk formatów HTML i PDF

Są dwa praktyczne formaty dla dokumentów hipertekstowych, które mogą być odczytane praktycznie na dowolnym komputerze, z dowolnym systemem operacyjnym: HTML i PDF.

Jednym zdaniem: preferowany format dla dokumentów WWW to HTML. PDF możemy wybrać dla dłuższych dokumentów (np. skrypty) oraz w przypadku, gdy zależy nam na przenośności nie tylko treści, ale też formy dokumentu.

Z tej dwójki, tylko HTML ma prawdziwe możliwości multimedialne i tylko HTML daje czytelnikowi możliwość wpływu na sposób prezentacji treści. Z drugiej zaś strony, tylko PDF daje sztywną formę; w najnowszych wersjach PDF daje także możliwość kontroli nad dostępem do samego dokumentu (np. drukowanie wyłącznie po podaniu hasła).

Pamiętajmy: w przeciwieństwie do PDF, ostateczny wygląd strony WWW zapisanej w HTML zależy od ustawień i możliwości konkretnej przeglądarki.

W ramach uzupełnień warto przeczytać tym, jakie są możliwości PDF w zakresie prezentacji treści (zob. http://www.alistapart.com/articles/pdf_accessibility) — na tle innych rozwiązań, takich jak HTML.

12.2.3 Przeglądarki dokumentów hipertekstowych

(DO ZROBIENIA → Czy usunąć cały ten rozdział)

Przeglądarki HTML

Działają według zasady: pokazać jak najlepiej i jak najwięcej. Jeśli na przykład w dokumencie znajduje się jakiś element, którego nie potrafią wyświetlić, po prostu pomijają go i wyświetlają resztę dokumentu!⁴ Zatem, jeśli na naszej stronie są jakieś elementy (np. aplety w Javie), których nie ob-

⁴To bardzo wielka zaleta, wystarczy przypomnieć sobie, co działo się, gdy do starej wersji popularnego edytora tekstu chcieliśmy wczytać dokument stworzony w innej wersji tego programu: program po prostu odmawiał wyświetlenia go, pisząc coś w stylu „*Upgrade to the most recent version!*”!

sługuje nasza przeglądarka, wciąż będziemy mogli zobaczyć resztę treści dokumentu HTML. Dobre i popularne przeglądarki⁵ to:

- Mozilla i Firefox (zob. <http://www.mozilla.org>)
- Opera (zob. <http://www.opera.com>)
- Microsoft Internet Explorer (zob. <http://www.microsoft.com/windows/ie>)

Przeglądarki PDF

Mają na celu wyświetlenie dokumentu w postaci jak najwierniej odpowiadającej oryginałowi, dlatego PDF doskonale nadaje się jako (niezależny od komputera!) format dla prezentacji, dużych publikacji z możliwością wydruku, itp. Popularne, dobre, darmowe i dostępne na wszystkie sensowne systemy przeglądarki PDF to:

- Acrobat Reader (zob. <http://www.adobe.com/products/acrobat/readstep2.html>)
- Ghostscript (zob. <http://www.cs.wisc.edu/~ghost>)

Niektóre dokumenty PDF (te, które nie mają osadzonych w dokumencie czcionek) mogą mimo wszystko źle się wyświetlać na komputerach, które używanych w dokumencie czcionek nie posiadają. Dlatego na wszelki wypadek lepiej jest w dokumentach PDF używać najpopularniejszych: Timesa lub Verdany.

12.3 Pierwsze strony WWW

Czym naprawdę są pliki HTML, najlepiej zobaczyć na przykładzie. Oto fragment pewnego pliku w formacie HTML; dla większej czytelności, znaczniki HTML wyróżniliśmy (tutaj i w dalszej części podręcznika) grubszą czcionką:

```
<p>Jestem wielkim zwolennikiem wakacyjnej włóczęgi, bez z góry  
określonego celu i <strong>bez</strong> wielkiego budżetu.  
Stosuję zawsze zasadę <q>wolniej jedziesz – więcej zobaczysz</q>.  
W ten sposób, w ciągu ostatnich pięciu lat udało mi się  
bez większych problemów poznać takie <em>niebanalne</em> okolice, jak  
<ul>  
<li> Czersk,  
<li> Zalesie Górne i Dolne,
```

Oglądając stronę WWW, możesz zawsze podejrzeć jej kod HTML! W większości przeglądarek wystarczy kliknąć prawym klawiszem myszy w pustym miejscu na stronie i wybrać *Pokaż źródło strony*.

⁵Każdą z nich można za darmo ściągnąć przez Internet w najnowszej wersji i zainstalować na swoim komputerze z dowolnym z systemów: Microsoft Windows, Linux (z wyjątkiem Microsoft Internet Explorera), Mac OS.

```
<li> Targówek.  
</ul>
```

Jak widzimy z powyższego przykładu, dokumenty w formacie HTML to zwyczajne, niesformatowane pliki tekstowe, których treść została uzupełniona o znaczniki⁶ w rodzaju `<p>` czy ``. Zadaniem znaczników jest poinformowanie przeglądarki (a za jej pośrednictwem — czytelnika) o tym, jakie znaczenie mają fragmenty tekstu do których się odnoszą. Przeglądarka interpretuje te znaczniki — a następnie, w zależności od swoich ustawień, wyświetla na ekranie (bądź prezentuje w inny sposób, np. odczytuje na głos). Dobra przeglądarka mogłaby powyższy tekst HTML wyświetlić na ekranie w następującej formie:

Jestem wielkim zwolennikiem wakacyjnej włości, bez z góry określonego celu i **bez** wielkiego budżetu. Stosuję zawsze zasadę „*wolniej jedziesz — więcej zobaczysz*”. W ten sposób, w ciągu ostatnich pięciu lat udało mi się bez większych problemów poznać takie *niebanalne* okolice, jak

- Czernsk,
- Zalesie Górne i Dolne,
- Targówek.

12.3.1 Znaczenie znaczników

Zadaniem znaczników HTML jest poinstruowanie przeglądarki HTML, jakie *znaczenie* ma tekst, które jest przez nie otoczony (stąd nazwa: znaczniki!). Dzięki temu przeglądarka będzie wiedzieć m.in. *w jaki sposób* ma wyświetlić tę treść.

Powiedzmy to wyraźnie: w swej czystej formie, znaczniki HTML *nie służą formatowaniu tekstu!*. Dokumenty HTML wyraźnie oddzielają od siebie warstwę znaczeniową tekstu od jego sformatowania. Wiąże się to z tym, że HTML w swoim założeniu ma na celu maksymalne uniezależnienie prezentacji treści od wybranego medium, przy jednoczesnej pełnej akceptacji faktu, że sposób i forma tej prezentacji nie są z góry określone.

Z bardzo podobną sytuacją mieliśmy do czynienia w systemie komputerowego składu tekstu \LaTeX . Rzeczywiście, tam również przede wszystkim podawaliśmy informację strukturalną dotyczącą tekstu (np. `\section{Znaczenie znaczników}`),

Znaczniki HTML spełniają analogiczną rolę jak instrukcje \LaTeX 'a: definiują specjalne znaczenie tekstu, którego dotyczą — a w zależności od tego znaczenia, tekst jest odpowiednio formatowany.

⁶Uwaga edytorska: w niniejszym podręczniku, dla większej czytelności tekstu, znaczniki HTML wyróżniamy pogrubioną czcionką (podobnie zachowują się niektóre edytory HTML, zob. rozdział 12.3.3. W rzeczywistości, w prawdziwym pliku HTML żadnych wytłuszczeń fizycznie nie ma!

a sposób sformatowania takiego tekstu zależał od *stylu* dokumentu, zdefiniowanego gdzieś na zewnątrz.

Przyjrzyj się poniższym przykładom. Zwróć uwagę, że wszystkie znaczniki w przykładach (i to dotyczy znakomitej większości⁷ znaczników HTML) występują w parach: znacznik otwierający (np. ``, będący oznaczeniem początku wypunktowania w przykładzie powyżej) i znacznik zamykający (np. ``, wskazujący koniec wypunktowania). W ten sposób przeglądarka „wie”, do jakiego fragmentu tekstu odnosi się dany znacznik!

Przykład 12.3.1. Wyróżnienie tekstu

To drzewo <code>naprawdę ma 100 lat!</code>	To drzewo <i>naprawdę</i> ma 100 lat!
---	--

Użyty znacznik ``, czyli wyróżnienia (ang. *emphasis*), nadaje słowu „naprawdę” dodatkowe (niezwerbalizowane w samym tekście!) znaczenie: na słowo „*naprawdę*” należy położyć nacisk. W drukowanym tekście zazwyczaj stosuje się w tym celu kursywę (tak, jak w przykładzie powyżej), ale np. urządzenie odczytujące ten tekst z pomocą syntezy mowy⁸ mogłoby przeczytać ten wyraz ze stosowną intonacją, dobitniej, właśnie po to, by wyróżnić je w odczytywanym tekście.

Przykład 12.3.2. Silne wyróżnienie

To drzewo ma <code>100 lat!</code>	To drzewo ma 100 lat!
---	------------------------------

Tu raczej chodziłoby nam o podkreślenie faktu, że to **aż sto** lat. Dla takiego silnego wyróżnienia, mamy w HTML znacznik ``: mocno podkreśla wagę zaznaczonego tekstu. I znów, w praktyce drukarskiej zazwyczaj w takich sytuacjach stosujemy pogrubiony tekst; odczytując ten tekst przy użyciu innego medium, moglibyśmy adekwatnie go zinterpretować.

Przykład 12.3.3. Cytat

Jest takie porzekadło: <code><q>Wysoki jak brzoza, a głupi jak koza</q>...</code>	Jest takie porzekadło: <i>“Wysoki jak brzoza, a głupi jak koza”...</i>
--	---

⁷Jest kilka wyjątków: np. ``, `
`, `<hr>` — są to znaczniki, które nie mają znacznika zamykającego. Ponadto, dla wygody autora stron HTML, niektóre znaczniki nie muszą mieć znacznika zamykającego, np. `<p>` lub ``, gdyż z kontekstu jasno wynika, gdzie się kończą.

⁸Tak, są takie urządzenia, np. dla niewidomych

Tym razem znacznik `<q>` oznacza krótki cytat (ang. *quotation*), wpleciony w tekst. Zauważmy, że rola `<q>` w HTML jest identyczna jak rola znaku cudzysłowu w zwykłym tekście drukowanym. Dlatego, jak widzimy po prawej stronie, dobra przeglądarka zinterpretuje ten znacznik otaczając tekst cudzysłowami (i, najczęściej, dodatkowo wyróżni tekst cytatu kursywą). Używając dobrego programu odczytującego na głos ten tekst zapisany w formacie HTML, zapewne usłyszelibyśmy coś w stylu:

Jest takie porzekadło: *(cytuje)* Wysoki jak brzoza, a głupi jak koza *(koniec cytatu)*...

Dowiedzieliśmy się więc, jakie znaczenie mają przynajmniej niektóre ze znaczników, które widzieliśmy w przykładowym tekście w formacie HTML. Pozostałe znaczniki mają charakter mniej znaczeniowy, a bardziej strukturalny.

Przykład 12.3.4. Tytuł

<code><h1>Moje podróże</h1></code>	Moje podróże
--	--------------

Pamiętaj, `<h1>` *nie jest* sposobem powiększania czcionki w tekście. Jest to oznaczenie *tytułu*: tylko tyle i aż tyle.

Tekst otoczony znacznikami `<h1>` i `</h1>` jest interpretowany jako nagłówek dla następującej po nim partii tekstu, czyli, trzymając się terminologii książkowej — jako tytuł rozdziału. Zwykle do wyświetlenia takiego znacznika przeglądarki dobierają dużą czcionkę, pogrubioną. Podobnie jak w książce, gdzie prócz rozdziałów znajdują się także podrozdziały, a w nich pod-podrozdziały, itp., w HTML mamy, oprócz znaczników `<h1>`, znaczniki niższego poziomu, tzn. `<h2>`, `<h3>`, ..., aż do `<h6>`!

Przykład 12.3.5. Akapit

Jedną z najmniejszych części struktury tekstu jest akapit. W HTML nawet każdy akapit musi być oznaczony znacznikiem: `<p>` (ang. *paragraph*). Ponieważ występuje bardzo często w plikach HTML, standard dopuszcza, by nie używać znacznika kończącego `</p>`: przyjmuje się domyślnie, że akapit kończy się tam, gdzie rozpoczyna się następny znacznik określający blokowy element tekstu: nowy akapit, nagłówek, lista, tabela, itp.

<p>Większość potraw z ryb może być przyrządzona niewielkim nakładem sił i środków. Nawet skromny budżet nie będzie przeszkodą, jeśli wybierzemy jedną z tańszych ryb, w rodzaju morszczuka czy mintaja. **<p>**Nie polecamy kupowania ryb mrożonych, które zazwyczaj pokryte są wyraźną warstwą lodu, za który niepotrzebnie płacimy wygórowaną cenę.

Większość potraw z ryb może być przyrządzona niewielkim nakładem sił i środków. Nawet skromny budżet nie będzie przeszkodą, jeśli wybierzemy jedną z tańszych ryb, w rodzaju morszczuka czy mintaja.

Nie polecamy kupowania ryb mrożonych, które zazwyczaj pokryte są wyraźną warstwą lodu, za który niepotrzebnie płacimy wygórowaną cenę.

12.3.2 Najprostszy kompletny dokument w formacie HTML

Zorientowaliśmy się, że pliki HTML są zwykłymi plikami tekstowymi, uzupełnionymi znacznikami — widzieliśmy jednak jedynie fragment takiego pliku. Każdy kompletny dokument HTML musi zawierać pewien zestaw znaczników, strukturalizujących jego zawartość. Czas teraz omówić taki kompletny dokument — najprostszą stronę WWW zapisaną w języku HTML.

```
<html>
  <head>
    <title>Strona internetowa Jana Kowalskiego</title>
  </head>

  <body>
    <p>
      Witajcie na mojej stronie internetowej.
    </p>
  </body>
</html>
```

Każda strona HTML rozpoczyna się zawsze znacznikiem **<html>**, a kończy się znacznikiem **</html>**.

Na zawartość strony HTML składają się dwie części: nagłówek i właściwa treść.

s

1. Nagłówek

Pierwszą częścią strony HTML jest część nagłówkowa (ang. *header*), zawierający dodatkowe informacje o stronie, czasami niezbędne dla prawidłowego wyświetlenia jej przez przeglądarkę

```
<head>
  ...zawartość nagłówka...
</head>
```

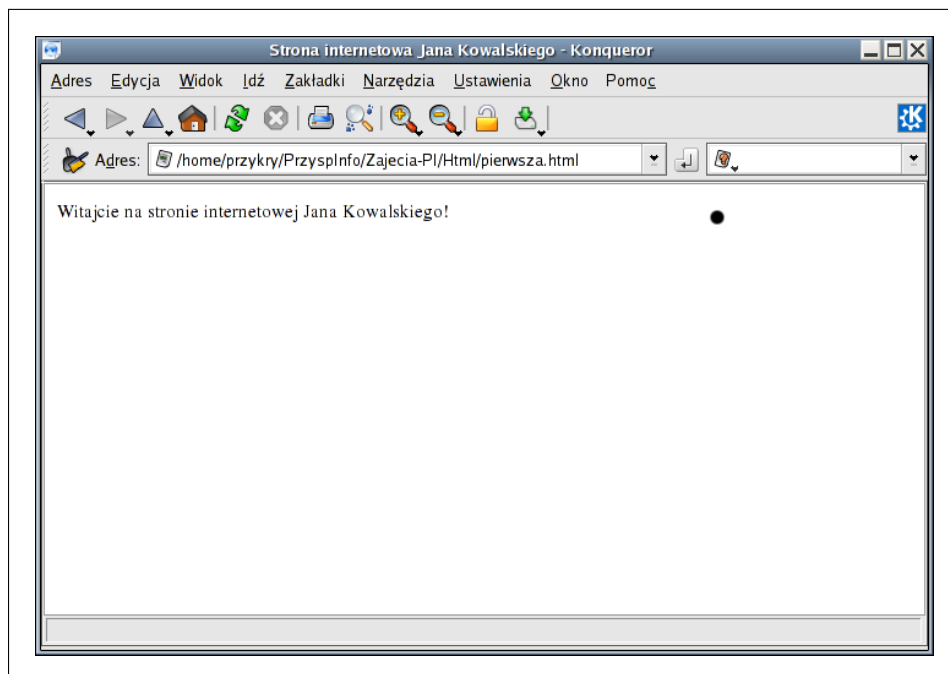
Większość danych zawartych w nagłówku strony *nie jest pokazywana* użytkownikowi przez przeglądarkę.

2. Właściwa treść strony

Drugą, z punktu widzenia czytelnika najważniejszą częścią strony HTML jest jej właściwa treść (ang. *body*).

```
<body>
  ...treść strony...
</body>
```

Właśnie ta druga część dokumentu HTML jest prezentowana czytelnikowi przez przeglądarkę!

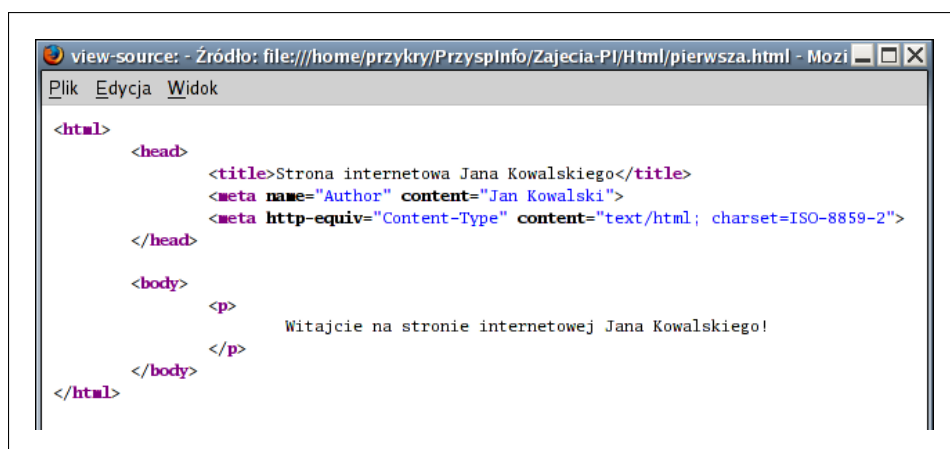


Widok omawianej strony w przeglądarce

W naszym przykładowym dokumencie HTML, zarówno nagłówek, jak i treść są bardzo skromne:

nagłówek (`<head>`) — jedynym jego elementem na naszej stronie jest znacznik `<title>`, czyli tytuł dla tego dokumentu, u nas: „Strona internetowa Jana Kowalskiego”. Zwyczajowo, przeglądarki wyświetlają zawartość znacznika `<title>` na pasku tytułowym okna przeglądarki, jak na obrazku powyżej. Jak wkrótce zobaczymy, typowy nagłówek prawdziwej strony WWW będzie musiał być trochę bardziej rozbudowany.

treść (`<body>`) — po prostu jeden akapit, „Witajcie na mojej stronie internetowej.”



```
view-source: - Źródło: file:///home/przykry/PrzyspInfo/Zajecia-PI/Html/pierwsza.html - Mozi
Plik Edycja Widok
<html>
  <head>
    <title>Strona internetowa Jana Kowalskiego</title>
    <meta name="Author" content="Jan Kowalski">
    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-2">
  </head>
  <body>
    <p>
      Witajcie na stronie internetowej Jana Kowalskiego!
    </p>
  </body>
</html>
```

Widok oryginalnej zawartości pliku HTML omawianej stron w przeglądarce (Firefox: Widok ► Źródło strony)

Tajemnice nagłówka `<head>`

W nagłówku dokumentu HTML powinniśmy zamieścić następujące informacje:

- Bardzo istotną dla przeglądarki informację⁹, o użytym kodowaniu dla polskich znaków, na przykład

```
<meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-2" >
```

Dlaczego potrzebna jest informacja o kodowaniu znaków narodowych — wyjaśniamy w następnym rozdziale.

- Tytuł strony (krótki, ale treściwy), np. gdyby nasz dokument był przeglądem gatunków warzyw uprawianych w Europie, to napisalibyśmy

```
<title>Warzywa krajów europejskich</title>
```

Przeglądarki zazwyczaj wyświetlają zawartość tego znacznika na pasku tytułowym okna.

- Informację o autorze, np. gdyby autorem był dr Sebastian Por, to zapisalibyśmy w nagłówku

```
<meta name="Author" content="Sebastian.Por" >
```

Jest to informacja czysto formalna, zazwyczaj przeglądarki nie mają z niej żadnego pożytku, natomiast mogłyby ją wykorzystać np.

⁹Zwróćmy uwagę na to, że znacznik `<meta>` jest kolejnym znacznikiem, który nie ma znacznika zamykającego!

programy indeksujące strony WWW na użytek wyszukiwarek internetowych.

Tak więc, przykładowy nagłówek w dokumencie HTML może ostatecznie wyglądać następująco

```
<html>
  <head>
    <meta http-equiv="Content-Type"
      content="text/html; charset=ISO-8859-2" >
    <title>Warzywa krajów europejskich</title>
    <meta name="Author" content="Sebastian Por" >
  </head>
  <body>
    ...właściwa treść dokumentu...
  </body>
</html>
```

Jak pisać po polsku?

Aby tworzyć strony WWW w naszym języku ojczystym w taki sposób, by dobrze się wyświetlały we wszystkich nowoczesnych przeglądarkach, musimy trochę więcej się napracować. Sprawcami zamieszania są tzw. polskie litery:

ż ó ł ć ą ę ...

Niestety, z powodów historyczno-marketingowych, litery polskiego alfabetu kodowane są obecnie na trzy (niezgodne ze sobą) sposoby:

ISO-8859-2 Międzynarodowy standard kodowania znaków narodowych (wymieniony dotyczy krajów Europy Środkowej), ustanowiony przez International Standarization Organization (ISO). Używany w starszych wersjach systemu Linux, obecnie następuje dość wolna zmiana tego standardu na lepszy — UTF-8. Osoby tworzące dokumenty HTML w Linuxie, bądź korzystając z dobrych narzędzi wspomagających tworzenie stron WWW, np. Kompozytora Stron w przeglądarce Mozilla, powinny wybrać właśnie to kodowanie i wpisać w nagłówku dokumentu HTML

```
<meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-2" >
```

Windows-1250 Niezgodny z ISO-8859-2 standard, narzucony w wielu produktach Microsoftu. Używany domyślnie w systemach typu Microsoft Windows w polskiej wersji językowej, dlatego osoby tworzące swoje dokumenty w tym środowisku korzystając np. z edytora Notepad powinny wybrać to kodowanie i wpisać w nagłówku dokumentu HTML

```
<meta http-equiv="Content-Type"
content="text/html; charset=Windows-1250">
```

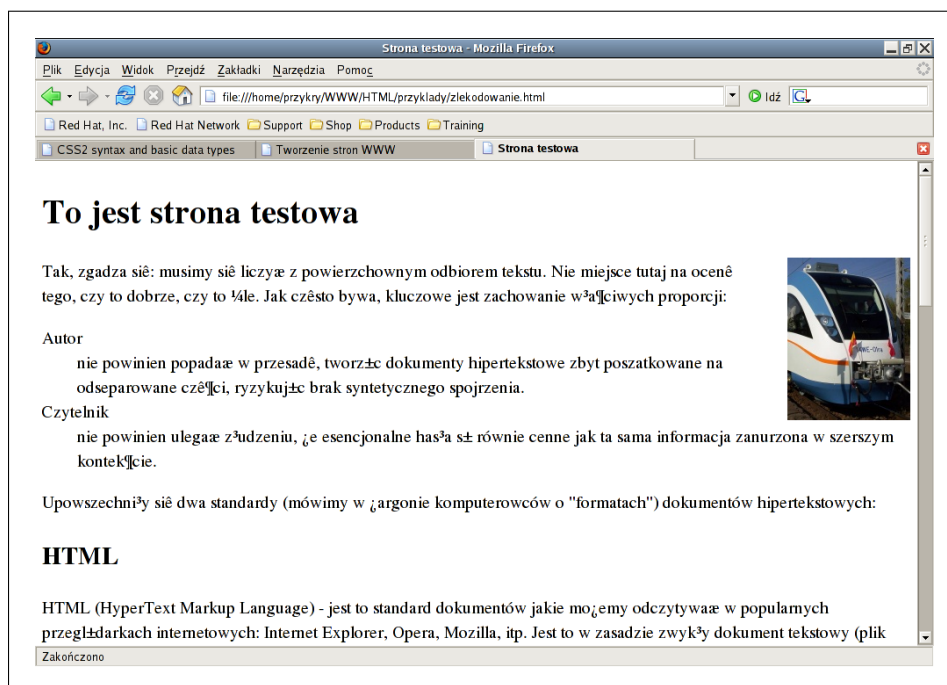
UTF-8 Nowy standard kodowania dokumentów wielojęzycznych, zapisanych przy użyciu zestawu znaków Unicode, bez niedogodności poprzednich, ale oczywiście niezgodny z nimi. System Linux używa obecnie w swych podstawowych aplikacjach właśnie UTF-8. Część aplikacji najnowszych wersji Windows także daje możliwość korzystania z UTF-8. W takim wypadku w nagłówku umieścimy linię

```
<meta http-equiv="Content-Type"
content="text/html; charset=UTF-8">
```

Wszystkie porządne przeglądarki obsługują wszystkie w/w sposoby kodowania, pod jednym wszelako warunkiem: że w nagłówku każdej publikowanej przez nas strony umieścimy znacznik informujący, jakie konkretnie kodowanie zostało użyte. Jeśli tego nie uczynimy, lub co gorsza, podamy informację niezgodną z rzeczywistością, efekty mogą być opłakane, co ilustruje poniższy przykład.

Przykład 12.3.6. Zła informacja o użytym kodowaniu polskich znaków

Oto co zobaczylibyśmy oglądając stronę zakodowaną faktycznie w ISO-8859-2, gdyby przeglądarka nie została o tym powiadomiona i domyślnie używała kodowania ISO-8859-1:



Polskie litery mogą wyświetlać się nieprawidłowo, gdy przeglądarka nie wie, w jakim standardzie są zakodowane

Należy mieć nadzieję, i wiele wskazuje na to, że opisany powyżej problem zaniknie za kilka lat, z chwilą, gdy wszystkie popularne systemy i aplikacje przejdą ostatecznie na kodowanie UTF-8. Ten proces już trwa.

12.3.3 Narzędzia tworzenia dokumentów w formacie HTML

Reguły tworzenia HTML są precyzyjnie określone. W naszym kursie uczymy się standardu HTML 4.01 (zob. <http://www.w3.org/TR/html4>). Standard ten został ustanowiony przez konsorcjum W3C (zob. <http://w3c.org>), które powołano właśnie w celu opracowywania najróżniejszych standardów obowiązujących w sieci WWW. Ich opis jest ostateczną instancją rozstrzygającą wątpliwości.

Edytor plików HTML

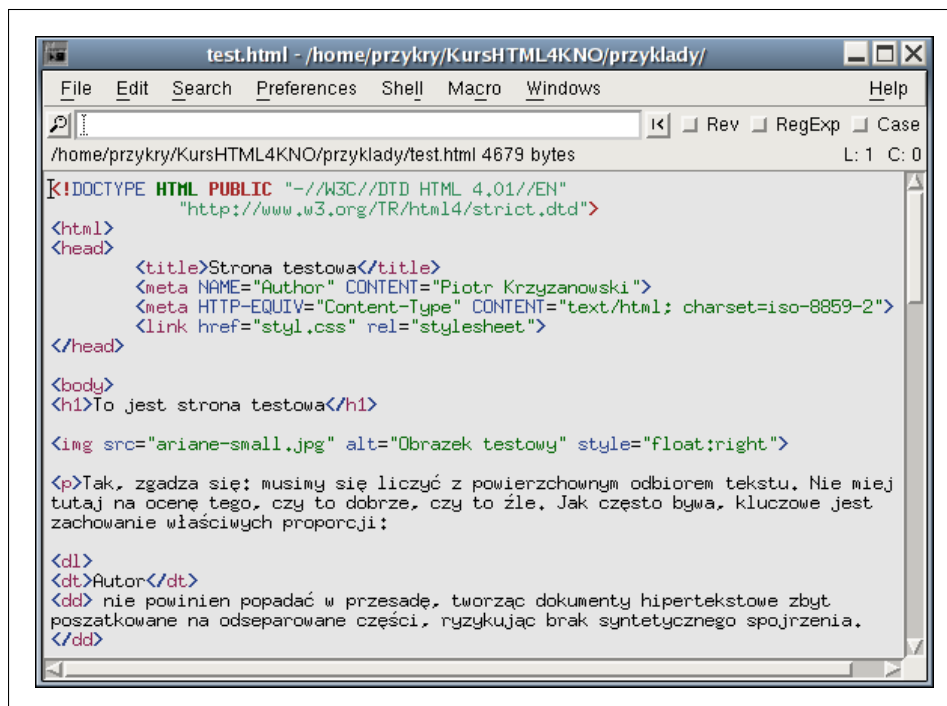
Ponieważ pliki HTML są *zwykłymi plikami tekstowymi*, do ich tworzenia wystarcza najprostszy z możliwych edytor (zob. rozdział 3), np:

Windows

- Notatnik
- WordPad

Linux

- pico
- nedit

A screenshot of a text editor window titled "test.html - /home/przykry/KursHTML4KNO/przyklady/". The window has a menu bar with "File", "Edit", "Search", "Preferences", "Shell", "Macro", "Windows", and "Help". Below the menu bar is a toolbar with icons for "Home", "Back", "Forward", "Rev", "RegExp", and "Case". The address bar shows the file path "/home/przykry/KursHTML4KNO/przyklady/test.html" and the file size "4679 bytes". The main text area contains the following HTML code:

```
!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
  <title>Strona testowa</title>
  <meta NAME="Author" CONTENT="Piotr Krzyzanowski">
  <meta HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-2">
  <link href="styl.css" rel="stylesheet">
</head>
<body>
<h1>To jest strona testowa</h1>

<p>Tak, zgadza się: musimy się liczyć z powierzchownym odbiorem tekstu. Nie miej
tutaj na ocenę tego, czy to dobrze, czy to źle. Jak często bywa, kluczowe jest
zachowanie właściwych proporcji:
<dl>
<dt>Autor</dt>
<dd> nie powinien popadać w przesadę, tworząc dokumenty hipertekstowe zbyt
poszatkowane na odseparowane części, ryzykując brak syntetycznego spojrzenia.
</dd>
```

Okno edytora nedit

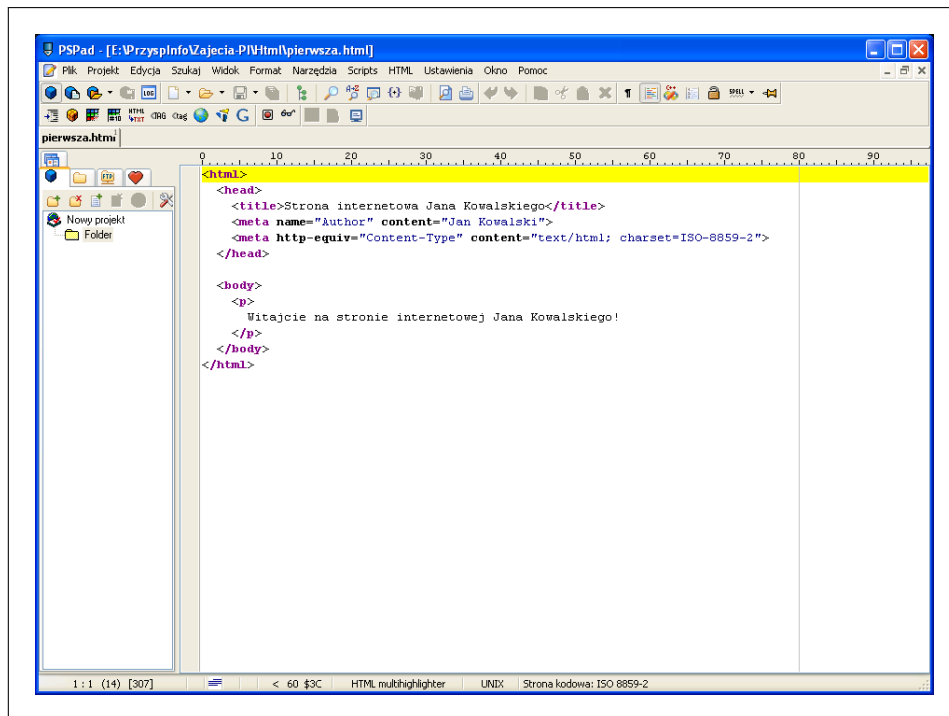
Ponadto istnieje mnóstwo tzw. edytorów webmasterskich, zwanych po prostu edytorami HTML, wspomagających tworzenie stron WWW. Jeszcze więcej jest — i są one bardzo popularne — edytorów udających edytor tekstu WYSIWYG (ang. *What you see is what you get*, tak działa np. edytor tekstów Word), ale moim zdaniem jest to najczęściej zwracanie głowy: efekty najczęściej nie są zachwycające, a jedyną zaletą takiego rozwiązania jest (pozorna) oszczędność czasu. Dlatego *zniechęcamy* do używania edytorów HTML w trybie WYSIWYG!

Do tej grupy można zaliczyć następujące aplikacje¹⁰:

Windows

- PSPad (zob. <http://www.pspad.com>) Czeski, darmowy edytor dla wielu typów plików, w tym HTML — godny polecenia, jako dobry edytor kodu źródłowego HTML!

¹⁰Dziękuję Januszowi Jabłonowskiemu za informacje o godnych uwagi edytorach webmasterskich dla Windows.



Edycja kodu źródłowego w PSPadzie. Składnia jest kolorowana, a użytkownik ma wiele narzędzi wspomagających pisanie.

- Ked (zob. <http://www.freezone.prv.pl/>) Polski, darmowy, edytor HTML; po kilku dniach przyzwyczajania się do drobnych specyficzności staje się przyjemny w użyciu.
- Pajaczek (zob. <http://www.creamsoft.com.pl/pajaczek/>) Komercyjny, polski edytor.
- Microsoft FrontPage (zob. <http://www.microsoft.com/frontpage/>) Komercyjny edytor giganta sprzedaży oprogramowania.

Linux

- Amaya (zob. <http://www.3c.org/Amaya>) Intensywnie rozwijany i propagowany przez konsorcjum W3C edytor, niestety nie umiejący jak dotąd (w wersji 9.3) w pełni współpracować polskimi czcionkami...

Pisząc polskie teksty, musimy zadbać o to, by w nagłówku pliku HTML podać poprawny sposób kodowania w nagłówku strony: taki, który faktycznie jest stosowany w naszym edytorze.

- Pisząc w Notatniku Windows, korzystamy domyślnie¹¹ z kodowania Windows-1250 i takie musimy podać w nagłówku strony;
- Dobre edytory webmasterskie zazwyczaj pozwalają ustawić kodowanie wedle życzenia i stosowny znacznik `<meta>` umieszczają w nagłówku pliku HTML; tak jest m.in. w Kompozytorze Stron przeglądarki Mozilla

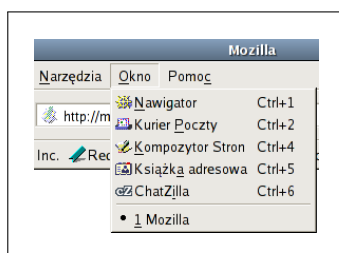
¹¹W Windows XP można wymusić kodowanie UTF-8.

- W przypadku innych programów musimy sprawdzić (zapewne eksperymentalnie), jakiego kodowania używają (nie ma reguły).

Kompozytor Stron przeglądarki Mozilla

Ta sekcja nie jest obowiązkowa.

Wygodną pozycję wśród edytorów HTML zajmuje edytor o pretensjonalnej nazwie Kompozytor Stron, który jest po prostu częścią składową (*Okno ► Kompozytor Stron*) przeglądarki Mozilla. Oto jego ważne zalety:

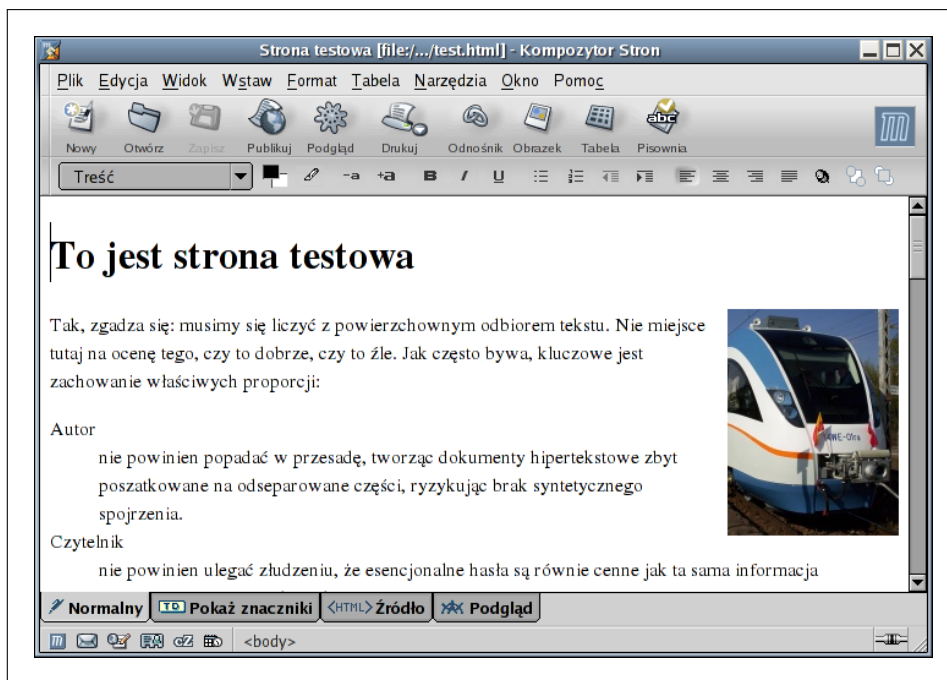


Otwieramy okno Kompozytora Stron w Mozilli

- tak jak cała Mozilla, działa praktycznie na każdym komputerze, z dowolnym systemem operacyjnym (Linux, Windows, Mac OS, itp.)
- od razu koduje polskie litery w pożądanym standardzie (np. ISO-8859-2, lub UTF-8);
- daje możliwość bezpośredniej edycji kodu źródłowego HTML (zakładka *HTML*; *Źródło* na dole okna), a z drugiej strony — podglądu widoku strony w przeglądarce (zakładka *Podgląd*)

Poważną (i zupełnie niezrozumiałą!) wadą używanego przez Kompozytora Stron edytora kodu źródłowego HTML jest to, że nie wyróżnia składników HTML w edytowanym tekście, przez co łatwiej o literówki!

- dla leniwych ma trochę możliwości formatowania WYSIWYG (ang. *What you see is what you get*) (zakładka *Normalny*). Taka metoda pracy z wielu względów nie jest zalecana.



Kompozytor Stron przeglądarki Mozilla w trybie WYSIWYG. Wyłącznie dla leniwych!

Używanie wizualnego edytora do tworzenia stron WWW zawsze prowadzi do niezbyt eleganckiego kodu, przy czym brak elegancji jest dwójakiego rodzaju:

- Nie tworzymy strony w duchu zgodnym z intencją twórców HTML:
 - * zamiast używać znaczników do oznaczenia *logicznej* struktury dokumentu, stosujemy w praktyce niemal wyłącznie znaczniki do wizualnego formatowania strony
 - * trudno zachować jednolity styl tworzonych stron, a jeszcze trudniej potem go zmienić, gdyż wszelkie formatowanie jest zaszyte w kodzie strony
- Utworzony przez edytor kod HTML najczęściej jest zaśmiecony niepotrzebnymi dodatkami: pustymi znacznikami, instrukcjami formatującymi, które zaraz są odwoływane, itp.
- Zazwyczaj generowany kod HTML jest dość przestarzały, w większości takich edytorów nawet niezgodny ze standardem HTML!

12.3.4 Jak stworzyć stronę w formacie HTML — i sprawdzić, co wyszło?

Najpierw, oczywiście, musimy wiedzieć, co napisać. Powiedzmy, że strona ma zawierać krótki tekst:

Witajcie na stronie internetowej Jana Kowalskiego!

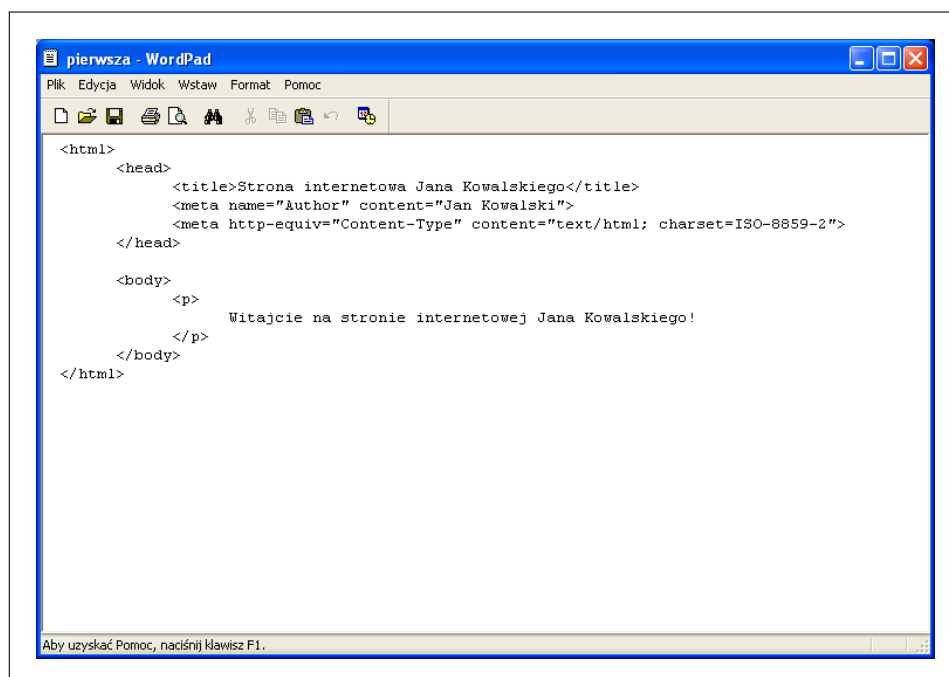
Oto przepis:

1. Uruchom dowolny edytor tekstu ASCII (np. Notatnik w Windows, nedit pod Linuxem) lub, na przykład, Kompozytor Stron przeglądarki Mozilla. O edytorach dowiesz się więcej w rozdziale 12.3.3.
2. Teraz musisz wpisać treść strony, uzupełnioną o znaczniki HTML.

Zgodnie z tym, co powiedzieliśmy dotychczas, nasz kod powinien mieć następującą postać:

```
<html>
  <head>
    <meta http-equiv="Content-Type"
      content="text/html; charset=ISO-8859-2" >
    <title>Strona internetowa Jana Kowalskiego</title>
    <meta name="Author" content="Jan Kowalski" >
  </head>
  <body>
    <p>
      Witajcie na stronie internetowej Jana Kowalskiego!
    </p>
  </body>
</html>
```

Oczywiście, rodzaj kodowania dobieramy do konkretnej sytuacji; powyżej wybrano ISO-8859-2, gdyż plik był tworzony w edytorze nedit pod Linuxem. Jeśli np. będziesz używać Notatnika pod Windows, wpiszesz raczej Windows-1250.



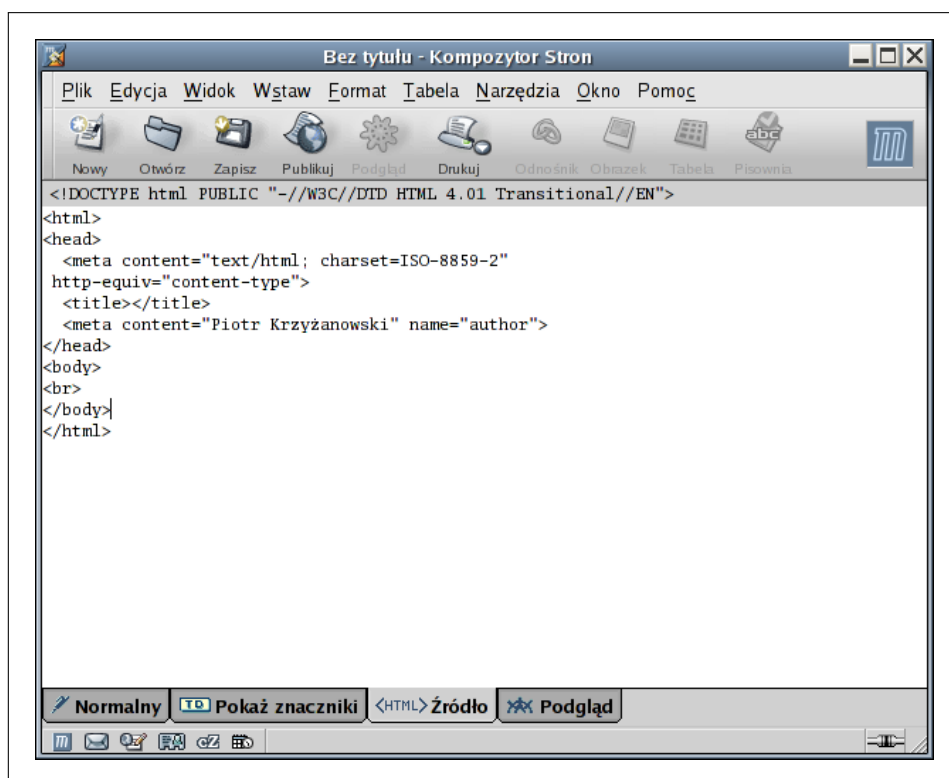
Edycja kodu źródłowego naszej strony w WordPadzie Windows.

- Jeśli używasz edytora tekstu, wpisz do edytora wskazany **powyżej** tekst.

Nie trzeba sztywno zachowywać wizualnego układu tekstu, równie dobrze można byłoby wpisać

```
<html><head><meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-2" >
<title>Strona internetowa Jana Kowalskiego</title> <meta
  name="Author" content="Jan_Kowalski"></head>
<body><p>Witajcie na stronie
internetowej Jana
Kowalskiego! </p></body></html>
```

Jednak chyba zgodzisz się, że *poprzedni* sposób rozmieszczenia znaczników i tekstu był znacznie korzystniejszy!



Rozpoczynamy edycję kodu źródłowego w Kompozytorze Stron

- Jeśli wybrałeś Kompozytor Stron przeglądarki Mozilla, kliknij zakładkę *Źródło*. Okaże się, że masz tam już wpisane niektóre znaczniki (a nawet więcej! nie przejmujesz się, tylko cieszysz, że samo się zrobiło) i jedynie we właściwym miejscu (tj.pomiędzy `<body>` i `</body>`) wpisujesz tekst „Witajcie na stronie internetowej Jana Kowalskiego!”

3. Zapisz plik na dysku, np. pod nazwą ¹² `index.html`. W nazwie pliku

¹²Użytkownicy Notatnika Windows: należy podać *explicite* rozszerzenie nazwy pliku,

nie używaj:

- polskich znaków, w rodzaju przykład.html,
- dużych liter, na przykład MojaStrona.html,
- nie-liter (tzn. znaków +,@,/, itp.): 2+2.html,
- spacji: moja strona.html.

4. Sprawdź, co wyszło, wczytując plik index.html do ulubionej przeglądarki (Plik ► Otwórz plik). W Kompozytorze Stron Mozilli, wystarczy wybrać zakładkę Podgląd.

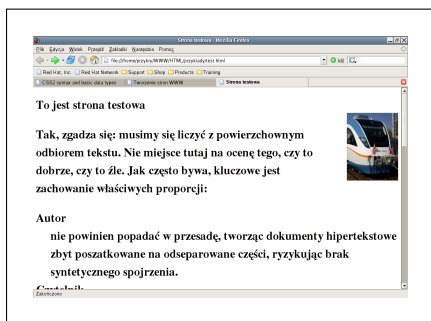
12.3.5 Uwaga na literówki!

Choć przeglądarki wybaczą wiele błędów popełnianych przez piszących strony WWW, nie wybaczą zazwyczaj

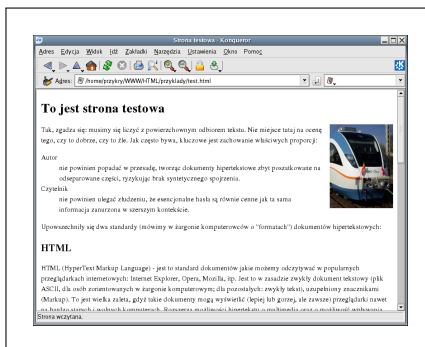
- literówek w nazwach znaczników,
- braku znacznika zamykającego (o ile jest wymagany przez standard języka).

Na pewno w trakcie swojej przygody z HTML wiele razy doświadczysz rozczarowania, gdy wskutek literówki lub drobnego zapomnienia, pieczołowicie konstruowana strona będzie się wyświetlać w nieprzewidywany sposób.

Wielką zmurą pisania w HTML są literówki uczynione w adresach internetowych: przeglądarki nie protestują, ale złe linki będą prowadzić najczęściej donikąd!



Widok pewnej strony HTML, w której zamiast znacznika kończącego `</h1>` użyto, wskutek pomyłki w trakcie pisania na klawiaturze, ponownie `<h1>`



Widok tej samej strony HTML z poprawionym kodem HTML: znacznik `<h1>` odnosi się wyłącznie do nagłówka: `<h1>`To jest strona testowa`</h1>`

- Ćwiczenie 12.3.1.** Utwórz stronę HTML według wzorca omawianego w rozdziale 12.3.4. Następnie sprawdź, co dzieje się, gdy będziesz z niego

tutaj: .html. W przeciwnym razie Notatnik przyjmie domyślne rozszerzenie .txt!

usuwać znaczniki otwierające lub zamykające. Sprawdź także, co będzie się działo, gdy w znacznikach popelnisz literówki, np. zamiast `<title>` napiszesz `<titel>`, itp. Jeśli możesz, sprawdź to w kilku przeglądarkach. □

□ **Ćwiczenie 12.3.2.** To ćwiczenie pokaże Ci, że przeglądarki są w stanie wybaczyć naprawdę wiele, chociaż nie wszystko. Utwórz prościutką stronę WWW:

```
<html>
<body>
To jest tekst. Sam go napisałem.
<a href="http://www.onet.pl">A to co to?</a> Oczywiście, link!
</body>
</html>
```

i zapisz do pliku `test.html`. Następnie sprawdź w ulubionej przeglądarce, co się dzieje, gdy:

1. Klikniesz na tekst „A to co to?”
2. Pominiesz jeden ze znaczników `<html>` lub `</html>`
3. Zamiast `` napiszesz `` albo `<\a>`
4. Zamiast `` napiszesz
 - (a) ``
 - (b) `a href="http://www.onet.pl"`
 - (c) `<a href="http://www.onet.pl`
 - (d) `<a href="http://www.onet.pl"`
 - (e) ``
 - (f) ``
 - (g) ``
 - (h) ``
 - (i) ``
5. Pominiesz znacznik ``
6. Zamiast `` napiszesz ``
7. Zamiast `` napiszesz `<a>`
8. Zamiast `` napiszesz `<b href="http://www.onet.pl">`.

Efekty sprawdź, wczytując zmieniony plik do swojej przeglądarki. Nie zapomnij, żeby po wprowadzeniu zmian w edytorze:

1. Zapisać zmiany na dysk
2. Odświeżyć widok strony w przeglądarce (w Mozilli: naciśnij guzik *Odśwież bieżącą stronę*)

□

□ **Ćwiczenie 12.3.3.**

Jaki będzie efekt wyświetlenia w przeglądarce następującego pliku HTML?

```
<html>
<head><title>Sformatowany(?) tekst</title></head>
<body>
Cześć!

Nazywam się Jan Kowalski.

W tym tygodniu mam następujące zajęcia:

1. Piłka nożna
2. Ping-pong
3. Pływanie

Zapraszam do wspólnych ćwiczeń!
                                Jan Kowalski

</body>
</html>
```

Najpierw postaraj się odgadnąć, a dopiero potem sprawdź w przeglądarce. Plik HTML zapisz np. pod nazwą `format.html`. □

Rozdział 13

Struktura strony WWW

13.1 Struktura logiczna

Każdy dobrze opracowany dokument tekstowy powinien mieć wewnętrzną strukturę logiczną. Krótka notatka — a najczęściej taki charakter będą miały strony WWW — powinna mieć tytuł (temat) i kilka sekcji (rozdziałów) opatrzonych nagłówkami (podtytułami). Każda sekcja powinna składać się z drobniejszych jednostek: akapitów. To wszystko możemy zapisać w HTML, korzystając z tak zwanych znaczników strukturalnych.

W dokumentach HTML najpierw określamy więc *logiczną strukturę* tekstu, a dopiero potem — jakby w oddzieleniu od treści tekstu — ustalamy, w jaki sposób *chcielibyśmy*, aby nasze dokumenty prezentowały się w różnych mediach: na ekranie komputera, w urządzeniu syntezującym mowę, na wydruku, itp.

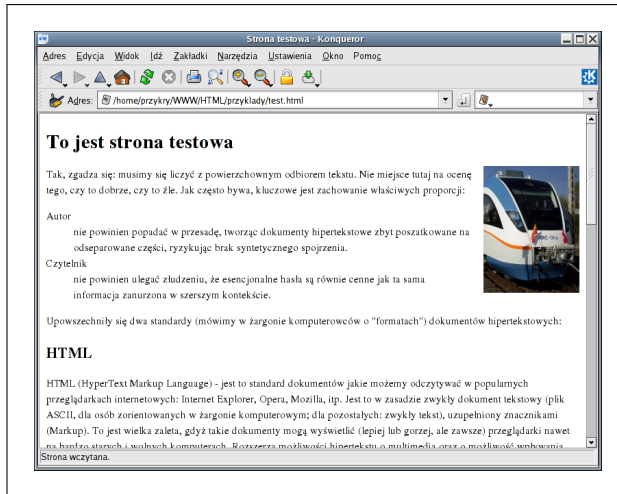
To kluczowe rozdzielenie, na

- czystą treść
- strukturę dokumentu
- wygląd (format) części składowych dokumentu

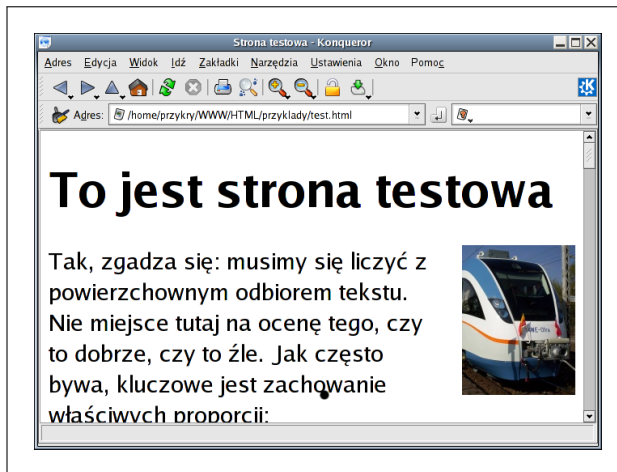
ma wielkie znaczenie, gdyż w konsekwencji uwalnia autora od przymusu zachowywania, a czytelnika od (jeszcze bardziej uciążliwego) przymusu respektowania, ustalonego formatu¹ dokumentu.

¹W niniejszej części podręcznika zajmujemy się wyłącznie strukturą logiczną tekstu. Zagadnienie zmiany jego formy omawiamy dopiero w rozdziale. [14.2](#).

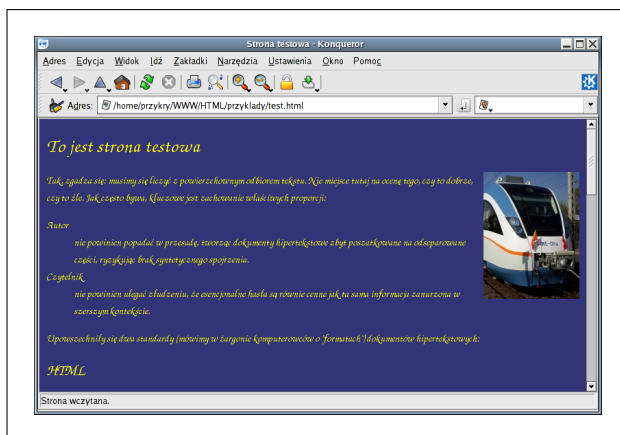
Jest to, jak już nadmienialiśmy, podejście podobne w założeniach do sposobu składu tekstu w L^AT_EX’u.



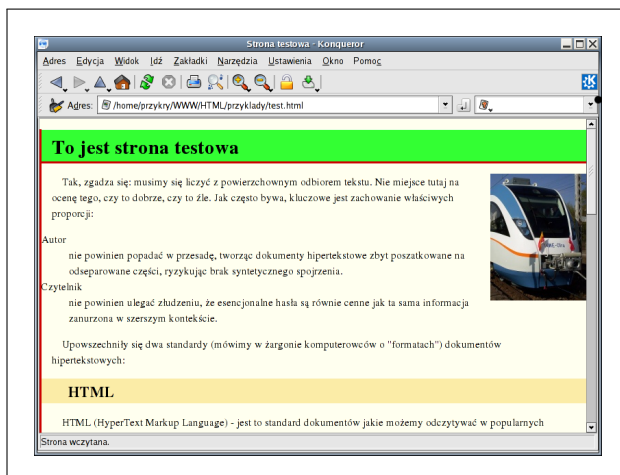
Widok pewnej strony internetowej przy standardowych ustawieniach przeglądarki



Widok tej samej strony, gdy czytelnik powiększył czcionkę w swojej przeglądarce. Zauważ, że automatycznie zmieniają się miejsca łamania linii tak, by tekst zawijał się odpowiednio do szerokości okna



Widok tej samej strony, gdy czytelnik wymusił w swojej przeglądarce wykorzystanie innych kolorów tła i czcionki.



Widok tej samej strony, gdy czytelnik wymusił na swojej przeglądarce — poprzez wskazanie własnego arkusza CSS — wykorzystanie innych kolorów tła i małej czcionki, a także zmienił sposób w jaki wyświetlane są nagłówki.

Jak widać z zamieszczonych powyżej różnych widoków tej samej strony WWW, autor w poważnym stopniu (w ekstremalnej sytuacji: praktycznie całkowicie) traci kontrolę nad sposobem, w jaki będzie wyświetlana treść dokumentu. Z punktu widzenia czytelnika dokumentu to dobrze, gdyż:

- forma dokumentu dopasowuje się do ustawień jego przeglądarki
- może np. w prosty sposób powiększyć lub zmniejszyć czcionkę
- standardowy tekst HTML zawija się tak, by zmieścić się w szerokości oknie przeglądarki, np.

Z punktu widzenia autora, jest to tylko pozornie niedobrze, bo musi on brać pod uwagę to, że jego dokument może być oglądany na rozmaite sposoby!

Na szczęście z pomocą przychodzi mu sam język HTML, bo, skoro nie ma kontroli nad formą, należy konsekwentnie ograniczyć się do kontroli **TREŚCI!** Używać jak najmniej formatowania wizualnego, a jak najwięcej logicznego (czy, jak kto woli, strukturalnego).

Zwykłemu użytkownikowi Worda pachnie to herezją:

Cooo?... Mam zrezygnować z jakiegokolwiek formatowania tekstu?! Niemożliwe! A więc to nie ja, ale czytelnik będzie decydować, czy cytaty w mojej pracy będą wyświetlone kursywą, czy może raczej na czerwono?!

Jednak, jak widzieliśmy w powyższych przykładach, **właśnie tak jest!** Za jakiś czas nauczymy się — wykorzystując arkusze stylów CSS (zob. rozdział 14.2.2), nadawać swoim dokumentom — o interesującej, miejmy nadzieję, treści — równie atrakcyjną formę, którą powinna respektować przeglądarka. Ale pamiętajmy: nigdy nie mamy gwarancji, że tak właśnie będzie.

Znaczniki struktury logicznej są więc podstawową — w skrajnych wypadkach jedyną — drogą poinformowania (przeglądarki) czytelnika o charakterze czytanego przezeń tekstu.

Omówimy teraz znaczniki określające główne zręby struktury logicznej tekstu. Znaczniki te dotyczą (dużych) bloków tekstu, rozdzielając dokument na części składowe o jasno określonym charakterze: rozdziały, akapity, itp. Dlatego — gdy oglądamy dokument HTML w przeglądarce — każdy z elementów strukturalnych standardowo rozpoczyna się od nowego wiersza.

13.1.1 Szkielet tekstu

Oto przykład typowej, krótkiej strony WWW w języku HTML²:

²Ograniczamy się tutaj wyłącznie do zawartości znacznika `<body>`; co powinno znajdować się w nagłówku takiej strony, niech będzie ćwiczeniem dla Czytelnika, zob. rozdział 12.3.2.

```
<body>
<h1>Dania rybne</h1>

<p>Większość potraw z ryb może być
przyrządzona niewielkim nakładem sił
i środków. Nawet skromny budżet
nie będzie przeszkodą, jeśli
wyberzemy morszczuka czy mintaja.

<p>Nie polecamy kupowania ryb
mrożonych, które zazwyczaj pokryte
są wyraźną
warstwą lodu, za który niepotrzebnie
płacimy wygórowaną cenę.

<h2>Dania na gorąco</h2>

<p>Ryba przygotowana na ciepło:
smażona bądź gotowana, jest jednym
z klasycznych głównych dań
obiadowych. Atrakcyjnie podana,
może stać się punktem
kulminacyjnym niedzielnego obiadu.

<h3>Ryby smażone</h3>

<p>Zazwyczaj przygotowujemy je w
panierce. Można podawać z frytkami
lub jasnym pieczywem.

<h3>Ryby gotowane</h3>

<p>Znacznie delikatniejsze od
smażonych, mają smak subtelniejszy,
przez co dla niewyrafinowanego
podniebienia mogą być mniej
atrakcyjne od smażonych

<h2>Dania na zimno</h2>

...i tak dalej...

</body>
```

Dania rybne

Większość potraw z ryb może być przyrządzona niewielkim nakładem sił i środków. Nawet skromny budżet nie będzie przeszkodą, jeśli wybierzemy morszczuka czy mintaja.

Nie polecamy kupowania ryb mrożonych, które zazwyczaj pokryte są wyraźną warstwą lodu, za który niepotrzebnie płacimy wygórowaną cenę.

Dania na gorąco

Ryba przygotowana na ciepło: smażona bądź gotowana, jest jednym z klasycznych głównych dań obiadowych. Atrakcyjnie podana, może stać się punktem kulminacyjnym niedzielnego obiadu.

Ryby smażone

Zazwyczaj przygotowujemy je w panierce. Można podawać z frytkami lub jasnym pieczywem.

Ryby gotowane

Znacznie delikatniejsze od smażonych, mają smak subtelniejszy, przez co dla niewyrafinowanego podniebienia mogą być mniej atrakcyjne od smażonych

Dania na zimno

...i tak dalej...

Jak widzimy — i jak wcześniej zapowiadaliśmy — właściwa treść naszego dokumentu jest ujęta pomiędzy znacznikami `<body>` i `</body>`. Dzięki zastosowaniu odpowiednich znaczników HTML, nasz dokument otrzymał strukturę rozdziałów i podrozdziałów, w których zostały umieszczone kolejne akapity tekstu.

Tytuły <h1>...<h6>

Do oznaczania tytułów rozdziałów (ang. *headings*), lub podrozdziałów, służą znaczniki <h1>,...,<h6>, przy czym <h1> używamy do zaznaczenia tytułów głównych rozdziałów (najwyższego poziomu),

<h1>Potrawy z ryb</h1>

Potrawy z ryb

Dla zaznaczenia tytułów podrozdziałów skorzystamy ze znacznika <h2>,

<h2>Dania na gorąco</h2>

Dania na gorąco

i konsekwentnie, <h3> użyjemy do zaznaczenia pod-pod-rozdziałów, np.

<h3>Ryby smażone</h3>

Ryby smażone

Jako się rzekło, możliwych poziomów pod-pod-...-rozdziałów jest aż sześć: do <h6>.

Akapity <p>

Znacznik <p> rozpoczyna nowy akapit (to znaczy: ciągły blok tekstu, ang. *paragraph*). Jak widzimy, ten znacznik nie musi mieć znacznika kończącego: przyjmuje się, że jeśli </p> nie wystąpił, to koniec akapitu jest tam, gdzie rozpoczyna się kolejny element strukturalny.

Ponieważ obszar ciągłego tekstu to właśnie akapit, w powyższym przykładzie wstawiliśmy znacznik <p> na początku każdego takiego zwartego bloku tekstu, np.:

<p>Większość potraw z ryb może być przyrządzona niewielkim nakładem sił i środków. Nawet skromny budżet nie będzie przeszkodą, jeśli wybierzemy jedną z tańszych ryb, w rodzaju morszczuka czy mintaja.

Większość potraw z ryb może być przyrządzona niewielkim nakładem sił i środków. Nawet skromny budżet nie będzie przeszkodą, jeśli wybierzemy jedną z tańszych ryb, w rodzaju morszczuka czy mintaja.

Większość przeglądarek, z wyjątkiem Opery i Konquerora, nie potrafi obecnie automatycznie numerować rozdziałów, chociaż taką możliwość przewiduje standard CSS 2.1.

Sekcje <div>

Dodatkowo, każdą z jakichkolwiek powodów logicznie spójną dużą część treści dokumentu możemy oznaczyć, korzystając ze znacznika <div>. Obecnie nie będziemy mieli zeń większego pożytku, ale gdy zaczniemy korzystać z formatowania stron za pomocą arkuszy stylów CSS, jego znaczenie istotnie wzrośnie. Przykładowo, będziemy mogli użyć tego znacznika do określenia większego fragmentu tekstu, który ma zostać wyróżniony poprzez otoczenie go ramką, zob. rozdział 14.2.4.

Przykład 13.1.1. Przeniesienie struktury tekstu drukowanego na znaczniki HTML Fragment gimnazjalnego podręcznika do plastyki³ moglibyśmy następująco zapisać w formacie HTML:

<code><h1>Sztuka manieryzmu i baroku</h1></code>	Sztuka manieryzmu i baroku
<code><p>Pogodna, pełna harmonii i umiaru sztuka odrodzenia, była odbiciem wewnętrznego spokoju człowieka. Po roku 1545 w sztuce, wyrażającej dotychczas radość płynącą z bezpieczeństwa człowieka wobec Boga i władzy, pojawił się niepokój i dramat.</p></code>	Pogodna, pełna harmonii i umiaru sztuka odrodzenia, była odbiciem wewnętrznego spokoju człowieka. Po roku 1545 w sztuce, wyrażającej dotychczas radość płynącą z bezpieczeństwa człowieka wobec Boga i władzy, pojawił się niepokój i dramat.
<code><h2>Rzeźba baroku</h2></code>	Rzeźba baroku
<code><p>Forma rzeźby barokowej, w porównaniu z rzeźbą epoki odrodzenia, uległa bardzo znacznym zmianom. Przede wszystkim statyczny, spokojny układ figur uległ silnemu poruszeniu.</p></code>	Forma rzeźby barokowej, w porównaniu z rzeźbą epoki odrodzenia, uległa bardzo znacznym zmianom. Przede wszystkim statyczny, spokojny układ figur uległ silnemu poruszeniu.
<code><h2>Malarstwo baroku</h2></code>	Malarstwo baroku

³S. Stopczyk, *Plastyka* (podręcznik do klasy I gimnazjum), WSIP, Warszawa 1999.

<p>...itd...

...itd...

Przykład 13.1.2. Rozbudowa strony index.html

Przyjrzyjmy się, jak możemy rozbudować naszą stronę index.html, korzystając z poznanych znaczników. Przedstawiamy pełny kod HTML (to znaczy, pełną zawartość pliku index.html).

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;_
      charset=ISO-8859-2" >
    <title>Strona internetowa Jana Kowalskiego</title>
    <meta name="Author" content="Jan_Kowalski" >
  </head>
  <body>
    <h1>Jan Kowalski</h1>

    <p>Witam wszystkich na swojej stronie internetowej.

    <h2>Życiorys</h2>

    <h3>Dzieciństwo</h3>

    <p>Urodziłem się 12 grudnia 1981r. Dzieciństwo spędziłem w
      Kałuszynie: małym
      miasteczku na wschodzie Polski.

    <h3>Szkoła podstawowa</h3>

    <p>...co to się działo!...

    <p>Jeśli wierzyć opowieściom starszego brata, na studiach czeka
      mnie coś podobnego: znów będę robić tylko to, co mnie fascynuje,
      a studia będą nie tylko odkrywaniem nowych możliwości, ale także
      nowych ludzi.

    <h3>Szkoła średnia</h3>

    <p>...itd...

  </body>
</html>
```

□ **Ćwiczenie 13.1.1.** Stwórz małą, własną stronę domową WWW, wzorując się na opisywanej wyżej stronie Jana Kowalskiego. Zapisz ją do pliku index.html. Pamiętaj, by podać właściwe parametry kodowania polskich liter i sprawdzić efekt w przeglądarce! □

□ **Ćwiczenie 13.1.2.** Napisz dokument, omawiający *skrótowo*, w kolejnych podrozdziałach, współczesną historię Polski. Pamiętaj o dobraniu właściwego tytułu dokumentu (znaczniki: <title> oraz <h1>) oraz treściwych

tytułach podrozdziałów <h2>. Zwiększ czytelność tekstu, dzieląc go na niezbyt obszerne akapity. Staraj się ponadto skorzystać ze wszystkich poznanych dotąd znaczników. Sprawdź efekt w swojej przeglądarce, a jeśli dysponujesz kilkoma — w każdej z nich. □

13.1.2 Listy

Trudno przecenić znaczenie list. W porównaniu do tej samej treści podanej w formie ciągłego tekstu, znacznie bardziej przykuwają uwagę czytelnika, punktując (no właśnie!) najważniejsze informacje np. lepiej czyta się poniższy fragment tekstu:

Popularne edytory tekstu to

- Microsoft Word
- Word Perfect
- OpenOffice Writer.

aniżeli ten sam tekst, ale bez struktury listy:

Popularne edytory tekstu to Microsoft Word, Word Perfect i OpenOffice Writer.

Zwłaszcza w dokumentach publikowanych w sieci WWW jest to bardzo ważne, gdyż statystyki wykazują, że czytelnik strony WWW zazwyczaj czyta ją dość pobieżnie. Na liście znacznie łatwiej „zawiesić oko”, niż na długim fragmencie ciągłego tekstu.

Pamiętajmy jednak, że listy mają też swoje minusy: prowadzą do skrótowości i szkicowości tekstu (łatwo to zrozumieć, patrząc na wręczalki z prezentacji).

Podobnie jak w innych dokumentach elektronicznych, w HTML również występuje kilka odmian list.

Wypunktowania

Chcąc wymienić swoje zainteresowania, użylibyśmy listy nienumerowanej, np.

Znacznik kończący element listy, może być pominięty.

```
<h2>Zainteresowania</h2>
```

```
<ul>
```

```
<li>HTML</li>
```

```
<li>brydż (sportowy)</li>
```

```
<li>filmy grozy</li>
```

```
<li>kuchnia kawalerska</li>
```

```
</ul>
```

Zainteresowania

- HTML
- brydż (sportowy)
- filmy grozy
- kuchnia kawalerska

Wyliczenia

Z kolei, gdybyśmy mieli np. ochotę podać na swojej stronie HTML przepis na jajecznicę, należałoby użyć listy numerowanej, jak poniżej

```
<h3>Przepis na jajecznicę</h3>
```

```
<ol>
```

```
<li>Rozpuścić na patelni odrobinę  
masła</li>
```

```
<li>Wbić 3 jajka</li>
```

```
<li>Mieszać energicznie zawartość,  
aż wszystko się zetnie</li>
```

```
<li>Podawać na gorąco, zagryzać  
ciemnym chlebem</li>
```

```
</ol>
```

Przepis na jajecznicę

1. Rozpuścić na patelni odrobinę masła
2. Wbić 3 jajka
3. Mieszać energicznie zawartość, aż wszystko się zetnie
4. Podawać na gorąco, zagryzać ciemnym chlebem

Listy z opisem <dl>

Czasami zachodzi potrzeba wyliczenia pewnych pojęć, do których podajemy następnie opis, na przykład:

Przeglądarki internetowe

Jest bardzo wiele przeglądarek stron WWW i każda z nich ma swoje wady i zalety. Poniżej opisujemy kilka najbardziej popularnych.

Mozilla/Firefox Popularna, bardzo dobra przeglądarka, która zadziała na praktycznie dowolnym komputerze

Internet Explorer Dostępna na każdym komputerze z MS Windows, nigdy nie będzie działać pod Linuxem. Ma wiele kaprysów.

Opera Ma grono zagorzałych zwolenników, mała, szybka, ale niestety w darmowej wersji wyświetla reklamy.

Taki fragment tekstu opisalibyśmy w HTML korzystając ze znaczników listy z opisem. Jak łatwo się domyślić, będzie ona ciut bardziej skomplikowana od poprzednich, gdyż dla każdej pozycji listy musimy zdefiniować: opisywane hasło oraz właściwy opis:

```
<h2>Przeglądarki internetowe</h2>
```

```
<p>Jest bardzo wiele przeglądarek stron WWW i każda z nich ma swoje wady i zalety. Poniżej opisujemy kilka najbardziej popularnych.</p>
```

```
<dl>
```

```
<dt>Mozilla/Firefox</dt>
```

```
<dd>Popularna, bardzo dobra przeglądarka, która zadziała na praktycznie dowolnym komputerze</dd>
```

```
<dt>Internet Explorer</dt>
```

```
<dd>Dostępna na każdym komputerze z MS Windows, nigdy nie będzie działać pod Linuxem. Ma wiele kaprysów.</dd>
```

```
<dt>Opera</dt>
```

```
<dd>Ma grono zagorzałych zwolenników, mała, szybka, ale niestety w darmowej wersji wyświetla reklamy</dd>
```

```
</dl>
```

Przykład 13.1.3.

Wzbogacimy naszą stronę internetową `index.html` o informacje dotyczące różnych sposobów kontaktu z Janem Kowalskim.

```
<h2>Kontakt</h2>
```

```
<dl>
```

```
<dt>Tel. stacjonarny</dt>
```

```
<dd>22 998 7777</dd>
```

```
<dt>Tel. komórkowy</dt>
```

```
<dd>88 998 7777</dd>
```

```
<dt>Gadu–Gadu</dt>
```

```
<dd>janek2005</dd>
```

```
<dt>E–mail</dt>
```

```
<dd>jan.kowalski@students.uw.edu.pl</dd>
```

```
</dl>
```

Kontakt

Tel. stacjonarny

22 998 7777

Tel. komórkowy

88 998 7777

Gadu-Gadu

janek2005

E-mail

jan.kowalski@students.uw.edu.pl

Ćwiczenie 13.1.3. Wzbogać własną stronę internetową `index.html` o listę zainteresowań. Pamiętaj o użyciu nagłówka i wyborze odpowiedniego wariantu listy.

Ćwiczenie 13.1.4. Popraw, m.in. uzupełniając go znacznikami, plik `format.html` z ćwiczenia 12.3.5 tak, by nadać mu odpowiednią strukturę, a tym samym wygląd w przeglądarce.

Ćwiczenie 13.1.5. Napisz stronę internetową `pomoc.html`, na której wymienisz kolejne kroki, jakie są niezbędne, by na Twoim komputerze utworzyć jakąś stronę internetową.

13.1.3 Tabele <table>

Są pozornie dość skomplikowanym elementem HTML. Kiedyś używano tabel także do pozycjonowania elementów składowych złożonych stron HTML, w rodzaju witryn portali (zob. <http://www.mimuw.edu.pl>), obecnie tabele tracą — na szczęście! — to zastosowanie na rzecz bardziej eleganckich rozwiązań jakimi są arkusze stylów CSS.

Wyniki testu

	Punkty	Ocena
Zadanie 1	10	5
Zadanie 2	8	4
Zadanie 3	4.5	2+

A oto kod HTML, stojący za powyższą prościutką tabelą. Pozornie jest dość skomplikowany — ale to tylko pozory!

```
<table>
<caption>Wyniki testu</caption>
<tr>
  <td></td> <th>Punkty</th> <th>Ocena</th>
</tr>
<tr>
  <th>Zadanie 1</th> <td>10</td> <td>5</td>
</tr>
<tr>
  <th>Zadanie 2</th> <td>8</td> <td>4</td>
</tr>
<tr>
  <th>Zadanie 3</th> <td>4.5</td> <td>2+</td>
</tr>
</table>
```

Przeanalizujmy ten kod.

Cała tabela znajduje się pomiędzy znacznikami `<table>` i `</table>`. Najpierw określamy podpis dla tabeli:

```
<caption>Wyniki testu</caption>
```

Następnie podajemy zawartość kolejnych wierszy tabeli:

```
<tr>...zawartość pierwszego wiersza...</tr>
```

```
<tr>...zawartość drugiego wiersza...</tr>
```

....i tak dalej.

W każdym wierszu tabeli `<tr>...</tr>` podajemy zawartość kolejnych komórek znajdujących się w tym wierszu. Ponieważ w każdym wierszu znajdują się trzy komórki, typowy wiersz jest postaci:

```
<tr>
  <th>Zadanie 2</th> <td>8</td> <td>4</td>
</tr>
```

Zawartość kolejnych komórek ujmujemy w znaczniki `<td>` i `</td>`. Poza `<td>`, jest jeszcze jeden znacznik do określenia zawartości komórek: `<th>` — znacznik dający logiczną informację, że dana komórka jest komórką *nagłówka* tabeli. Takimi właśnie komórkami są w naszym przykładzie komórki zawierające numer zadania.

Przeglądarki zwykle wyświetlają zawartość znaczników `<th>` wytłuszczoną czcionką. Komórką nagłówkową może być dowolna komórka tabeli.

Parametry tabeli

Sam znacznik `<table>` może mieć kilka parametrów określających sposób wyświetlenia tabeli. Przykładowo,

```
<table border="1px" width="80%" >
```

oznacza, żeby tabela zajęła 80% szerokości strony, a jej komórki zostały obwiedzione wąską: ramką grubości 1 piksela.

Staramy się nigdy nie używać bezwzględnych jednostek miary (pikseli, centymetrów, itp.) do określenia szerokości tabeli i jej komórek. Jeśli już chcemy wpływać na szerokość wyświetlanej tabeli, to używamy określeń w procentach (bieżącej szerokości okna przeglądarki), co gwarantuje większą niezależność strony od środowiska: `<table border="1px" width="80%" >`. Wszak nasza strona może wyświetlać się na monitorach o różnych rozdzielczościach/rozmiarach, dlatego specyfikacja szerokości w jednostkach bezwzględnych (cm, px), prowadziłaby czasem do przykrych niespodzianek!

Wysokość tabeli praktycznie zawsze pozostawiamy swobodną: po prostu niech tabela będzie „tak wysoka, jak potrzeba”). Ponadto zwracam uwagę, że szerokość ramki tabelki podajemy mimo wszystko w pikselach! (dlaczego?)

Przykład 13.1.4. Centrowanie tabeli na stronie Czasami chcemy, by tabela wyświetliła się pośrodku strony, jak poniżej:

	Punkty	Ocena
Zadanie 1	10	5
Zadanie 2	8	4
Zadanie 3	4.5	2+

W tym celu musimy użyć dodatkowego parametru znacznika `<table>`, mianowicie parametru `style`:

```
<table style="margin:auto;" >  
....  
</table>
```

13.1.4 Podsumowanie

Znaczniki HTML, jakich możemy użyć do oznaczenia logicznej zawartości dużych fragmentów (bloków) tekstu to

- Nagłówki rozdziałów i podrozdziałów: `<h1>`, `<h2>`, ... `<h6>`
- Akapity: `<p>`
- Cytat bloku tekstu: `<blockquote>` (wymaga wewnątrz np. akapitu!)
- Listy

- Numerowane: ``
- Nienumerowane: ``
- Z opisem: `<dl>`
- (Pozycja na liście: ``)
- Tabele: `<table>`, `<tr>`, `<td>`, `<th>`
- Inne logiczne sekcje dokumentu (do późniejszego wykorzystania w CSS): `<div>`

To, jak faktycznie będą formatowane nasze nagłówki, paragrafy itp. możemy pozostawić przeglądarce (i na razie tak robimy), albo określić — hurtem dla wielu dokumentów — korzystając z arkusza stylu CSS, o czym mówimy w rozdziale [14.2.2](#).

13.2 Znaczniki nadające tekstowi dodatkowy sens

W poprzednim rozdziale omówiliśmy znaczniki strukturalne, tzn. dzielące dokument na duże bloki o określonym charakterze. Obecnie zajmiemy się znacznikami, które stosuje się do oznaczania krótkich fragmentów tekstu — a więc niejako *wewnątrz* bloków tekstu oznaczanych znacznikami strukturalnymi.

Z częścią tych znaczników już spotkaliśmy się, przypominamy je teraz dla porządku.

13.2.1 Linki `<a>`

To one, oprócz możliwości wpływu na wygląd (oraz wszystkich udogodnień z jakimi wiąże się elektroniczna forma dokumentu: wyszukiwania, kopiowania, itp., a także — trochę przesadnie eksponowanych — możliwości zamieszczania multimediiów), stanowią o jakościowej wyższości dokumentów HTML nad zwykłym tekstem drukowanym.

W przykładzie [13.1.1](#) znajduje się m.in. takie zdanie:

Dzieciństwo spędziłem w Kałuszynie, małym miasteczku....

Zdanie to, zamieszczone na stronie WWW, niewątpliwie zyskałoby nowy walor, gdyby słowo „Kałuszyn” było *linkiem* do strony internetowej poświęconemu temu miastu. Taka strona rzeczywiście istnieje, a jej adres internetowy to

`http://www.kaluszyn.pl`

Nie zdziwi nas zapewne, że utworzenie linku w pliku HTML jest bardzo proste⁴! Frazę, która ma być linkiem do innej strony (u nas: po prostu słowo „Kałuszyn”) należy otoczyć znacznikami `<a>` i `` (ang. *anchor*), przy czym jako parametr znacznika otwierającego podajemy adres strony, do której link ma prowadzić:

Dzieciństwo spędziłem w <code></code> Kałuszynie <code></code> , małym miasteczku...	Dzieciństwo spędziłem w Kałuszynie, małym mia- steczku...
---	---

Dając odsyłacz na zewnątrz naszych stron WWW należy oczywiście podać pełny adres internetowy, łącznie z „http://...”, np. w powyższym przykładzie: ``. Natomiast gdy odsyłacz prowadzi do innej strony, która znajduje się na tym samym serwerze co bieżąca, wygodniej jest użyć adresu względnego (tzn. względem położenia strony na której ten link się znajduje), np. ``, o ile link wiedzie do strony `formatowanie.html`, znajdującej się w tym samym katalogu serwera WWW, co nasza strona.

Możliwe jest także podanie linku do swojej skrzynki e-mail:

```
<a href="mailto:jan.kowalski@students.uw.edu.pl">Napisz do mnie e-mail!</a>
```

Kliknięcie w taki link spowoduje (w prawidłowo skonfigurowanym systemie) otwarcie aplikacji poczty elektronicznej z od razu wpisanym adresatem, tutaj: `jan.kowalski@students.uw.edu.pl`.

Uwaga dla osób pracujących w systemie Windows

Użytkownicy systemu Windows muszą pamiętać, że katalogi w adresie internetowym oddzielamy od siebie znakiem „/”, a nie „\”, do czego być może przywykli w Windows. Chcąc odwołać się do dokumentu `test.html`, położonego w podkatalogu `przyklady`, napiszemy

```
<a href="przyklady/test.html">
```

a nie ``. To wyjątkowo denerwujący, choć całkiem częsty błąd!

Przykład 13.2.1. Adres email w stopce strony WWW

Jednowierszowe tabele doskonale nadają się do tego, by część elementów wiersza była na stronie dosunięta do lewego, a część do prawego marginesu. Daje to sposobność utworzenia elegancko wyglądającej stopki strony WWW:

⁴W końcu HTML nie bez kozery oznacza **HyperText Markup Language**!

Koniecznym pamiętać: adres internetowy *musisz* podać używając podwójnych cudzysłówów, dokładnie tak, jak w podanym przykładzie! Dodatkowo, wielką zgorą pisaną w HTML są literówki uczynione w adresach internetowych: przeglądarki nie protestują, ale złe linki będą najczęściej prowadzić donikąd!

Linki bez znacznika zamykającego `` są zabójcze dla strony!!!

...
...tekst strony...

...

© Jan Kowalski

[Powrót do strony głównej](#)

Taki efekt uzyskamy, stosując jednowierszową tabelę, której komórki uzupełnimy informacją o sposobie rozmieszczenia w nich tekstu:

```
<hr>
<table width="100%" border="0">
<tr>
  <td style="text-align:left" >
    &copy; <a href="mailto:jan.kowalski@students.uw.edu.pl">Jan
      Kowalski</a>
  </td>
  <td style="text-align:right" >
    <a href="http://students.uw.edu.pl/jan.kowalski">
      Powrót do strony głównej
    </a>
  </td>
</tr>
</table>
```

Znacznik `<hr>` daje efekt odkreślenia poziomą (ang. *horizontal*) linią.

13.2.2 Wyróżnienia

Proste wyróżnienie ``

To drzewo
``naprawdę``ma
100 lat!

To drzewo *naprawdę* ma
100 lat!

Użyty znacznik ``, czyli wyróżnienia (ang. *emphasis*), nadaje słowu „naprawdę” dodatkowe (niezwerbalizowane w samym tekście!) znaczenie: na słowo „*naprawdę*” należy położyć nacisk. Przeglądarki internetowe stosują w tym celu kursywę (tak, jak w przykładzie powyżej).

Mocne wyróżnienie ``

To drzewo ma ``100
lat``!

To drzewo ma **100 lat!**

Dla silnego podkreślenia faktu — że to **aż sto** lat! — mamy w HTML znacznik ``: mocno zaznacza wagę tekstu, którego dotyczy. Przeglądarki zgodnie wyświetlają zawartość takiego znacznika pogrubioną czcionką.

Cytat `<q>`

Jest takie porzekadło: <code><q></code> Wysoki jak brzoza, a głupi jak koza <code></q></code> ...	Jest takie porzekadło: “ <i>Wysoki jak brzoza, a głupi jak koza</i> ”...
---	---

Znacznik `<q>` oznacza krótki cytat (ang. *quotation*), wpleciony w tekst. Zauważmy, że rola `<q>` w HTML jest identyczna jak rola znaku cudzysłowu w zwykłym tekście drukowanym. Dlatego, jak widzimy po prawej stronie, dobra przeglądarka zinterpretuje ten znacznik otaczając tekst cudzysłowami (i, najczęściej, dodatkowo wyróżni tekst cytatu kursywą).

Inne fragmenty tekstu ``

Stosowany w tym samym celu, co `<div>`, ale w odniesieniu do krótkich — logicznie spójnych — fragmentów dłuższych tekstów (w odróżnieniu do `<div>`, który odnosił się do *bloku* tekstu). Warto go stosować dopiero w połączeniu z CSS.

Ćwiczenie 13.2.1. Uzupełnij swoją stronę `index.html` o elegancką stopkę, zawierającą link do Twojej skrzynki pocztowej oraz link do strony głównej Twojego Wydziału.

Ćwiczenie 13.2.2. Zapisz na swojej stronie `index.html` dodatkowe informacje, tym razem o Twoim wykształceniu. Zamieść linki do strony domowej wydziału, na którym studiujesz, oraz do głównej strony Uniwersytetu Warszawskiego. Tam, gdzie to możliwe, dodaj także linki do szkół, jakie ukończyłeś.

Ćwiczenie 13.2.3. Utwórz stronę `przedmiot.html`, na której w kilkunastu akapitach opowiesz o swoim ulubionym przedmiocie. Pamiętaj, by:

- Poprawnie sformułować zawartość nagłówka `<head>` strony
- Nadać dobre śródtytuły poszczególnym częściom tekstu
- Użyć w tekście właściwych znaczników strukturalnych i znaczeniowych. Zapewne zechcesz skorzystać, prócz nagłówków i akapitów, z list oraz wyróżnień. Dla słów lub fraz wymagających objaśnienia, dodaj linki do konkretnych haseł w Wikipedii (zob. <http://www.wikipedia.pl>)

Ogólna reguła: tekst z linkami powinien dawać się gładko przeczytać na głos i mieć taki sam sens jak tekst bez linków.

Ćwiczenie 13.2.4. Dopisz na końcu strony `index.html` linki do wszystkich innych do tej pory stworzonych przez Ciebie plików HTML. Użyj znacznika listy do wymienienia kolejnych linków. Nie zapomnij także o gładkim sformułowaniu linkowanego tekstu: linki postaci

```
Informacje o UW znajdziesz <a
href="http://www.uw.edu.pl" >
tutaj </a>
```

Informacje o UW znaj-
dziesz **tutaj**

są zwykle świadectwem niechlujstwa redakcyjnego! Albowiem znacznie lepiej wygląda

```
<a
href="http://www.uw.edu.pl" >
Informacje o UW</a>
```

Informacje o UW

Ćwiczenie 13.2.5. Uzupełnij swoją stronę o motto życiowe, będące pewnym cytatem. Nie zapomnij, że na oznaczenie cytatu jest w HTML specjalny znacznik. Pamiętaj również, że cytując musisz także podać źródło!

13.2.3 Przydatne drobiazgi

Twardy znak łamania wiersza

Ten znacznik nie ma znacznika zamykającego!

Daje nam możliwość zręcznego pisania np. adresów i innych akapitów, gdzie czasem trzeba niespodziewanie złamać wiersz:

```
<address>
  Jan Kowalski<br>
  E-mail: <a href="mailto:jan.kowalski@students.uw.edu.pl" >
    jan.kowalski@students.uw.edu.pl</a><br>
  WWW: <a href="http://students.uw.edu.pl/jan.kowalski" >
    http://students.uw.edu.pl/jan.kowalski</a>
</address>
```

Nie należy go nadużywać: podstawowym sposobem łamania tekstu jest jego podział na akapity (znacznikiem <p>)!

Znaczniki formatowania wizualnego <big>, <small>, , <i> oraz <tt>

Ta sekcja nie jest obowiązkowa.

W specyfikacji HTML znalazło się kilka szczątkowych znaczników formatowania wizualnego, obecnych w standardzie od samego początku. Mamy do dyspozycji pięć takich znaczników, które bez trudu rozpoznają użytkownicy L^AT_EX'a:

- — pismo **pogrubione**,
- <i> — pismo *pochyle*,
- <tt> — pismo **maszynowe**,
- <big> — powiększenie czcionki,
- <small> — zmniejszenie czcionki.

Należy jednak korzystać z nich z umiarem, a zwłaszcza — nie zastępować nimi znaczników logicznych w rodzaju czy , które domyślnie dają ten sam efekt, co <i> i , ale mają zupełnie inne znaczenie!

Z drugiej strony, rzeczywiście czasem po prostu chcielibyśmy, by fragment tekstu po prostu wyglądał odmiennie od reszty — ale wcale nie dlatego, że jest cytatem (<q>), bądź chcemy nań położyć nacisk (). Dobrze to ilustruje następujący przykład.

Przykład 13.2.2. Czasami w tekście mogą przydać się proste znaczniki formatowania wizualnego

Powiedzmy, że chcielibyśmy w tekście zamieścić spis literatury. Byłoby bardziej elegancko, jeśli ten spis wyglądałby następująco:

- **Mickiewicz**, Adam, *Pan Tadeusz*, Ossolineum, Warszawa, 1924.
- **Sienkiewicz**, Henryk, *Pan Wołodyjowski*, Wydawnictwo Literackie, Warszawa, 2002.
- ...itd...

W powyższym tekście nie występują żadne silne wyróżnienia, ale raczej ozdobniki — ułatwiające czytanie. Dlatego w HTML zapisalibyśmy to następująco:

```
<ul>
<li> <b>Mickiewicz</b>, Adam, <i>Pan Tadeusz</i>, Ossolineum, Warszawa,
1924.
<li> <b>Sienkiewicz</b>, Henryk, <i>Pan Wołodyjowski</i>, Wydawnictwo
Literackie, Warszawa, 2002.
<li> ...itd...
</ul>
```

Hmmm... eksperci od języka HTML zapewne zaleciłoby zastosowanie znacznika <cite> dla oznaczenia tytułu wymienianych dzieł, zob. także str. 359.

Kilka nietypowych znaków

Ta sekcja nie jest obowiązkowa.

Praktycznie nigdy nie będą nam potrzebne, ale dla porządku:

- Znak twardej spacji w treści dokumentu HTML uzyskujemy, pisząc ` `;
- Znak „Copyright”, ©, dostaniemy wstawiając `©`, np.

```
Copyright &copy; Jan Kowalski,  
2005
```

```
Copyright © Jan Kowal-  
ski, 2005
```

- Znaki „<” i „>”: `<` i `>`, np.

```
<h1>Znaczenie znacznika  
&lt;h1&gt;</h1>
```

```
Znaczenie znacznika  
<h1>
```

13.2.4 Podsumowanie

Dla krótkich (mniejszych od akapitu) fragmentów tekstu, HTML definiuje m.in. następujące znaczniki:

- Wyróżnienia: ``
- Mocne wyróżnienia: ``
- Cytat: `<q>`
- Odsyłacz: `<a>` (hiperlink)
- Inne krótkie fragmenty tekstu o jednolitej strukturze logicznej (do późniejszego wykorzystania w CSS): ``
- Szcątkowe znaczniki formatowania wizualnego: `<big>`, `<small>`, ``, `<i>` oraz `<tt>`: używać z umiarem!

Poznane elementy HTML opisują *sens* fragmentu tekstu, do którego się odnoszą. Tak więc, używamy np. znacznika `` nie po to, by wytłuszczyć tekst, ale po to, by zaznaczyć, że na ten fragment tekstu chcemy położyć silny nacisk. Akurat tak się składa, że przeglądarki tak oznaczony tekst wyświetlają pogrubioną czcionką. Ale już np. przeglądarka sprzężona z syntezą mowy mogłaby po prostu **dobitniej (głośniej?) odczytać** oznaczony fragment.

Zatem nasz tekst składamy z „rozdziałów” odseparowanych od siebie tytułami `<h1>`, lub podtytułami `<h2>`, ... `<h6>`. Czasem wyróżniamy w nim

sekcje korzystając ze znacznika `<div>`. Każdy paragraf markujemy znacznikiem `<p>`. Czasem warto skorzystać z wizualnego separatora `<hr>`. Skoro nie mamy pełnej kontroli nad tym, jak ma wyglądać dokument, skupiamy się na tym, jaką ma nieść treść. Piękne strony o miłej treści zostawmy amatorom.

□ **Ćwiczenie 13.2.6.** Jakich znaczników powinno się użyć do oznaczenia sensu poniższych fraz:

1. „Litwo, ojczyzno moja...”
2. Rozdział 3
3. Moje ulubione potrawy to: pierogi, bigos i zasmażka (którą po prostu uwielbiam!)

□

13.3 Publikacja

13.3.1 Poprawność językowa stron HTML

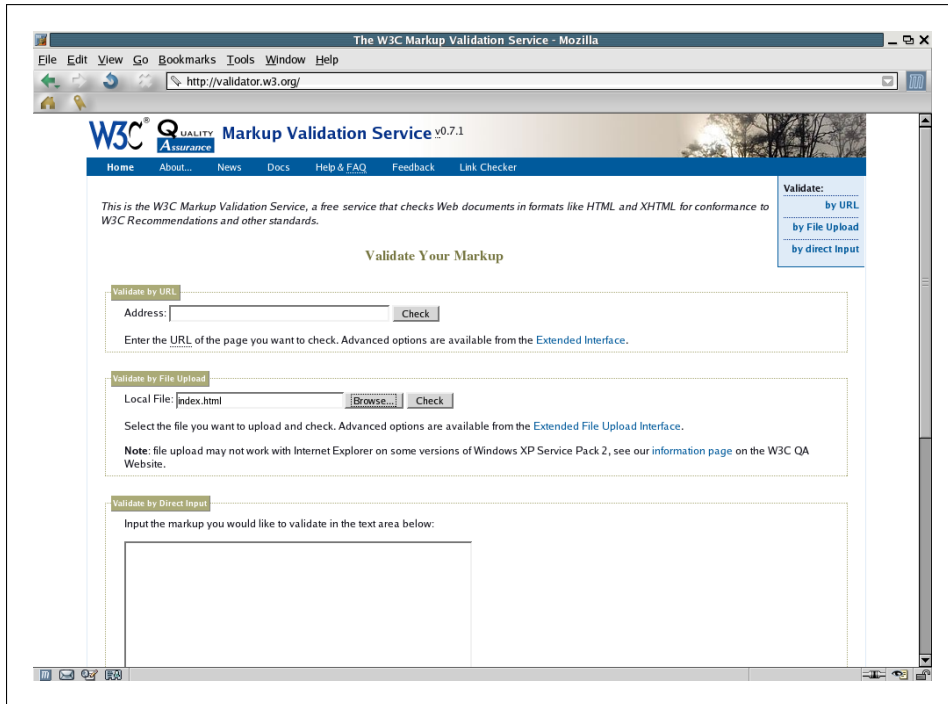
Dobre strony WWW powinny być poprawne nie tylko w zakresie języka polskiego, ale także — co równie ważne — w zakresie składni języka HTML.

Każdy dokument HTML warto, najlepiej jeszcze przed opublikowaniem, sprawdzić pod względem poprawności zawartego w nim kodu HTML. Pisząc w HTML bardzo łatwo o literówki, niedomknięte znaczniki, itp. które mogą zruinować wygląd naszej strony, w najlepszym razie: we wszystkich, a w **najgorszym** — tylko w *innych* przeglądarkach.

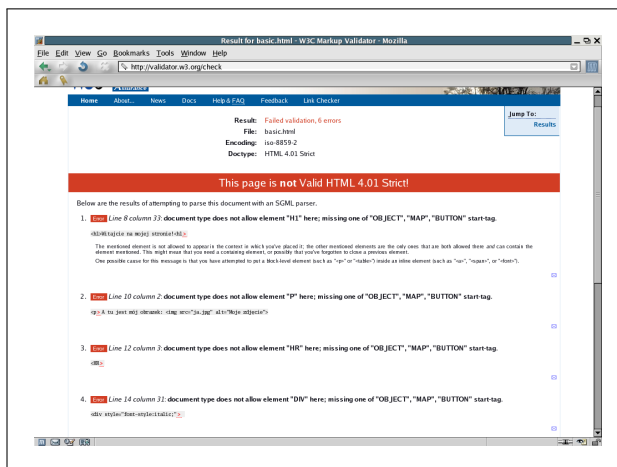
Formalną poprawność składni HTML najlepiej sprawdzić automatycznie, korzystając ze strony Markup Validation Service (zob. <http://validator.w3.org/>), zwanej popularnie Validatorem.

Składnię naszego pliku możemy sprawdzić na trzy sposoby:

- po opublikowaniu na serwerze WWW, podając adres internetowy strony
- wysyłając plik ze swojego komputera
- wpisując tekst źródłowy pliku do okienka na stronie <http://validator.w3.org/> (zob. <http://validator.w3.org/>).



Validator w akcji



Wynik sprawdzenia poprawności pliku w Validatorze. Jedyнным błędem popełnionym na stronie jest brak „/” w znaczniku zamykającym `</h1>` (przez co zamiast `</h1>` jest `<h1>`). Ten jeden błąd wywołuje jednak w Validatorze całą lawinę komunikatów o błędach.

Aby skutecznie korzystać z narzędzi sprawdzających poprawność składniową HTML, musimy precyzyjnie określić, jaką wersję standardu HTML mieliśmy na myśli, pisząc stronę HTML. Dlatego właśnie tworzone przez nas pliki zawsze uzupełniamy o informację o użytym standardzie HTML:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd" >
<html>
  <head>

```

```

    ....
    </head>

    <body>
    ....
    </body>
</html>

```

Podaliśmy tu informację, że na stronie został wykorzystany standard HTML 4.01 w wariantcie rygorystycznym (ang. *Strict*), to znaczy: korzystającym tylko z elementów języka HTML dopuszczalnych przez tę konkretną wersję (4.01) standardu HTML. Osoby mniej akuradne zapewne będą stosować inny wariant:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd" >
<html>
  <head>
    ...
  </head>

  <body>
    ....
  </body>
</html>

```

— a więc HTML 4.01 dopuszczający konstrukcje z poprzednich, mniej rygorystycznych (składniowo) wersji standardu HTML. Ogólna zasada jest taka, że łatwiej napisać poprawny składniowo kod, gdy założy się wersję HTML 4.01 Transitional.

Gdy nie mamy stałego dostępu do internetu, zamiast z Validatora, można ewentualnie skorzystać z gotowych programów wspomagających w rodzaju programu tidy (zob. <http://www.w3.org/People/Raggett/tidy/>) (m.in. edytor webmasterski PSPad korzysta właśnie z tej aplikacji w celu sprawdzenia poprawności tworzonej strony HTML).

Pamiętajmy, że wspomniane narzędzia sprawdzają jedynie samą składnię języka HTML. Nigdy nie sprawdzą nam merytorycznej poprawności, np. czy czasami nie zażądaliśmy wyświetlenia czarnego tekstu na czarnym tle (CSS), albo czy do cytowania większego fragmentu „Pana Tadeusza” niewłaściwie wybraliśmy znacznik `<code>` zamiast strukturalnie poprawnego `<blockquote>`. Mało tego, nawet nie sprawdzą, czy linki zamieszczone na naszej stronie prowadzą do faktycznie istniejących obiektów!

□ **Ćwiczenie 13.3.1.** Sprawdź wszystkie swoje dotychczas stworzone dokumenty HTML w Validatorze. Nie załamuj się, gdy zobaczysz dziesiątki błędów na krótkiej stronie HTML, tylko staraj się je *po kolei* usunąć! Składnię sprawdza automat, dlatego czasem błąd w jednym znaczniku skutkuje lawiną pozornych błędów dla dalszej części dokumentu. □

□ **Ćwiczenie 13.3.2.** To nie jest ćwiczenie na jeden raz: od tej pory, *zawsze* sprawdzaj wszystkie swoje dokumenty HTML w Validatorze. Powiedzmy to wyraźnie: odkąd wiesz, że istnieje Validator, nie ma żadnego

Między innymi, Kompozytor Stron produkuje kod HTML 4.01 Transitional.

Warto sprawdzać strony *po publikacji* na serwerze WWW! Zob. rozdział 13.3.2.

usprawiedliwienia dla ewentualnych błędów składniowych na Twoich stronach HTML! □

13.3.2 Publikacja na serwerze WWW

Nasze pieczołowicie przygotowane strony WWW musimy w końcu okazać światu. Wymaga to umieszczenia strony na serwerze WWW, który ostatecznie udostępni je zainteresowanym. Procedura postępowania zależy od konkretnego serwera WWW. Generalnie, z punktu widzenia autora stron, na serwerze jest wydzielone dlań konto i folder, do którego autor powinien skopiować publikowane pliki — uprzednio stworzone na komputerze autora. Sama procedura kopiowania zazwyczaj odbywa się z wykorzystaniem protokołu FTP, chociaż możliwe są prostsze rozwiązania.

Rola pliku `index.html`

Większość serwerów WWW traktuje w specjalny sposób pliki o nazwie `index.html` (bądź `index.htm`). Mianowicie, jeśli w przeglądarce podamy adres WWW bez podawania nazwy pliku HTML, np.

```
http://www.mimuw.edu.pl/~przykry
```

to domyślnie zostanie wyświetlona zawartość pliku

```
http://www.mimuw.edu.pl/~przykry/index.html
```

Ta reguła znacznie upraszcza podawanie adresów witryn internetowych: dlatego swojej głównej (domowej) stronie internetowej warto nadać nazwę właśnie `index.html`.

Darmowe serwery WWW

Nie każdy może sobie pozwolić, ze względu na koszty, na zakup domeny internetowej i postawienie własnego serwera, bądź na wykupienie miejsca na komercyjnym serwerze.

Rozwiązaniem stosunkowo wygodnym i całkowicie niegroźnym dla własnej sakiewki jest zamieszczenie strony na darmowym serwerze stron WWW. Naturalnie, jak wszystko, co darmowe, wiąże się to z pewnymi niedogodnościami: nasze konto będzie miało małą pojemność (kilka, czasem kilkadziesiąt MB), zazwyczaj otwarciu naszej strony będą towarzyszyć reklamy, a sam serwer zapewne nie zagwarantuje nam wiele więcej ponad obsługę plików HTML (na przykład, obsługa skryptów PHP może być ograniczona do kont płatnych). Jednym z takich darmowych serwerów jest [Republika.pl](http://republika.pl) (zob. <http://republika.onet.pl>) i na jego przykładzie omówimy procedurę publikacji własnych stron WWW.

Założenie konta jest proste. Na stronie głównej serwisu wybieramy założenie bezpłatnego konta `RepublikaSTART` i po rejestracji możemy już działać.

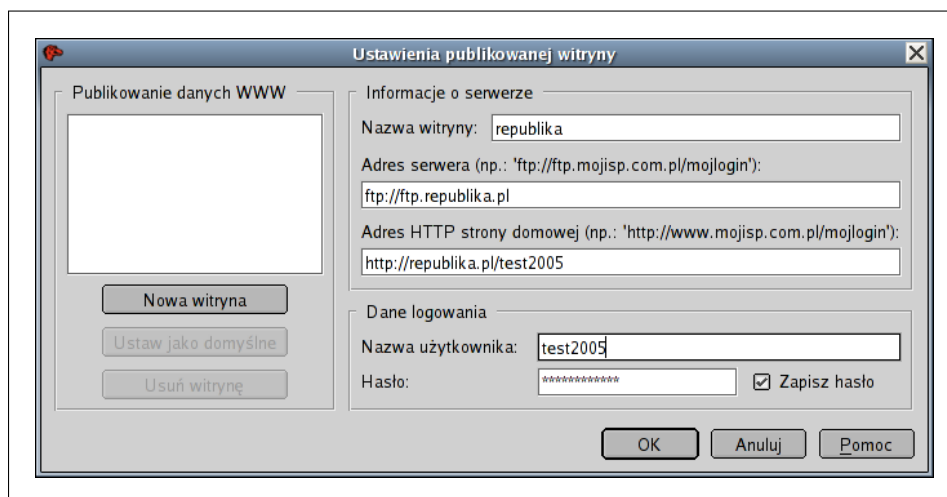
Plikiem głównym jest plik o nazwie `index.html` i taki plik *musimy* zamieścić

na serwerze obsługującym konta Republika.pl. Inne nasze strony możemy nazywać już do woli, pamiętając by (bezpośrednio lub pośrednio) wiódł do nich link z naszej strony głównej `index.html`: w przeciwnym razie (prawie) nikt na nie nie wejdzie!

Praktycznie wszystkie serwery WWW obsługują wstawianie stron przy użyciu FTP, tak jest też w przypadku Republiki. Dokładne parametry transferu (tzn. gdzie konkretnie należy wysłać plik) znajdziemy w informacjach o konfiguracji konta. Właściwe parametry transferu FTP to:

- Adres FTP: `ftp://ftp.republika.pl/`
- Użytkownik: [nazwa Twojego konta]
- Hasło: [hasło do Twojego konta]

Kompozytor Stron przeglądarki Mozilla⁵ umożliwia wygodną publikację tworzonej strony przez naciśnięcie przycisku *Publikuj*. Musimy jednak najpierw skonfigurować parametry publikacji. Najpierw, sprawdzamy, w jaki sposób należy przysyłać strony HTML na nasze konto.

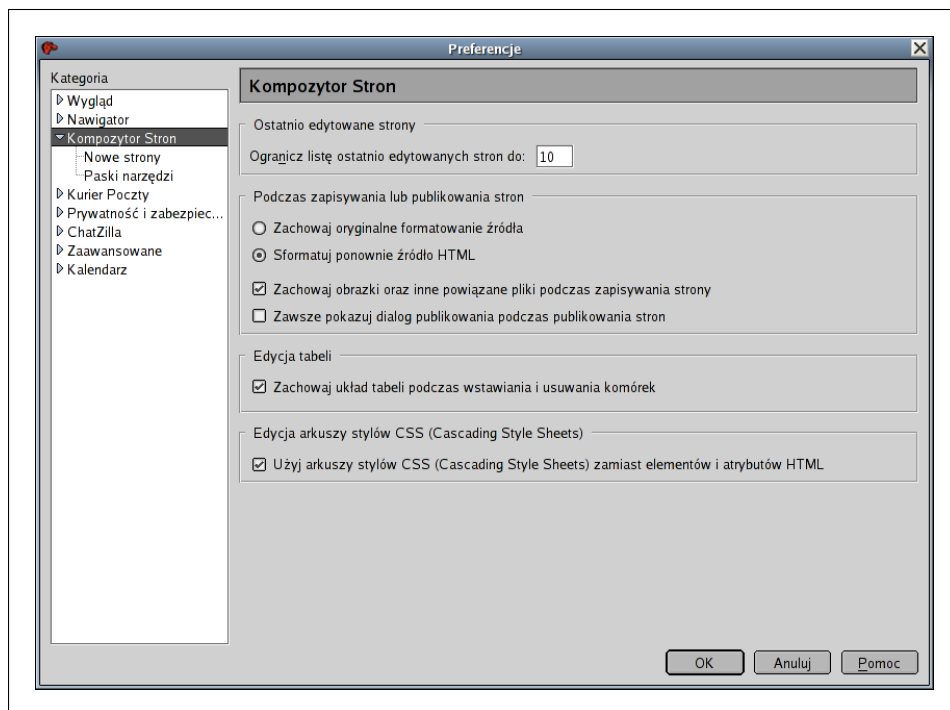


Ustawienia publikacji na serwerze Republika.pl (Kompozytor Stron: Edycja ► Ustawienia publikowanej witryny)

W przypadku, gdy nie korzystamy z Kompozytora Stron, musimy użyć jakiegokolwiek klienta FTP do przesłania swoich plików HTML na serwer WWW. (**DO ZROBIENIA** → **odsyłacz do rozdziału o FTP**)

Warto jest ustawić parametry publikowania tak, by wraz z plikiem HTML były przesyłane związane z nim pliki (np. zanurzone w treści grafiki); w tym celu zmieniamy *Preferencje* Kompozytora Stron:

⁵Podobnie jest w innych edytorach wspomagających tworzenie stron WWW.



Zaznacz Zachowaj obrazki oraz inne powiązane pliki w preferencjach Kompozytora Stron

13.3.3 Niezależność od sprzętu i przeglądarki

Prawdziwy fan przenośności powinien, przed opublikowaniem strony, dodatkowo obejrzeć ją uważnie w kilku różnych przeglądarkach, pod kilku różnymi systemami operacyjnymi. Dopiero wtedy,⁶ gdy efekt jest zadowalający we wszystkich kombinacjach, powinien ją opublikować.

W tym kursie wybraliśmy drogę prostszą, choć może mniej wizualnie spektakularną drogę: używamy oszczędnego formatowania i zadowolamy się tym, że nasze strony są formalnie zgodne ze standardami.

Należy również oszczędnie gospodarować multimediami. Wiele komputerów i/lub przeglądarek nie będzie w stanie odtworzyć plików zapisanych w mniej lub bardziej egzotycznych formatach.

⁶Jak wiadomo, przeglądarki — jak większość używanego przez nas oprogramowania — są niedoskonałe i mogą niepoprawnie implementować standard HTML i/lub CSS, zob. str. 335.

ZDECYDOWANIE NALEŻY UNIKAĆ ZAMIESZCZANIA LINKÓW DO DOKUMENTÓW W RODZAJU

- PLIKÓW WORDA,
- PREZENTACJI POWERPOINTA,
- ARKUSZY KALKULACYJNYCH EXCELA,
- INNYCH DOKUMENTÓW MICROSOFT OFFICE LUB OPENOFFICE,

TO NIE SĄ WŁAŚCIWE FORMATY PUBLIKACJI INFORMACJI NA WWW.

Więcej na ten temat znajdziesz w rozdziale [14.4](#).

13.3.4 Konserwacja stron WWW

Jedną z częściej spotykanych przypadłości opublikowanych już stron WWW — jest ich stopniowa dezaktualizacja. Powody mogą być rozmaite:

- Zmienił się nam adres e-mail, lub inne dane, które zawsze zamieszczaliśmy w stopce dokumentu
- Grafiki rastrowe, które znajdują się na naszych stronach, mają rozdzielczość nieadekwatną do rozdzielczości współczesnych monitorów
- Strony, do których prowadziły linki, już nie istnieją
- Zmieniły się sposoby publikacji stron WWW, przez co nasze są anachroniczne

Jedyną radą na takie problemy jest, niestety, ciągła aktualizacja stron. Jednak z biegiem czasu i wzrostem ilości publikowanych przez nas dokumentów zadanie aktualizacji może stawać się nadmiernie pracochłonne...

Warto wprowadzić podział na dokumenty bieżące (konserwowane) i archiwalne⁷ (których już nie będziemy modyfikować, o czym jasno uprzedzamy Czytelnika).

Kilka sposobów na to, by zautomatyzować niektóre czynności konserwacyjne omawiamy w rozdziale [14.6](#).

⁷Nie należy jednak traktować archiwum stron jako swojego śmietnika: tam również powinien panować ład, a strony niepotrzebne lub nieistotne powinny być usuwane nawet z archiwum.

Rozdział 14

Atrakcyjne strony WWW

14.1 Multimedia

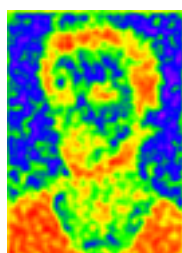
Należy zachować szlachetny umiar w nasyceniu swoich stron multimediami. Odbiorca informacji zapewne będzie wolał skorzystać z wyszukiwarki i odnaleźć w tekście konkretną frazę, niż śledzić godzinny wykład nagrany na kamerze cyfrowej lub w formie pliku mp3. Obrazki mogą ożywić stronę i wzbogacić jej treść, ale w nadmiarze szybko przeradzają się w niewiele znaczące ozdobniki: efektowne, ale bez głębszej treści.

14.1.1 Grafika

Zazwyczaj źródłem obraz(k)ów, które będziemy umieszczać na stronie WWW nie będzie program do obróbki grafiki, ale aparat cyfrowy, skaner, czy wreszcie zewnętrzne źródło w Internecie. Gdy tylko taki obrazek, o nazwie — powiedzmy — `moje-zdjecie.jpg` zamieścimy w tym samym katalogu, co nasz dokument HTML, z łatwością możemy go zanurzyć w treści dokumentu:

```
<p>  

```



Tak więc, zamieszczaniu grafiki na stronie WWW służy znacznik ``. Przyjrzyjmy mu się bliżej.

Po pierwsze, jest to ten z niewielu znaczników, który *nie ma* znacznika zamykającego! Ma on dwa obligatoryjne parametry. Parametr `src` jest nazwą wklejanego pliku z grafiką; może to być, w przypadku zamieszczania obrazków znajdujących się na innych serwerach WWW, pełny adres internetowy

tego pliku, np.

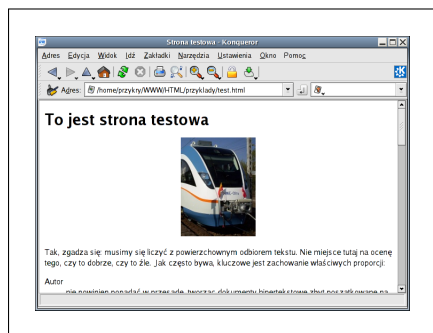
```

```

Drugi parametr jest ukłonem w stronę przeglądarek *nie obsługujących* grafiki (tak, tak — są i takie!) i zawiera krótki tekst, który taka nie-graficzna przeglądarka powinna umieścić tam, gdzie powinien znaleźć się nasz obrazek.

Jeśli chcesz wyśrodkować jakiś obrazek na stronie, powinieneś umieścić go w odpowiednio sformatowanej sekcji `<div>` (formatowanie przeprowadzisz oczywiście przy użyciu stylu CSS! — więcej na ten temat w rozdziale 14.2.2):

```
<div style="text-align:center;" >  
    
  <br >  
  Halny: nowy pociąg SKM.  
  <i>Warszawa, 2005.</i>  
</div >
```



Zdjęcie pociągu: wycentrowane na stronie



Zdjęcie pociągu: tekst oblewa grafikę z lewej strony

Zwróć uwagę, że dzięki znacznikowi `div` jasno stwierdzamy, że grafika i podpis stanowią jedną całość, tutaj: *ilustrację*.

Jeśli zaś wolisz, by tekst swobodnie oblewał (powiedzmy, z lewej strony) ten sam obrazek, musisz wpisać

```
<div style="float:right;" >  
    
  <br >  
  Halny: nowy pociąg SKM.  
  <i>Warszawa, 2005.</i>  
</div >
```

lub, dla obrazka bez podpisu,

```

```

W ten sam sposób możesz decydować, w którym miejscu na stronie będą wklejone dowolne inne obiekty, o których piszemy poniżej.

14.1.2 Formaty grafiki WWW

Grafika rastrowa

Przeładowarki kiepsko skalują grafikę, dlatego najlepiej zamieszczać na stronie grafiki uprzednio przygotowane (w dobrym edytorze plików graficznych, np. GIMP) tak, by nasza grafika miała od razu właściwe rozmiary (w pikselach) i można było zamieszczać ją na stronie bez potrzeby dodatkowego skalowania.

Równie ważny jak dobór rozmiarów obrazka, jest wybranie odpowiedniego do jego zawartości formatu.

Obecnie najbardziej popularne są trzy formaty plików graficznych, które można zamieszczać na stronach WWW. Każdy z nich jest doskonale obsługiwany przez wszystkie wiodące przeglądarki internetowe.

JPEG Doskonale nadaje się do prezentacji zdjęć, dzięki dużej kompresji pliki w tym formacie mają małą objętość (a więc strony szybciej się ładują). Jako format, w którym zawartość grafiki jest poddana kompresji stratnej, niezbyt nadaje się do prezentowania wykresów, schematów, obrazków z dużą ilością tekstu, bo elementy o ostrych krawędziach wydają się w nim zazwyczaj trochę rozmazane.

PNG Format kompresji bezstratnej, doskonale nadaje się do prezentacji wykresów, obrazków zawierających teksty (np. skanów z rękopisu) itp.

GIF Starszy format o podobnych właściwościach, co PNG.

Wszystkie powyższe formaty mają z punktu widzenia filozofii HTML poważną wadę, mianowicie grafika w nich ma na sztywno zakodowany rozmiar obrazu lub jego rozdzielczość: są to obrazy *rastrowe*. Powoduje to wiele kłopotów, m.in. podczas powiększania czcionki oglądanego dokumentu (obrazki nie skalują się proporcjonalnie (zob. str 297) i cały dokument zaczyna dziwnie wyglądać), podobnie wiele rozczarowań może nas czekać podczas wydruku stron WWW zawierających grafikę. Wreszcie, nawet tak banalna rzecz jak zmiana rozdzielczości monitora na większą może spowodować, że obrazki zamieszczone na stronie będą małe i nieczytelne.

Grafika wektorowa

Takich ograniczeń nie mają grafiki wektorowe. Dlatego od niedawna coraz większą popularność zdobywa sobie więc wektorowy sposób prezentowania grafiki na stronach HTML: SVG (ang. *Scalable Vector Graphics*). Co ciekawe, choć po zastanowieniu zrozumiałe, pliki SVG są plikami tekstowymi¹, zawierającymi informację o tym, w jaki sposób *skonstruować* obrazek zapisany w tym formacie. Grafika w formacie SVG wymaga obecnie zazwyczaj²

¹Zapisanymi z wykorzystaniem języka znaczników XML.

²Najnowszy Firefox ma już wbudowaną obsługę plików SVG.

instalacji dodatkowej — darmowej — wtyczki do przeglądarki, a tworzenie grafik w tym formacie jest ograniczone do kilku aplikacji. Jedną z nich jest linuxowy program Sodipodi (zob. <http://www.sodipodi.com>).

14.1.3 Nagrania dźwiękowe i filmy

W dobie powszechnej dostępności aparatów cyfrowych, kamer internetowych i komputerów z obsługą multimediiów (np. mikrofonu), łatwo jest stworzyć pliki zawierające materiały filmowe lub dźwiękowe. W specyficznych sytuacjach można wzbogacić zawartość naszych stron internetowych o tego typu obiekty.

Generalnie, najlepszym rozwiązaniem jest po prostu umieszczenie na stronie linku, prowadzącego wprost dożądanego klipu, jak tutaj:

```
<a href="http://anon.nasa-global.speedera.net/neptune.mov" >  
Zobacz film NASA o Neptunie!  
</a> (Rozmiar pliku: 0.5 MB)
```

Po kliknięciu, otwiera się okno właściwego odtwarzacza multimediiów i wszyscy są zadowoleni. Zwróć uwagę, że link uzupełniono informacją o wielkości pliku wideo: zazwyczaj takie pliki są duże (a nawet: *bardzo duże*), dlatego informacja o rozmiarze pliku pomoże czytelnikowi podjąć decyzję, czy warto w ogóle taki plik ściągać.

Oczywiście, nie jest to prawdziwe *zanurzenie* multimediiów *w stronie* (plik multimedialny jest na zewnątrz strony), ale takie rozwiązanie ma wiele zalet:

1. Znakomita większość przeglądarek dobrze sobie z tym radzi.
2. Czytelnik ściąga (zazwyczaj dużego rozmiaru) plik audio/wideo wyłącznie wtedy, gdy tego zechce, a decyzję podejmie na podstawie opisu. (Do dobrego tonu należy podanie przybliżonego rozmiaru tego pliku, tak jak uczyniono w przykładzie powyżej.)

14.1.4 Kilka słów o dobrych obyczajach

Na swojej stronie WWW można także zamieścić obrazek, którego oryginał znajduje się na cudzej stronie WWW. Warto pamiętać, że wcale nie wymaga to uprzedniego skopiowania tego obrazka na swój komputer. Ba, takie kopiowanie cudzej grafiki w celu udostępnienia jej w swoich dokumentach jest źle widziane w społeczności internetowej. Wiąże się to z interpretacją aktu zamieszczenia obrazka (czy jakiegokolwiek innego dokumentu) przez jego autora na stronie WWW:

- Skoro autor zamieszcza coś na swojej stronie WWW, to znaczy, że chce, by to było publicznie dostępne

- Jeśli nie wyrazi innej woli, chce, by było wiadome, że on jest autorem zamieszczanych dokumentów
- Autor dopuszcza i akceptuje, że każdy może zrobić *na własny użytek* prywatną kopię tego dokumentu i przechowywać ją po wsze czasy
- Usunięcie dokumentu z jego strony WWW jest **równoznaczne** z cofnięciem zgody autora na to, by ten dokument był *publicznie* dostępny

Dlatego, korzystając z cudzych źródeł (np. grafiki) zawsze podawajmy jej autora (oraz link do oryginału) i zamieszczajmy ją poprzez wklejenie jej do naszej strony wprost ze strony źródłowej:

```
<div>

<br>
<cite>Źródło: <a href="http://www.leonardo.com">Galeria Leonarda da
Vinci</a></cite>
</div>
```

Dzięki temu, gdy obrazek zostanie przez jego autora usunięty z serwisu WWW, przestanie się także pojawiać na naszych stronach!

Uwaga. Powyżej przedstawiono jedynie zwyczaje panujące w społeczności internetowej. Należy jednak bezwzględnie pamiętać o tym, że od strony prawnej zagadnienia prawa autorskiego są regulowane ustawą o prawie autorskim i prawach pokrewnych z 4 lutego 1994 r. i z późniejszymi zmianami. Wyjaśnienie tych kwestii wykracza poza zakres niniejszego kursu.

Ćwiczenie 14.1.1. Uzupełnij własną stronę `index.html` o Twoje zdjęcie. Najlepiej zamieść je po prawej stronie dokumentu, oblewane z lewej strony tekstem. Pamiętaj, aby przed publikacją poddać zdjęcie edycji za pomocą dobrego programu graficznego (np. GIMP) tak, by dopasować jego format i rozdzielczość do wymogów prezentacji na stronie WWW.

Ćwiczenie 14.1.2. Zrób stronę `kosmos.html`, na której opowiesz o wrażeniach, jakie wywarły na Tobie zdjęcia wybranych ciał niebieskich wykonane teleskopem Hubble'a (zob. <http://hubblesite.org>). Oczywiście, aby nie być gołosłownym, wklej na stronie zdjęcia obiektów, o których piszesz. Pamiętaj, by nie robić kopii tych zdjęć na swoim komputerze!

14.1.5 Zanurzanie multimediiów w naszej stronie WWW

Ten podrozdział nie jest obowiązkowy.

Inne, nietypowe obiekty (nagrania dźwiękowe, filmy, aplety w Javie, itp.) a także już nam znane: grafikę, a także inne strony WWW, możemy zamieszczać wprost w naszym dokumencie przy użyciu znacznika `<object>`. Niestety, ten — zgodny ze standardem HTML 4.01 (zob. <http://www.w3>.

org/TR/html4/struct/objects.html) znacznik — jest bardzo różnie obsługiwany przez różne przeglądarki. W szczególności, Microsoft Internet Explorer bardzo kapryśnie obsługuje ten znacznik!

Grafika jako <object>

Ze względu na to, że Microsoft Internet Explorer ma kłopoty z obsługiwaniem grafiki wklejanej w ten sposób, a znacznik jest obsługiwany bezproblemowo i też jest elementem standardu HTML 4.01 — nie będziemy go tu omawiać.

Inna strona WWW jako <object>

```
<object data="technikalia.html" type="text/html"
  width="80%" height="30%" standby="Ładuję obiekt...Czekaj..." >
[Nie udało się wkleić <a href="technikalia.html">żądanego obiektu</a>.]
</object>
```

Zawartość znacznika <object> jest wyświetlana tylko wtedy, gdy wklejany obiekt nie może być wyświetlony. Tą zawartością może być np. drugi znacznik <object>, z czego skorzystamy poniżej.

Musimy pamiętać, że Microsoft Internet Explorer wymaga podania rozmiarów width i height obiektu; w przeciwnym razie nie będzie go poprawnie wyświetlać. Innym przeglądarkom wymiary oczywiście nie zaszkodzą. Poza tym, ze względu na to, że Microsoft Internet Explorer (wersja 6.0 SP1) nie obsługuje prawidłowo znacznika <object> i strony zawierające poprawnie wklejone pliki będą czasem nie najlepiej wyglądać.

Klipy audio/wideo jako <object>

Ponieważ jest to wciąż stosunkowo nowa technologia w HTML, nie wszystko tu jeszcze działa tak, jak spodziewalibyśmy się. Omówimy dwie najważniejsze metody, zgodne ze standardem HTML 4.01.

Zanurzenie klipu na stronie powoduje wyświetlenie na niej mini-odtwarzacza, w którym ukazuje się nagranie. Czytelnik strony może, korzystając z przycisków panelu sterującego odtwarzacza, przewijać nagranie, zatrzymywać je, itd.

Dawniej uzyskiwano taki efekt przy użyciu **niezgodnego ze standardem** znacznika <embed>, i — o dziwo — znakomita większość przeglądarek całkiem nieźle go obsługuje. Nieobecność tego znacznika w jakimkolwiek standardzie HTML powoduje, że nie powinniśmy go używać, chcąc jednocześnie korzystać z usług Validator HTML. Jego analogonem w standardzie HTML 4.01 jest znacznik <object>. Niestety, jak już stwierdziliśmy, Microsoft Internet Explorer nie radzi sobie z prawidłową obsługą znacznika <object>.

Ponieważ stoimy na stanowisku, że kodu HTML niezgodnego ze standardem nie powinniśmy zamieszczać na swoich stronach WWW, przedstawimy

Pamiętajmy jednak, że dla dostępu do zawartości multimedialnej, konieczne jest, by na naszym komputerze były dostępne odpowiednie wtyczki dla przeglądarki.

poniżej rozwiązanie kompromisowe: zgodne ze standardem i dające szansę właściwego włączenia obiektu pod Windows. Niestety, będzie ono bardziej skomplikowane, a to ze względu na dziwaczne preferencje przeglądarki Microsoft Internet Explorer. Zastosowana przez nas metoda da w pełni satysfakcjonujący efekt w wielu popularnych przeglądarkach: Opera, Firefox/Mozilla, Konqueror. Niestety, w przeglądarce Microsoft Internet Explorer wynik będzie tylko w 50% zadowalający...

Dla zapewnienia jednolitej formy, przyjęliśmy, że w środowisku Microsoft Windows będziemy używać wtyczki Apple QuickTime (zob. <http://www.apple.com/qt>) w celu pokazania zawartości multimedialnej.

Zanurzenie pliku wideo (.mov) w stronie WWW Nasz przykład będzie dotyczyć klipu wideo w popularnym formacie (m.in. wykorzystywanym w niektórych aparatach cyfrowych) QuickTime (zob. <http://www.apple.com/qt>). Oto jak go poprawnie zanurzyć w naszej stronie WWW.

```
<p>
  <object
    type="video/quicktime"
    width="320" height="260"
    classid="clsid:02BF25D5-8C17-4B23-BC80-D3488ABDDC6B"
    codebase="http://www.apple.com/qtactivex/qtplugin.cab" >
    <param name="src"
      value="http://anon.nasa-global.speedera.net/anon.nasa-global/HST_neptune/320.mov" >
    <param name="controller" value="true" >
    <param name="autoplay" value="0" >

  <object
    data="http://anon.nasa-global.speedera.net/anon.nasa-global/HST_neptune/320.mov"
    type="video/quicktime"
    width="320" height="260" >

    <param name="controller" value="true" />
    <param name="showdisplay" value="true" />
    <param name="autoplay" value="0" />
    <param name="autostart" value="0" >
    [ <i>Nie udało się wkleić
      <a
        href="http://anon.nasa-global.speedera.net/anon.nasa-global/HST_neptune/320.mov" >
        żądanego obiektu</a>.</i>]
  </object>
</object>
</p>
```

Wyróżniony kolorem znacznik **<object>** ma na celu jedynie zaspokojenie przeglądarki Internet Explorer, omówimy więc najpierw zanurzony w nim drugi znacznik **<object>** zrozumiały dla pozostałych przeglądarek.

Sens pierwszych pięciu parametrów drugiego znacznika **<object>** wydaje się zrozumiały: podajemy informację o tym, skąd pobrać plik wideo, następnie, w jakim jest on formacie oraz informację o rozmiarze obrazu (w

pikselach). Co prawda sam film ma klatkę wymiaru 320 × 240 pikseli, ale podajemy wysokość 260 pikseli, aby zmieścił się także panel sterowania. Parametry znaczników `<param>` zawierają dodatkowe instrukcje dla przeglądarki, jak zaprezentować nasze wideo: `<param name="controller" value="true">` poleca oprócz samego wideo zamieścić także mały panel sterowania odtwarzaniem (przyciski przewijania filmu, stopu, itp.), natomiast `<param name="autoplay" value="0">` rezygnuje z automatycznego włączenia odtwarzania wideo, co z pewnością docenią czytelnicy strony.

Jeśli zaś chodzi o zewnętrzny znacznik `<object>` — napisany specjalnie dla Internet Explorera.

- `classid="clsid:02BF25D5-8C17-4B23-BC80-D3488ABDDC6B"` Jest to konieczny identyfikator (oczywiście, identyfikator mający sens jedynie w systemie Windows!) aplikacji odpowiedzialnej za wyświetlenie obiektu — tutaj: QuickTime.
- `codebase="http://www.apple.com/qtactivex/qtplugin.cab"` Jest to opcjonalna informacja, skąd ściągnąć stosowną wtyczkę do przeglądarki Microsoft Internet Explorerze, o ile nie jest jeszcze zainstalowana w systemie.
- `<param name="src" value="http://anon.nasa-global.speedera.net/neptune.mov">`

Tak, to konieczne w Microsoft Internet Explorerze...

Tak skonstruowana strona jest w 100% zgodna ze standardem HTML i będzie doskonale działać m.in. w przeglądarkach Mozilla/Firefox i Opera, zarówno pod Windows, jak Linuxem. Niestety, Microsoft Internet Explorer w wielu wypadkach będzie ją dalej niepoprawnie wyświetlać, a dokładniej, będzie próbował wyświetlić (wbrew ustaleniom standardu HTML!) zawartość obu znaczników `<object>`. Oczywiście, zakończy się to wyświetleniem obok siebie dwóch okienek, przy czym drugie będzie puste.

Jak to zrobić, by również Microsoft Internet Explorer pokazał tylko jedno okienko (to właściwe) — można przeczytać na stronach WWW (zob. <http://realdev1.realise.com/rossa/rendertest/quicktime.html>). Nie omawiamy tutaj szczegółów — wystarczy, jeśli powiemy, że rozwiązanie opiera się na *innych błędach* w Microsoft Internet Explorerze!

Więcej na temat wklejania wideo *tak, by prawidłowo wyświetlał je Microsoft Internet Explorer* i o związanych z tym kłopotach, można przeczytać na stronach WWW:

1. „Adding Windows Media to Web Pages” (zob. http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwm/html/adding_windows_media_to_web_pages__etse.asp) — oryginalne materiały Microsoftu, czyli jak to zrobić tak, by stronę oglądać (wyłącznie) spod Microsoft Internet Explorerze w Windows (inne przeglądarki nie interesują autorów)

2. „Free tutorial HTML 4.01 : embedding objects part 5 — using the object element” (zob. <http://tutorials.findtutorials.com/read/id/318/p/3>) — czyli rzut oka z ukosa na nurtujące nas problemy.
3. „Robin’s HTML 4.0 Test of Objects: Video” (zob. <http://www.robinlionheart.com/stds/html4/objvideo>) — czyli sprawdzenie zgodności naszej przeglądarki z protokołem HTML 4.0 w części dotyczącej obiektów wideo.

Zanurzenie pliku audio (.mp3) w stronie WWW Przebrnąwszy przez zanurzanie wideo, zanurzenie pliku dźwiękowego zapisanego w popularnym formacie mp3 wydaje się dziecinnie łatwe.

```
<p>
<object
  width="200" height="50"
  type="audio/mpeg"
  classid="clsid:02BF25D5-8C17-4B23-BC80-D3488ABDDC6B"
  codebase="http://www.apple.com/qtactivex/qtplugin.cab">

  <param name="src" value="demo.mp3">
  <param name="autostart" value="0">
  <param name="controller" value="true">

  <object
    type="audio/mpeg"
    data="demo.mp3"
    width="200" height="50">
    <param name="autostart" value="0">
    <param name="controller" value="true">

    Ściągnij plik <a href="demo.mp3">demo.mp3</a>
  </object>
</object>
</p>
```

Zewnętrzny znacznik `<object>` jest ponownie tak spreparowany, by zadowolić przeglądarkę Microsoft Internet Explorer. Wszystkie inne przeglądarki, o ile nie rozumieją tak spreparowanego obiektu, spróbują skorzystać z drugiego, zanurzonego w nim. Ponieważ drugi znacznik jest zapisany w standardowy sposób, „normalne” przeglądarki powinny go bez trudu obsłużyć. Wymiary obiektu będą oczywiście jedynie rezerwować miejsce na mini-odtwarzacz. Wklejając pliki audio pamiętajmy, by podając `<param name="autostart" value="0">` zablokować automatyczne odtwarzanie dźwięków po otwarciu strony. Czytelnicy będą głęboko wdzięczni!

Tak skonstruowana strona jest w 100% zgodna ze standardem HTML i będzie doskonale działać m.in. w przeglądarkach Mozilla/Firefox i Opera, zarówno pod Windows, jak Linuxem. Niestety, Microsoft Internet Explorer w wielu wypadkach będzie ją dalej niepoprawnie wyświetlać, a dokładniej, będzie próbować wyświetlić (wbrew standardowi!) zawartość *obu* znaczników `<object>`. Oczywiście, zakończy się to wyświetleniem obok siebie dwóch okienek, przy czym najczęściej drugie będzie puste. Jak to zrobić,

by również Microsoft Internet Explorer pokazał tylko jedno okienko (to właściwe) — można przeczytać na stronach WWW (zob. <http://realdev1.realise.com/rossa/rendertest/quicktime.html>). Nie omawiamy tutaj szczegółów — wystarczy, jeśli powiemy, że rozwiązanie opiera się na innych błędach w Microsoft Internet Explorerze!

Aplet w Javie

Java (zob. <http://java.sun.com>) jest językiem programowania wysokiego poziomu, wymyślonym dla *wirtualnej* maszyny. Programy w Javie są dlatego w 100% przenośne między najdziwniejszymi architekturami, od telefonów komórkowych po superkomputery, gdyż „komputer” na którym działają aplikacje Javy istnieje jedynie jako software!

Wszystkie nowoczesne przeglądarki obsługują Javę, umożliwiając zamieszczanie małych interakcyjnych programików — apletów — na naszej stronie WWW. Oczywiście, aby taki programik zamieścić, trzeba go wcześniej samemu napisać, a wymaga to nauki programowania w języku Java. Aby prawidłowo pokazać aplet, przeglądarki czasem wymagają zainstalowania dodatkowego oprogramowania — owego wirtualnego komputera. Jakkolwiek specyfikacja HTML 4.01 (zob. <http://www.w3.org/TR/html4/struct/objects.html>) twierdzi, że zalecaną metodą włączania apletów w Javie do stron HTML jest znacznik `<object>`, na razie dla zamieszczania apletów na stronach WWW lepiej³ jest używać starego, ale sprawdzonego (i wciąż zgodnego ze standardem) znacznika `<applet>`. Wtedy jednak musimy zadeklarować na samym początku, że używamy standardu HTML 4.01 Transitional, a nie Strict:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD_HTML_4.01_Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd" >
<html>
  <head>
    ...
  </head>
  <body>
    <p style="text-align:center;" >

    <applet code="ReactionDiffusion.class"
      codebase="http://www.design.chalmers.se/ReactionDiffusion"
      width="200" height="250"
      alt="Równanie_reakcji-dyfuzji" >
    <param name="xmax" value="100" >
    <param name="ymax" value="100" >
    aplet modelujący zjawisko reakcji-dyfuzji.
    </applet>

    Autor:
    <a href="http://www.design.chalmers.se/reactiondiffusion.html" >
    Joakim Linde</a>
```

Stosując znacznik `<applet>`, musimy wskazać w DOCTYPE standard HTML 4.01 Transitional!

³Zobacz rozdział 14.1.5.

```
....  
</body>  
</html>
```

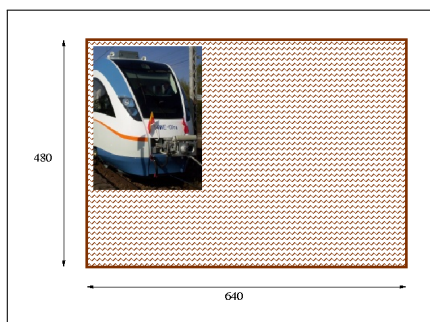
Tworzenie dobrych apletów w Javie jest sztuką w sobie i wykracza poza ramy niniejszego kursu. Na szczęście(?) jest wiele miejsc w Internecie, skąd można ściągnąć bardziej lub (zwłaszcza!) mniej przydatne gotowe i darmowe aplety w Javie. Ze względów bezpieczeństwa, należy jednak ostrożnie podchodzić do *kopiowania* "darmowych" apletów na własny komputer i zamieszczania ich jako własne. Jeśli ktoś jest zainteresowany bezpieczeństwem apletów w Javie, warto przeczytać artykuł Java Security Evolution and Concepts, Applet Security (zob. <http://java.sun.com/developer/technicalArticles/Security/applets>) oraz kilka wyjaśnień na ten temat (zob. <http://java.sun.com/sfaq/>).

Zamieszczając cudze aplety na własnej stronie, koniecznie sprawdzimy, czy ich autor wyraża na to zgodę i w jakim zakresie. Warto też podać informację o autorze, jak uczyniono powyżej.

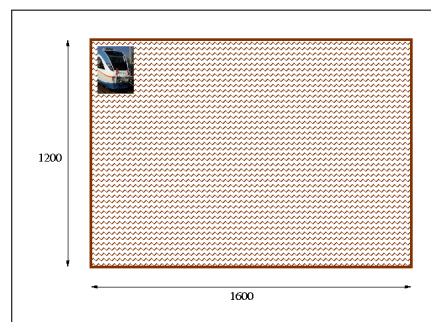
14.1.6 Wielkość *ma* znaczenie

Należy pamiętać o tym, by unikać zamieszczania zbyt wielkich plików multimedialnych (przykładowo, max. 100kB na *duży* obrazek), wybierając właściwy poziom rozdzielczości a także stopień i sposób kompresji:

Zdjęcia Najwłaściwszy jest format JPEG; zdjęcie o rozdzielczości 640x480 będzie wyraźne nawet przy powiększeniu na cały ekran, gdy chodzi o grafikę zanurzaną na stronie możemy używać znacznie mniejszych rozdzielczości. Rzeczywista wielkość obrazka o wymiarach np. 100 × 200 pikseli będzie inna na monitorze o rozdzielczości 800 × 600, a inna na 1600 × 1200!



Pewien obrazek rastrowy na ekranie o rozdzielczości 640 × 480



Ten sam obrazek na ekranie o rozdzielczości 1600 × 1200

Inne grafiki Tu na wszelki wypadek powinniśmy użyć formatu używającego kompresji bezstratnej: PNG lub GIF, kryterium doboru rozdzielczości identyczne jak w JPEG.

Nagrania dźwiękowe Obecnie najlepszym formatem dla udostępniania plików audio na stronach WWW jest popularny format mp3. Uwa-

ga: być może będzie wymagać od odbiorcy zainstalowania dodatkowej wtyczki do przeglądarki (zwłaszcza w Linuxie); zapis naszych nagrań z jakością CD i w stereo jest najczęściej przesadą, do zwykłych zastosowań na WWW wystarczy jakość dwukrotnie gorsza i mono.

Nagrania wideo — MPEG-4; uwaga: być może będzie wymagać od czytelnika zainstalowania dodatkowej wtyczki do przeglądarki, np. obsługującej DivX. Należy unikać stosowania mało znanych (np. najnowszych) — lub specyficznych dla jednego systemu operacyjnego — kodeków, gdyż odbiorca może nie mieć możliwości ich odczytania.

Przy tej okazji zwróćmy uwagę, że *lwią część* plików audio i wideo, które chcielibyśmy zamieścić na WWW, da się zastąpić i to (**z korzyścią dla czytelnika!**) plikami tekstowymi, być może uzupełnionymi kilkoma starannie dobranymi grafikami/zdjęciami.

Tworzenie dobrych grafik, filmów, czy nagrań dźwiękowych (także na potrzeby WWW) jest sztuką w sobie i oczywiście wykracza poza ramy niniejszego kursu.

Pamiętajmy wszelako, zanurzając multimedia wprost na stronie WWW, że taka strona, która nie dość, że długo się ładuje, to jeszcze przy każdym obejrzeniu wydaje dźwięki (bo uruchamia się odtwarzanie ścieżki dźwiękowej ściągniętego pliku) — jest *wyjątkowo* denerwująca!

To, jakie obiekty mogą zostać skutecznie wklejone do naszej strony, zależy od ustawień przeglądarki i możliwości systemu *czytelnika*. Nie jest więc w dobrym tonie zamieszczanie multimediiów o których wiadomo, że działają wyłącznie na jednej platformie (np. Windows), albo takich, które wymagają nietypowych wtyczek dla przeglądarki. Pamiętajmy, publikujemy dla czytelnika (a nie dla własnej przyjemności), musimy więc zadbać o to, by odbiorca nie miał kłopotu z odczytaniem naszych plików, w tym — multimediiów.

Dla użytkowników Windows: wielkość liter też *ma* znaczenie

Pamiętajmy, że serwery WWW rozróżniają wielkość liter w nazwach plików! Dlatego jeśli masz na stronie wpisane ``, a w rzeczywistości Twój plik nazywa się (Windows lubi czasem tak zrobić!) `zdjecie.JPG`, będziesz mieć problemy z zobaczeniem obrazka na stronie po publikacji na serwerze.

□ Ćwiczenie 14.1.3.

Na stronach zawierających zdjęcia z teleskopu Hubble'a (zob. <http://hubblesite.org/gallery/>) jest wiele zdjęć o bardzo dużej rozdzielczości i jeszcze większym rozmiarze (zwłaszcza te w formacie TIFF). Ściągnij na swój komputer jedno z takich *wielkich* zdjęć, np. zdjęcie galaktyki spiralnej z poprzeczką NGC 1300 (zob. http://imgsrc.hubblesite.org/hu/db/2005/01/images/a/formats/full_tif.TIFF) i poddaj obróbce w programie graficznym tak, by dobrze się prezentowało na typowej stronie WWW.

Wybierz właściwe: rozmiary obrazka w pikselach, format pliku oraz stopień kompresji. Pamiętaj, że w typowych sytuacjach (załóżmy, że Twoja jest typowa...) rozmiar pliku obrazka nie powinien przekraczać kilkunastu (maksymalnie: kilkudziesięciu) kilobajtów. □

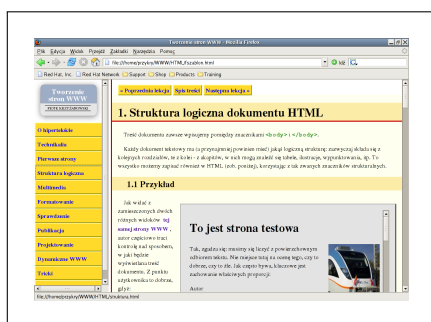
14.2 Formatowanie stron WWW

Wreszcie nadamy naszym stronom indywidualny charakter, odpowiednio formatując zapisaną **TREŚĆ**, uprzednio wzbogaconą strukturalnymi znacznikami HTML. Skorzystamy z technologii, która w prosty sposób umożliwi nam zachowanie jednolitego, charakterystycznego wyglądu dla wszystkich naszych stron WWW!

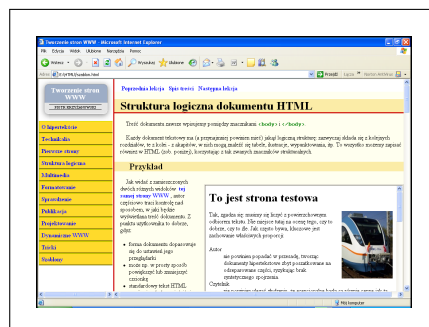
14.2.1 Potrzeba oddzielenia formy od treści

Jest wiele sposobów wpływania na wygląd stron WWW. My musimy wziąć pod uwagę kilka istotnych ograniczeń.

- Nie możemy spędzić pół życia na formatowaniu stron WWW, zatem nasze strony powinny mieć minimalistyczny *design* (co nie znaczy, że nieciekawym)
- Każda przeglądarka traktuje HTML i CSS odrobinę inaczej, a dokładniej: ma inny zestaw domyślnych ustawień formatowania znaczników. Zatem efekt będzie inny w Microsoft Internet Explorerze, a inny w Firefoxie, na przykład:



Widok pewnej strony w Firefoxie



Widok tej samej strony w Internet Explorerze

Zamiast próbować z tym walczyć⁴, lepiej to zaakceptować i ograniczyć formatowanie strony tylko do takich form, które są zgodne ze standardami, więc — przynajmniej w teorii, zob. obrazki powyżej — jednakowo dobrze (choć nie identycznie) będą wyglądać w wiodących przeglądarkach.

⁴Jeśli ktoś zamierza *profesjonalnie* zająć się *designem* stron WWW, oczywiście konieczne powinien zacząć z tym walczyć!

- Pamiętajmy, że „złośliwy” czytelnik może *wyłączyć w swojej przeglądarce wszystkie nasze zabiegi formatujące* i zastosować wyłącznie własne. Jest to oczywiście tylko ciekawostka, bo skoro sam chciał...
- W razie wątpliwości, jak coś sformatować, jest tylko jedna odpowiedź: **zgodnie ze standardem!**

Nowoczesne formatowanie stron WWW polega na tym, że zasady, według których będzie wyświetlana zawartość znaczników danego rodzaju definiujemy *niezależnie* od treści konkretnego dokumentu HTML. Dlatego właśnie tak ważne było aby, kodując tekst do formatu HTML, używać znaczników *właściwie identyfikujących* znaczenie tekstu, którego dotyczą.

Zasady formatowania są również bardzo proste: wystarczy zazwyczaj podać jedynie, jaki aspekt wyglądu znacznika (np. kolor czcionki) chcemy zmienić. Pozostałe cechy (np. krój pisma, tło, obramowanie, ...) przyjmą wartości domyślne.

Zapowiadany przez nas sposób formatowania stron HTML nosi nazwę arkuszy CSS, ang. *Cascading Style Sheets*. Arkusz CSS jest — podobnie jak plik HTML — zwykłym plikiem tekstowym. Taki plik zawiera *wyłącznie* informację o tym, jak należy zmienić domyślne formatowanie wybranych znaczników HTML.

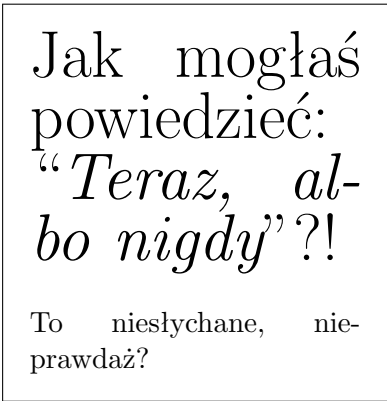
Reguły tworzenia CSS są precyzyjnie określone. Ich opis jest ostateczną instancją rozstrzygającą wątpliwości. W naszym kursie uczymy się standardu CSS 2.0 (zob. <http://www.w3c.org/TR/css2>). Został on ustanowiony przez konsorcjum W3C (zob. <http://w3c.org>), które powołano właśnie w celu opracowywania najróżniejszych standardów obowiązujących w sieci WWW.

Przypuśćmy, że w stylu definiującym sposób formatowania naszego pliku HTML podaliśmy jedynie, że zawartość znacznika `<p>` będzie zawsze wyświetlana czcionką wielkości 25 punktów:

```
p {
  font-size: 25pt;
}
```

Zobaczmy, jak wyświetli się poniższy fragment tekstu HTML:

```
<p>Jak mogłaś powiedzieć:
<q>Teraz, albo nigdy</q>?!
</p><div>To niesłychane,
nieprawdaz?</div>
```



A więc, mimo, że specyfikacja formatu dotyczyła tylko znacznika `<p>`, przyjęto rozsądnie, że dotyczyć będzie również wszystkich znaczników, które zostały umieszczone wewnątrz zawartości `<p>` — w tym także tekstu w znaczniku `<q>`. Taki sposób interpretacji reguł (formatowania) nazywa się *dziedziczeniem*. Dlatego tekst w znaczniku `<q>` został złożony czcionką tej samej wielkości, co reszta akapitu!⁵ Co więcej, zgodnie z naszymi oczekiwaniami, inne jego parametry przyjmą wartości domyślne dla `<q>`, co tutaj znaczy m.in., że tekst zostanie złożony kursywą.

14.2.2 Arkusz CSS

Są trzy sposoby, na jakie możemy określić reguły formatowania strony WWW za pomocą CSS:

Zewnętrzny plik CSS

Najwygodniej jest zapisać ogólne reguły formatowania (wybranych) znaczników w zewnętrznym pliku tekstowym, np. `styl.css`. Na przykład, taki plik mógłby mieć następującą zawartość (jej znaczenie stanie się jasne później):

```
strong {  
  color: red;  
}
```

Możemy domyślić się, że chodzi tu o to, by zawartość znacznika `` została dodatkowo wyświetlona na czerwono.

Następnie wystarczy podać w nagłówku dokumentu HTML, w jakim pliku znajduje się arkusz CSS według którego należy sformatować zawartość naszego dokumentu HTML:

```
<html>  
<head>  
  ....  
  <link href="styl.css" rel="stylesheet" >  
  ....  
</head>  
<body>  
  ....  
  <p>Każde silne wyróżnienie teraz <strong>zawsze będzie na  
  czerwono</strong>! Oczywiście, będzie — jak dotychczas — wyświetlone  
  także pogrubioną czcionką.</p>  
  ....  
</body>  
</html>
```

Wpis `<style>` w dokumencie

Ta sekcja nie jest obowiązkowa.

⁵Oczywiście, będzie się tak działo zawsze, pod warunkiem, że styl nie określa nigdzie *explicite* wielkości czcionki dla znacznika `<q>`.

Możliwe jest także wpisanie tych samych reguł wprost do nagłówka `<head>` pliku HTML, korzystając ze znacznika `<style>`, np.

```
<head>
...
<style type="text/css" >
  div {
    text-align: center;
    border-width: 1px;
    border-style: solid;
  }
</style>
...
</head>
```

Jest to rozwiązanie mniej elastyczne od poprzedniego, gdyż każdy plik ma własne reguły formatowania; jeśli byśmy więc chcieli ujednoczyć/zmienić te reguły dla wszystkich plików, musielibyśmy wszystkie po kolei zmienić.

Formatowanie pojedynczych znaczników HTML

Ta sekcja nie jest obowiązkowa.

Dodatkowo, w *wyjątkowych* sytuacjach możliwe jest sformułowanie zasad formatowania konkretnego znacznika występującego w treści gotowego dokumentu HTML, poprzez dodanie do znacznika parametru `style`, np.

<pre>A teraz zobaczycie... <strong style="background:_blue; _color:_yellow;_padding: _1em;" >niespodziankę! </pre>	<p>A teraz zobaczycie...</p> <p>niespodziankę!</p>
---	---

Jest to zdecydowanie najgorsze rozwiązanie, gdyż znacznie zwiększa objętość plików HTML, jest najbardziej żmudne w realizacji i bardzo kłopotliwe w przypadku wprowadzania jednolitych zmian w gotowym formacie.

Oprócz naprawdę wyjątkowych sytuacji, formatowanie ograniczymy wyłącznie do utworzenia jednego pliku, powiedzmy `styl.css`, w którym zapiszemy nasze ustawienia wyświetlania standardowych znaczników HTML. Z tego pliku będą korzystały *wszystkie* tworzone przez nas strony WWW, dzięki czemu:

- Wszystkie nasze strony WWW będą miały konsystentny, profesjonalny wygląd.
- Gdy zechcemy coś zmienić (upiększyć, poprawić, ...) w wyglądzie naszych stron, wystarczy, jeśli zmienimy jeden plik, w naszym przykładzie: `styl.css`; żadnych zmian w dokumentach HTML nie będziemy musieli wprowadzać!

- Będziemy mogli dostosować wygląd naszych dokumentów do tego, czy dokument jest oglądany w graficznej przeglądarce, czy też odsłuchiwany, albo np. drukowany!

Typowy dokument korzystający z formatowania zawartego w arkuszu CSS o nazwie styl.css będzie miał następujący nagłówek:

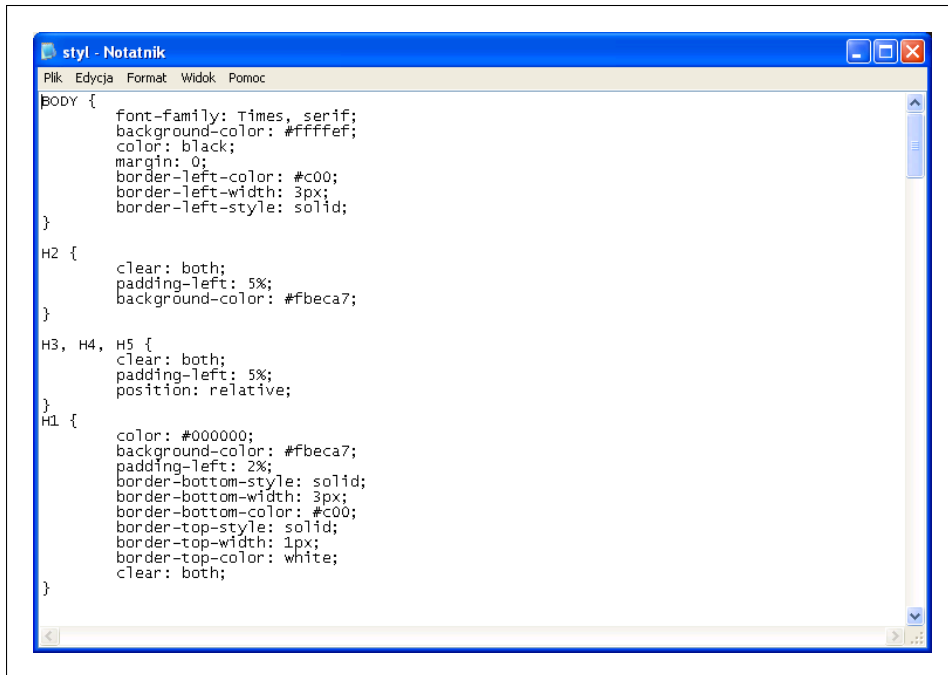
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD_HTML_4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd" >
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html;
    charset=iso-8859-2" >
  <title>Jan Kowalski – strona domowa</title>
  <meta name="Author" content="Jan_Kowalski" >
  <link href="styl.css" rel="stylesheet" >
</head>
<body>
  <h1>Jan Kowalski</h1>
  <p>Witajcie na mojej stronie internetowej.
  ....
</body>
</html>
```

Naturalnie, tak jak w pliku HTML możemy korzystać z grafik zamieszczonych na zewnętrznych serwerach WWW, tak samo możemy wykorzystać arkusze położone w innym miejscu niż nasz dokument HTML, np.

```
<head>
  ....
  <link href="http://www.mimuw.edu.pl/css/fresh.css" rel="stylesheet" >
  ....
</head>
```

Edytor plików CSS

Ponieważ pliki CSS to zwykle pliki tekstowe, do edycji plików CSS wybieramy ulubiony edytor tekstowy: Notatnik, nedit, itp.

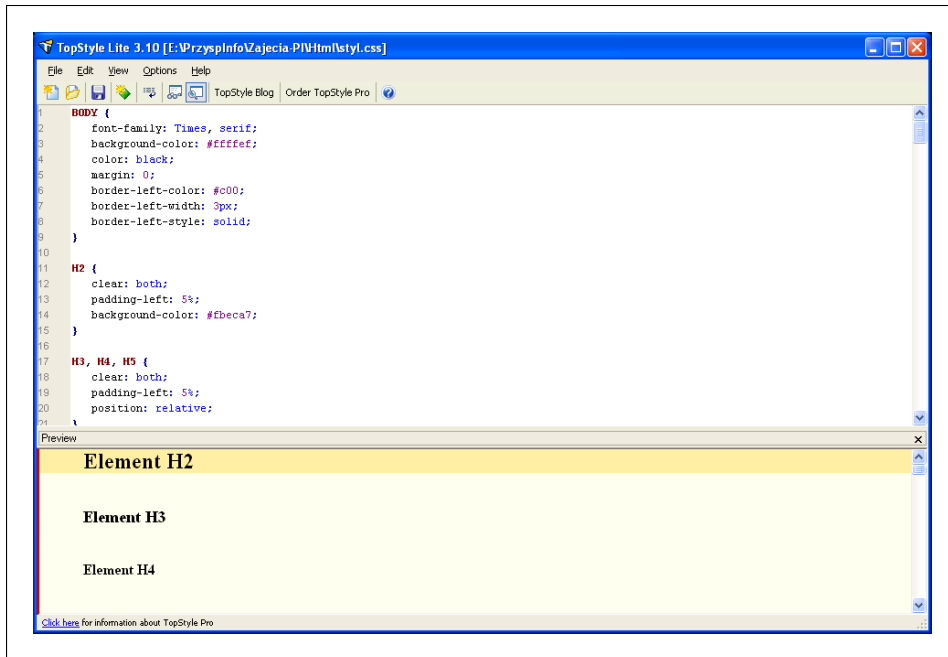


Edycja pliku zawierającego arkusz CSS w Notatniku Windows

Przykład 14.2.1. Jak stworzyć plik CSS w Notatniku pod WinXP(PL)

1. Otwieramy Notatnik (Start → Programy → Akcesoria → Notatnik)
2. Piszemy to, co chcemy napisać... (co konkretnie, dowiemy się w dalszej części podręcznika!)
3. Plik → Zapisz → Zapisz jako typ: Dokumenty tekstowe. Jako nazwę, podajemy nazwę z *rozszerzeniem* .css, np. styl.css
4. Gotowe!

Jak zwykle, można trafić także na mniej lub bardziej udane edytory dedykowane, np. darmowy TopStyleLite (zob. <http://www.bradsoft.com/topstyle/tslite/>) dla Windows, czy też cssed (zob. <http://cssed.sourceforge.net>) dla Linuxa i Windows.



Edycja pliku w programie TopStyleLite. Zwróć uwagę na kolorową składnię oraz możliwość podglądu wprowadzanych efektów.

Pliki CSS będziemy na szczęście edytować sporadycznie, nie ma więc większego sensu inwestycja w specjalny edytor wspomagający. Pamiętajmy, aby plikom CSS (np. tworzonym przy użyciu Notatnika) nadawać zawsze rozszerzenie `.css`!

14.2.3 Struktura arkusza CSS

Jak już zdążyliśmy zorientować się z powyższych przykładów, pliki CSS są plikami tekstowymi o następującej strukturze:

```
znacznik {
  jakaś-zmieniana-własność: wartość;
  inna-zmieniana-własność: inna-wartość;
  ...itd, kolejne własności...
}
```

... itd, kolejne znaczniki....

Pamiętaj o dwukropku po nazwie zmienianej własności i o średniku po nadawanej wartości!

Przykład 14.2.2.

Aby ustalić raz na zawsze, by znaczniki `` były wyświetlane pogrubioną i powiększoną czcionką w kolorze czerwonym na żółtym tle, dopisaliśmy w pliku CSS

```
strong {
  font-weight: bold;
  font-size: larger;
  color: red;
  background-color: yellow;
```

```
}
```

Oczywiście, zmiany w formatowaniu zostaną uwzględnione dopiero wtedy, gdy w nagłówku strony HTML wpiszesz poprawny znacznik `<link>`, zob. rozdział [14.2.2](#).

14.2.4 Klasy

Ten podrozdział nie jest obowiązkowy.

Prowadząc formatowanie przy użyciu arkuszy CSS szybko zorientujemy się, że w HTML jest *za mało* znaczników: bo jak poradzić sobie, gdy chcemy, by jedna część tekstu (np. formułka do zapamiętania) była otoczona czerwoną ramką z zielonym tłem, a inna (np. dygresja w większym tekście) — zieloną ramką z żółtym tłem. Do tego służą *klasy* znaczników. Przynależność konkretnego znacznika do danej klasy zaznaczamy podając dlań wartość parametru `class`, np.

```
<div class="formuła" >
    Pole prostokąta jest równe iloczynowi długości jego boków.
</div>

<div>
    Takiej formułki będziemy używać w dalszych rozważaniach.
</div>

<div class="ciekawostka" >
    Czy wiesz, że ta formuła jest podstawą do wprowadzenia pojęcia
    <em>całki</em> w matematyce wyższej?
</div>
```

Teraz podobne rozróżnienie będziemy mogli zastosować w arkuszu CSS:

```
div {
    margin-left: 2em;
}

div.formuła {
    border-style: solid;
    border-color: red;
    background-color: green;
}

div.dygresja {
    border-style: outset;
    border-color: green;
    background-color: yellow;
}
```

Jak widać, ustaliliśmy, żeby każdy znacznik `<div>` będzie miał lewostronny margines szerokości 2em (i nie będzie miał żadnych ramek, bo takie jest domyślne ustawienie znacznika `<div>`). Dodatkowo, zawartość znaczników należących do klasy `formuła` zostanie wyświetlona na zielonym tle i otoczone kolorową ramką. A co stanie się ze znacznikiem z klasy `ciekawostka`? — wszak

nie została wymieniona w arkuszu CSS! Odpowiedź jest prosta: skoro nie ma tej klasy w arkuszu CSS, zostanie użyte formatowanie domyślne, a więc takie, jak dla zwykłego znacznika `<div>`.

14.2.5 Wybrane parametry formatowania CSS

Jednostki miary stosowane w CSS

W arkuszach CSS (i w starszych wersjach HTML) mamy możliwość specyfikowania rozmiarów różnych elementów strony. Można to czynić na dwa sposoby, używając miar względnych, lub bezwzględnych.

Względne (preferowane, gdyż określają wielkość w stosunku do pewnych parametrów przeglądarki użytkownika, dzięki czemu części dokumentu zachowują wzajemne proporcje)

em — rozmiar podany jako wielokrotność bieżącego rozmiaru⁶ czcionki (tzn. własności `font-size`), np.

```
<p style="margin-left:.2.5em" >
```

sprawi, że tak oznaczony akapit będzie miał z lewej strony margines o szerokości 2.5 rozmiaru bieżącej czcionki

% — rozmiar podany w procentach wielkości elementu, w którym znajduje się znaczniki: okna, sekcji dokumentu, itp. — w zależności od kontekstu, np.

```
<table style="width:.80%" >
```

definiuje tabelę, której szerokość wynosi 80% szerokości strony.

Bezwzględne (niezalecane, bo powodują, że wygląd dokumentu, w tym proporcje pomiędzy elementami zależą m.in. od użytego sprzętu)

cm — rozmiar podawany w centymetrach; w dobrym przybliżeniu nie jest zależny od rozdzielczości; przeglądarki zazwyczaj dość sztywno trzymają się podanej wartości, co np. może nas mocno rozczarować podczas wydruku, np.

```
<img style="width:.12cm;" >
```

przeskaluje obrazek proporcjonalnie tak, by miał szerokość 12 cm.

pt — rozmiar podany w punktach (drukarskich jednostkach miary, tych samych które w edytorach WYSIWYG stosuje się do określania rozmiaru czcionki); często (nad)używany do określania rozmiaru czcionki, ma podobne wady, co rozmiar w centymetrach, np.

```
<body style="font-size:.12pt;" >
```

ustawi w całym dokumencie domyślną czcionkę wielkości 12 punktów.

⁶Rozmiar ten może zależeć, m.in. od ustawień przeglądarki, bądź od innych parametrów arkusza stylu CSS.

px — (zazwyczaj nie zalecane) rozmiar podawany w pikselach; najczęściej jest to najgorszy sposób wymiarowania elementów, gdyż ostateczny efekt zależy od użytej rozdzielczości monitora; warto ewentualnie stosować go tylko wtedy, kiedy dotyczy on szerokości ramek, podkreśleń, np.

```
<div style="width: 80%; border-style: solid; border-width: 1px;" >
```

choć trzeba pamiętać, że dla border mamy niezależne od sprzętu parametry thin, medium i thick.

Czasem wartości w pikselach używa się do przeskalowania obrazków zanurzanych w dokumentach HTML; jednak zalecanym postępowaniem jest przeskalowanie grafiki w dobrym programie graficznym do żądanych rozmiarów i zanurzenie jej bez skalowania!

Marginesy

Parametr margin określa szerokość marginesu, jaki ma otaczać z zewnątrz zawartość znacznika; domyślna jego wartość to zero.

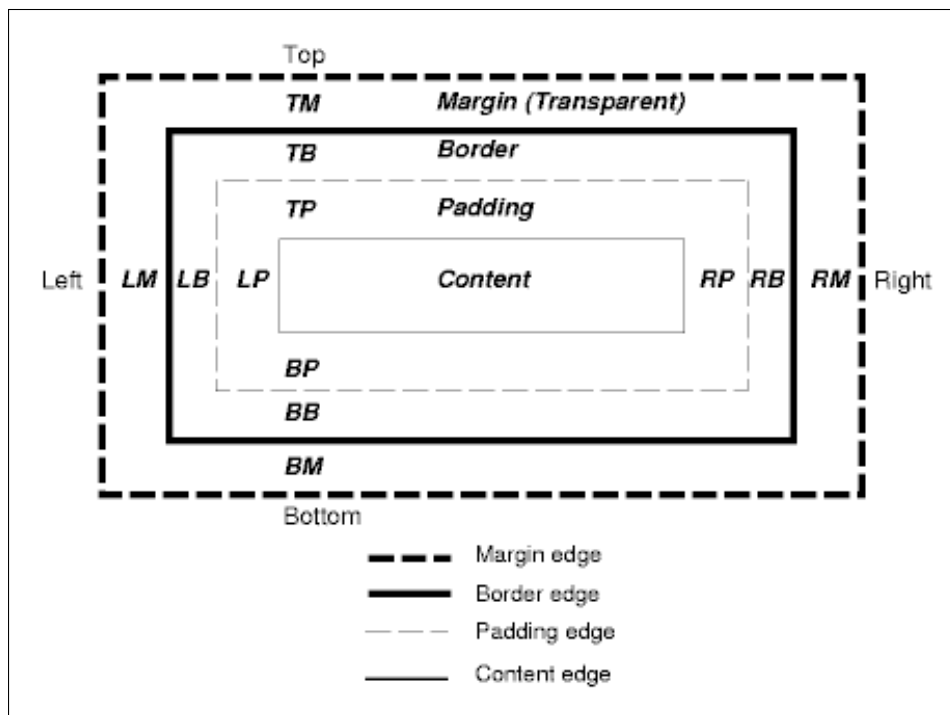
```
p {
  margin: 2em;
}
```

znaczy, że wszystkie akapity (to znaczy: zawartość wszystkich znaczników <p>...</p>) zostaną otoczone zewnętrznym marginesem wielkości podwójnego rozmiaru bieżącej czcionki (zob. rozdział 14.2.5).

Czasem zachodzi potrzeba utworzenia marginesu tylko z jednej strony (np. z lewej), bądź z kilku (np. z lewej i od góry). Wtedy można użyć ściślejszych nazw właściwości: margin-left, margin-right, margin-top, margin-bottom:

```
p {
  margin-top: 1em;
  margin-left: 2em;
}
```

Ta sama zasada stosuje się także do parametrów border i padding, o których poniżej.



Zależności pomiędzy margin, border i padding. Źródło: *Cascading Style Sheets, level 2.1*, <http://www.w3c.org/TR/css2> (zob. <http://www.w3c.org/TR/css2/box.html>). Copyright © World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved.

Obramowania

Jeśli chcemy wymusić obramowanie danego elementu HTML, musimy wyspecyfikować parametry ramki: jej rodzaj: `border-style`, szerokość `border-width` oraz kolor `border-color`.

- `border-style` — rodzaj ramki, m.in.
 - `none` — brak ramki (domyślna wartość)
 - `solid` — linia ciągła
 - `double` — linia podwójna
 - `dotted` — linia kropkowana
 - `dashed` — linia przerywana
 - `inset` — dająca złudzenie „wciśnięcia” zawartości ramki w powierzchnię
 - `outset` — dająca złudzenie „wystawiania” zawartości ramki nad powierzchnię

- `border-width` — szerokość obramowania. Można użyć wartości względnych, określanych przez przeglądarkę (np. w zależności od środowiska):
 - `thin` — cienka
 - `medium` — średnia (domyślnie, chyba, że `border-style` jest `none`, wtedy oczywiście jest równa zero)
 - `thick` — gruba
 lub wartości w jednostkach bezwzględnych (np. pikselach).
- `border-color` — Kolor obramowania. Domyślnie taki sam jak tekstu.

Poniższy kod CSS sprawi, że wszystkie znaczniki `<h1>` i `<h2>` będą miały podkreślenie grubą czerwoną kreską:

```
h1, h2 {
  border-bottom-style: solid;
  border-bottom-width: thick;
  border-bottom-color: red;
}
```

Odstępy

Tym razem chodzi o marginesy jakby *wewnątrz* znacznika (zob. rysunek na str. 344), a więc o odstęp pomiędzy otaczającą go hipotetyczną ramką a zawartością znacznika. Tą własnością steruje

- `padding`, na przykład

```
td, th {
  padding: 0.5em;
}
```

ustali, że zawartość każdej komórki tabeli będzie oddzielona odstępem wielkości *pół litery* — tzn. pół rozmiaru bieżącej czcionki — od krawędzi komórki.

Kolory

Dla wszystkich elementów HTML możemy także ustalić kolory: tekstu i tła.

Kolor tekstu Steruje nim parametr

`color` — może przyjąć jedną z predefiniowanych wartości `aqua`, `black`, `blue`, `fuchsia`, `gray`, `green`, `lime`, `maroon`, `navy`, `olive`, `purple`, `red`, `silver`, `teal`, `white`, `yellow` (oczywiście: angielskie nazwy wybranych kolorów).



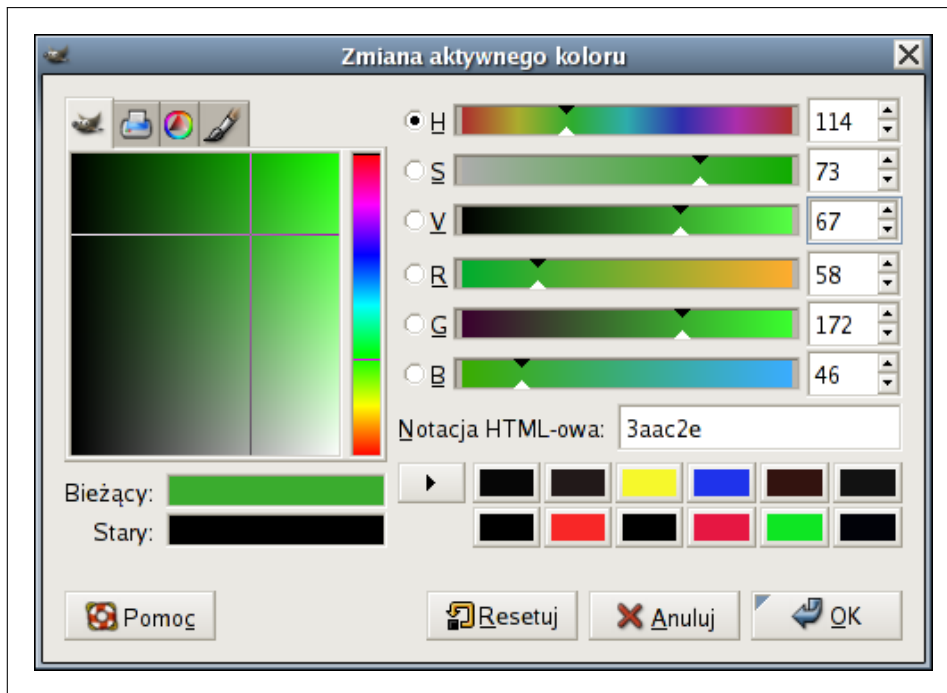
Kolory o predefiniowanych nazwach w CSS 2.1. Źródło: Cascading Style Sheets, level 2.1, <http://www.w3c.org/TR/css21> (zob. <http://www.w3.org/TR/CSS21/syndata.html>). Copyright © World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved.

```
strong {
  color: red;
}
```

Aby skorzystać z szerszej palety barw, nasz kolor musi zostać zakodowany jako wartość numeryczna zapisana w postaci szesnastkowej, opisująca barwę w przestrzeni kolorów podstawowych RGB (ang. *Red, Green, Blue*): np. kolorowi czerwonemu odpowiadać będzie numeryczna wartość #ff0000:

```
p {
  color: #ff0000;
}
```

Większość edytorów graficznych, a także edytorów webmasterskich, pozwala łatwo dopasować do wybranej barwy odpowiadający jej kod w postaci szesnastkowej (np. GIMP nazywa to wręcz „Notacją HTML-ową”):

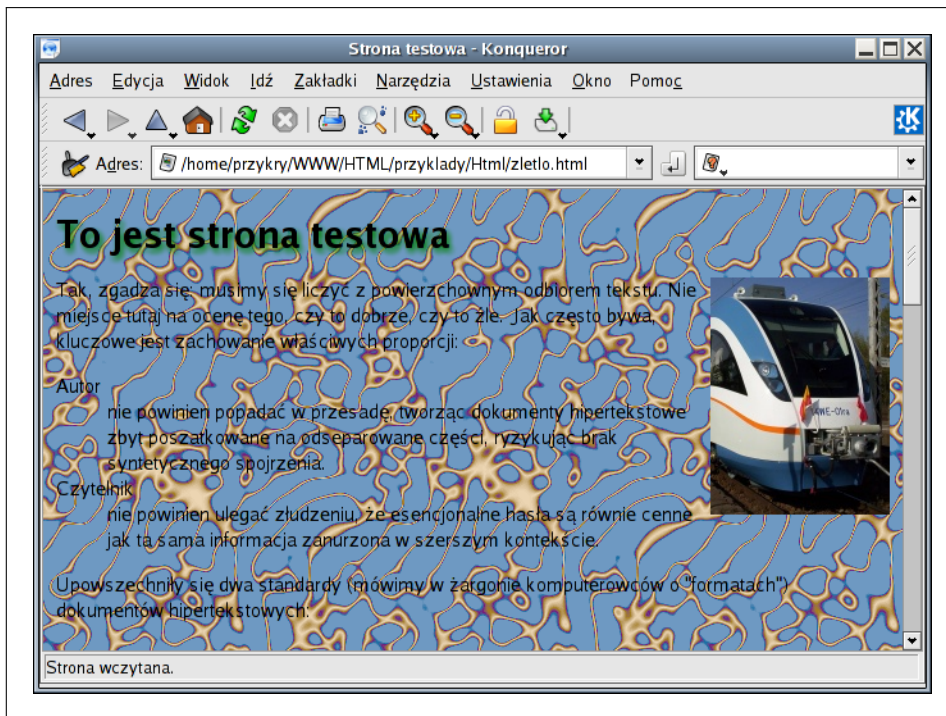


Okno dialogowe wyboru koloru w GIMPie. Dla wybranego koloru podana jest wartość w „Notacji HTML-owej”. Aby tekst akapitu miał właśnie taki kolor, napisalibyśmy p color: #3aac2e;

Należy wszakże pamiętać, że — w zależności od posiadanego przez czytelnika sprzętu — niektóre subtelnosci zdefiniowanych przez Ciebie kolorów mogą nie być prawidłowo wyświetlone.

Kolor tła Jest tu kilka możliwości, ale musimy pamiętać, że źle dobrane tło może skutecznie utrudnić percepcję treści strony, co widać na poniższym przykładzie:

Przykład 14.2.3. Źle dobrane tło strony WWW



Tło — owszem, piękne — ale... czy jesteś w stanie cokolwiek przeczytać?. Dla rozwiania wątpliwości: jest to ta sama testowa strona, co zwykle.

- `background-color` — kolor tła; takie same możliwości jak dla parametru `color`;

```
strong {
  color: red;
  background-color: yellow
}
```

- `background-image` — obrazek używany jako tło, np.

```
body {
  background-image: url("http://www.mojastrona.org.pl/kafel.jpg");
}
```

— używaj z wielkim umiarem!

- `background-repeat` — sposób powtarzania obrazka tła na całej stronie (zazwyczaj obrazek tła jest — i powinien być! — niewielkich rozmiarów). Możliwe wartości same się wyjaśniają:
 - `repeat-x` — powtarzaj w poziomie
 - `repeat-y` — powtarzaj w pionie
 - `repeat` — powtarzaj i w pionie i w poziomie (tak jak kafelki na podłodze)

Krój pisma

Ponadto, możemy dokładnie określić parametry użytego kroju pisma. Należy jednak pamiętać, że pożądane efekty będą osiągalne jedynie pod warunkiem, że czytelnik ma na swoim komputerze zainstalowane stosowne

czcionki — w przeciwnym razie przeglądarka wybierze coś w zamian, według swojego uznania.

- `font-family` — krój pisma. Musimy tutaj uważać, gdyż nie każda czcionka jest dostępna na dowolnym komputerze, dlatego zwykle najlepiej ograniczyć się do najbardziej podstawowych:

- serif (pismo szeryfowe)
- sans-serif (pismo bezszeryfowe)
- monospace: (pismo maszynowe, tzn. o stałej szerokości czcionki)

Oczywiście, można postulować konkretną czcionkę, np. Verdana, bądź Cosmic Zippy Zap, ale wtedy zawsze należy na końcu dodać, do której z w/w klas krojów pisma należy — na wypadek gdyby żądana konkretna czcionka nie była dostępna na komputerze odbiorcy:

```
h1 {  
  font-family: Verdana, sans-serif;  
}
```

— jeśli nie będzie dostępna Verdana (która jest czcionką bezszeryfową, zazwyczaj miłą dla oka, gdy oglądana na ekranie komputera), do wyświetlenia znacznika zostanie użyta domyślna czcionka bezszeryfowa: efekt nie będzie *identyczny* z zamierzonym, ale wciąż całkiem zadowalający.

- `font-size` — wielkość czcionki. Jest wiele sposobów określenia tej wartości: względem bieżącej wielkości (zalecane): `larger` lub `smaller`;

```
a {  
  font-size: larger;  
}
```

względem domyślnej wielkości czcionki „średniej wielkości”: `xx-small`, `x-small`, `small`, `medium`, `large`, `x-large`, `xx-large`:

```
h7 {  
  font-size: xx-small;  
}
```

Wreszcie, co najmniej zalecamy (przynajmniej w typowych sytuacjach), można zdefiniować czcionkę tak, jak robi się to w edytorach WYSIWYG, czyli w punktach:

```
h7 {  
  font-size: 8pt;  
}
```

- `font-style` — pochylenie czcionki: `normal`, `italic`, `oblique`
- `font-variant` — wariant czcionki: `normal`, `small-caps`

```
strong.uwaga {  
  font-variant: small-caps;  
}
```


- `font-weight` — wytłuszczenie normal, bold, oraz zwiększenie wytłuszczenia bolder lub jego zmniejszenie lighter

```
em.uwaga {
  font-weight: bolder;
}
```

Prezentacja tekstu

CSS daje nam także możliwość sterowania bardziej ogólnymi parametrami dotyczącymi sposobu prezentacji tekstu: liter, odstępów, słów, akapitów.

- `text-indent` — wcięcie pierwszego wiersza tekstu (nie polecam stosowania tego parametru)

- `text-align` — sposób wyrównania: left, right, center, justify

Na przykład, możemy zdefiniować styl dla znacznika `<p>` klasy center,

```
p.center {
  text-align: center;}

```

a potem zastosować go w tekście HTML, uzyskując akapit wyśrodkowany na stronie

```
<p class="center">Bum!</p>
```

Bum!

- `text-decoration` — dodatkowe efekty w tekście; należy stosować z wielkim umiarem!

none (brak ozdobników), underline⁷, overline, line-through, blink

Poniżej pokazujemy w jaki sposób uniknąć podkreślenia linków na stronie, co zazwyczaj daje ładny efekt:

```
a {
  text-decoration: none;
}
```

- `text-transform` — zmiana prezentacji tekstu w stosunku do tego, co zostało *faktycznie* zapisane w tekście: capitalize, uppercase, lowercase, none

```
<span style="text-transform:
  _uppercase;">To jeST
  DZiwne</span>
```

TO JEST DZIWNIE

⁷Nigdy nie stosuj `underline` na swoich stronach oprócz ewentualnego oznaczenia linków! Czytelnicy są tak przyzwyczajeni do tego, iż linki są podkreślane, że gdy widzą podkreślony tekst, są pewni, że kryje się pod nim jakiś link!

14.2.6 Kilka przykładów praktycznych zastosowań CSS

Zanim przejdziemy do przykładów, uwaga praktyczna. W sytuacji wyjątkowej, gdy chcemy aby zawartość jednego-jedynego znacznika miała odmienne od całości formatowanie, nie warto zamieszczać tego we wspólnym pliku CSS; lepiej wtedy zapisać wprost w pliku HTML, na przykład

```
<strong style="color:_green;_font-size:_larger;_text-transform:_uppercase;" >Za trzy dni biorę ślub!!! </strong>
```

ZA TRZY DNI BIORĘ ŚLUB!!!

W sytuacji *jeszcze bardziej wyjątkowej*, gdy nie bardzo wiemy, jakiego logicznego znacznika użyć w celu specyficznego sformatowania małego fragmentu tekstu, możemy użyć znacznika ``

```
<p>Mała przewodniczka <span style="color:_red;" >nie szkodzi</span>
```

Mała przewodniczka nie szkodzi

A teraz czas na przykłady.

Przykład 14.2.4. Efekt ramki

Aby uzyskać efekt ramki dla sekcji `<div class="ramka">`, w pliku CSS `styl.css` zapisałem

```
div.ramka {
  background-color: yellow;
  margin: 2em;
  padding: 1em;
  border-style: solid;
  border-width: thin;
  border-color: black;
}
```

Potem wystarczy już napisać

```
<div class="ramka">Pamiętajcie, żeby zawsze formatować tekst korzystając z CSS!</div>
```

Pamiętajcie,
żeby zawsze formatować tekst
korzystając z
CSS!

Przykład 14.2.5. Zmiana sposobu wyświetlania linków

```
a {
  font-weight: bolder;
  text-decoration: none;
  color: blue;
  background-color: white;
}

a:hover {
  background-color: yellow;
}
```

(Jest to pierwszy moment, gdy wygląd naszej strony zmienia się dynamicznie, w zależności od sytuacji. Więcej takich sztuczek w rozdziale 14.5.)

Ustaliliśmy, że linki w naszym tekście będą wyświetlać się pogrubioną, niebieską czcionką na białym tle i, że nie będą podkreślane — co poprawia moim zdaniem ogólny wygląd strony. Dodatkowo, po najechaniu kursorem myszki nad olinkowany element strony, jego tło zmieni się na żółte; gdy kursor usuniemy znad elementu, tło wróci do białego.

Przykład 14.2.6. Zmiana formatowania nagłówków

```
h1 {
  color: #000000;
  background-color: #fbeca7;
  padding-left: 2%;
  border-bottom-style: solid;
  border-bottom-width: 3px;
  border-bottom-color: #c00;
  border-top-style: solid;
  border-top-width: 1px;
  border-top-color: white;
  clear: both;
}
```

Kolory tekstu i tła — oczywiste. Dalej, robimy lekkie wcięcie tekstu tytułu (na 2% szerokości strony) w stosunku do ramki. Następnie definiujemy właściwości obramowania. Tylko na górze i na dole nagłówka ma pojawić się ciągła linia krawędzi: dolna linia grubsza i czerwona, górna cieńsza i biała. Na zakończenie podajemy parametr `clear: both`, który powoduje, że nasz nagłówek nie będzie stykał się np. z wcześniej zamieszczoną grafiką opływającą przez tekst (w rodzaju ``).

Ćwiczenie 14.2.1. Stwórz w pliku `styl.css` arkusz stylów, w którym zażadasz, by wszystkie nagłówki były wyświetlane na żółtym tle. Następnie wykorzystaj go we wszystkich swoich plikach HTML, nadając im oryginalny ale jednolity wygląd.

Następnie zmień sposób formatowania nagłówków, żądając, by dodatkowo były obwiedzione zieloną ramką, a każdy akapit miał margines na 3 em'y. Zobacz, że dokonane zmiany aplikują się teraz do wszystkich stron!

Ćwiczenie 14.2.2. Dla zabawy sprawdź, jak okropnie wyglądałyby Twoje strony, gdyby *każdy* akapit miał swoją ramkę!

□

□ **Ćwiczenie 14.2.3.** Dla zabawy sprawdź, jak wyglądałyby Twoje strony, gdybyś zastosował na swojej stronie arkusz CSS z zewnątrz, np.

```
<head>
....
<link href="http://www.mimuw.edu.pl/css/fresh.css" rel="stylesheet" >
....
</head>
```

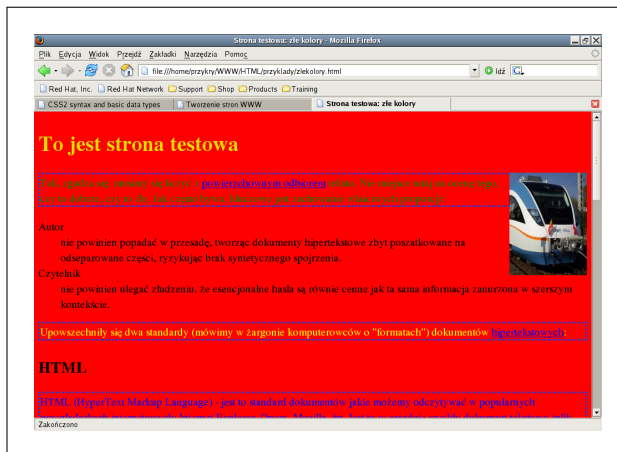
□

14.2.7 O potrzebie umiaru

Generalna zasada jest następująca: im mniej skomplikowanego formatowania, tym lepiej. Należy zrezygnować z podawania rozmiarów w absolutnych jednostkach długości, na rzecz jednostek względnych (procentów, em-ów, itp.). Nie należy przesadzać z kolorowaniem wszystkiego co możliwe i generalnie należy stosować wszystkie standardowe zasady składu tekstu.

Strona WWW to nie choinka!

Spójrz na poniższą stronę:



Strona, na której źle skorzystano z możliwości samodzielnego formatowania

Została poprawnie zapisana przy użyciu znaczników HTML, wykorzystuje składniowo poprawny arkusz CSS, ale... jest oczywiste, że autor przesadził! Kolorował wszystko, co się dało i jak się dało, w dodatku stosując agresywne, drażniące kolory, zestawiając ze sobą mało kontrastujące ze barwy dla tekstu (czarny lub niebieski) i tła (czerwony). W efekcie, strona krzykliwie rzuca się w oczy, ale... jest bardzo mało czytelna!

Na domiar złego, autor zmienił znaczenie powszechnie używanych wizualnych wyróżników: na przykład, użył niebieskiej czcionki z podkreśleniem do wyróżnienia fragmentów zdań, tymczasem jest powszechnym zwyczajem w taki właśnie sposób zaznaczać linki! A więc dodatkowo takie formatowanie prowadzi do nieporozumień.

Stosując CSS i jak najmniej uciążliwego formatowania pieczemy dwie pieczenie na jednym ogniu:

1. Oszczędzamy czas, bo formatowanie "robi się samo"!
2. Nasze strony wyglądają atrakcyjnie i mają jednolity, "nasz", charakter!

Zbyt dużo ozdóbek na stronie utrudnia jej percepcję, o czym z pewnością każdy zdążył się już przekonać surfując po WWW. Pamiętajmy także o tym, że wśród naszych odbiorców mogą znajdować się osoby łączące się z Internetem np. przez powolny modem!

Bardzo dużo ciekawych artykułów na temat formatowania z użyciem CSS znajdziemy w portalu A list apart (zob. <http://www.alistapart.com/topics/code/css/>) (polecam na początek starsze artykuły).

14.2.8 Media

Jedną z ciekawszych możliwości arkuszy CSS jest możliwość uzależnienia sposobu formatowania od rodzaju urządzenia, używanego do prezentacji dokumentu po stronie czytelnika. Mamy do wyboru m.in. następujące media:

- **aural** — synteza mowy (nie zajmujemy się tym, choć formatowanie dokumentu dla syntezy mowy to sztuka w sobie)
- **handheld** — podręczny wyświetlacz (np. telefon komórkowy)
- **screen** — ekran monitora komputerowego
- **print** — wydruk z drukarki komputerowej

Dotychczas nie podawaliśmy, jakiego rodzaju medium dotyczyły nasze reguły formatowania zapisane w CSS. Oznaczało to po prostu, że — w miarę fizycznych możliwości — do każdego. Możemy jednak dla konkretnego medium nieco zmienić przyjęte reguły formatowania, np.

```
body {
  font-size: 12pt;
  font-family: sans-serif;
}

/* jak drukować */
@media print {
  body {
    font-size: 10pt;
    font-family: serif;
  }
  A {
    font-weight: bold;
  }
}
```

W powyższym przykładzie, *na wydruku* żądamy złożenia wszystkiego (bo dotyczy to znacznika `<body>`!) bazową czcionką szeryfową, wielkości 10 punktów i wytłuszczenia linków na stronie. Dla wszystkich innych mediów, tam gdzie to możliwe, żądamy ustawienia bazowej wielkości czcionki na 12pt i użycia czcionki bezszeryfowej, dobrze wyglądającej na elektronicznych wyświetlaczach.

14.2.9 Gotowe arkusze CSS

Jak już wspomniano, arkusz CSS używany do prezentacji treści dokumentu HTML może pochodzić z zewnątrz — nawet z innego serwera. Aby umożliwić osobom początkującym korzystanie z fachowo zaprojektowanych⁸ arkuszy CSS, utworzono zestaw kilku „gotowców”, o nazwie W3C Core Styles (zob. <http://www.w3.org/StyleSheets/Core/>).

Aby skorzystać z któregoś z nich w naszym dokumencie HTML, musimy wpisać w nagłówku (tzn. pomiędzy znacznikami `<head>` i `</head>`)

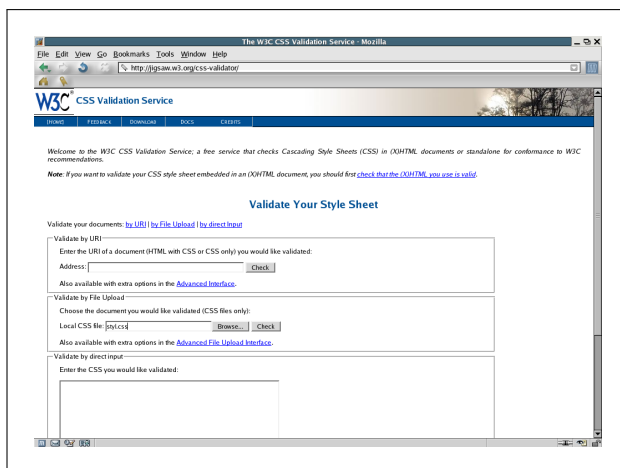
```
<link rel="stylesheet"
      href="http://www.w3.org/StyleSheets/Core/ULtramarine"
      type="text/css" >
```

gdzie Ultramarine jest nazwą konkretnego stylu. Inne dostępne style to Chocolate, Midnight, Modernist, Oldstyle, Steely, Swiss, Traditional oraz Ultramarine. Efekt ich zastosowania można zobaczyć na stronie <http://www.w3.org/StyleSheets/Core/preview> (zob. <http://www.w3.org/StyleSheets/Core/preview>).

14.2.10 Poprawność arkusza CSS

Równie łatwo jak w HTML, możemy zrobić literówkę w arkuszu CSS — i z podobnymi skutkami. Na szczęście i w tym przypadku możemy automatycznie sprawdzić poprawność składniową arkusza CSS, którego używa nasza strona WWW, korzystając z usługi CSS Validation Service (zob. <http://jigsaw.w3.org/css-validator/validator-uri.html>), zwanej popularnie Validatorem CSS.

⁸Jednak w chwili obecnej (kwiecień 2006), nie rozróżniają one mediów (ekranu, drukarki, itp), przez co np. w druku możemy dostać dokument o ciemnym tle. Ale mimo to, jest to wciąż dobry punkt wyjścia do własnych modyfikacji.



Validator CSS w akcji

- Ćwiczenie 14.2.4.** To nie jest ćwiczenie na jeden raz: od tej pory, po każdej istotnej modyfikacji *zawsze* sprawdzaj swój arkusz stylów w Validatorze CSS. Sprawdź go teraz i popraw ewentualne błędy. Powiedzmy to wyraźnie: odkąd wiesz, że istnieje Validator CSS, nie ma żadnego usprawiedliwienia dla ewentualnych błędów składniowych w Twoim arkuszu CSS!
-

14.3 Projektowanie stron WWW

Zalety wykorzystania WWW dla publikacji dokumentów:

- Dostępność i łatwość prezentacji informacji
- Aktualność dokumentów
- Łatwość rozpowszechniania

Wady tego rozwiązania

- Dokumenty nie zawsze czytelne/dostępne dla odbiorcy
- Informacja nietrwała i nieweryfikowana
- Kłopotliwa(?) publikacja

Mnóstwo konkretnych informacji o tym, jak projektować *naprawdę dobre* strony WWW możesz uzyskać ze znakomitego, okraszonego mnóstwem przykładów, podręcznika internetowego Web Style Guide (zob. <http://www.webstyleguide.com>). Gorąco polecam!

Przejrzyjmy kolejne etapy tworzenia dokumentu przeznaczonego do publikacji na stronach WWW:

1. Projekt
 - (a) Cel, odbiorca, treść strony WWW
 - (b) Forma
2. Realizacja
 - (a) Treść otoczona znacznikami HTML
 - (b) Nadanie treści jednolitej, eleganckiej formy za pomocą CSS
 - (c) Sprawdzenie poprawności HTML i CSS w Validatorach
3. Publikacja na serwerze WWW
4. Sprawdzenie w (obu) Validatorach (i ewentualnie w kilku wiodących przeglądarkach)

To jest banał, ale w przypadku dokumentów internetowych także musimy w jakiś sposób ustalić sobie, kto jest ich głównym adresatem. Piszemy nie dla siebie, ale dla czytelnika! Dlatego na przykład, zanim zamieścimy swój klip wideo na stronie WWW, musimy zadać sobie pytanie: czy ma dostatecznie nowoczesny komputer by poradził sobie z odtwarzaniem plików multimedialnych?

14.3.1 Specyfika stron domowych

Moja strona domowa świadczy o mnie! To współczesna odmiana wizytówki, połączonej z CV, albumem zdjęciowym i innymi specjałami — w zależności od stopnia ekshibicjonizmu autora.

Adresatem takiej strony jest więc teoretycznie każdy, ale w praktyce będziemy mieli zapewne na myśli: znajomych ze studiów i nie tylko, potencjalnych pracodawców, rodzinę, itp.

Ponieważ głównym celem strony domowej jest prezentacja autora, powinna być schludna, efektowna (ale bez zbędnej przesady), a przede wszystkim: *treściwa*.

Na dobrej stronie domowej powinny znaleźć się następujące elementy:

- Imię i nazwisko
- Kilka słów o sobie
- Kontakt, w tym adres e-mail i (opcjonalnie) publiczny klucz PGP
- Zdjęcie (opcjonalnie)
- Lista zainteresowań, z linkami do adekwatnych stron
- Lista publikacji
- Linki do innych istotnych stron/dokumentów autora (np. do życiorysu w formacie PDF)

- Data aktualizacji strony

Oto przykładowy kod HTML takiej strony domowej.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
      charset=ISO-8859-2">
    <title>Strona internetowa Jana Kowalskiego</title>
    <meta name="Author" content="Jan_Kowalski">
    <link href="http://students.uw.edu.pl/~jan.kowalski/styl.css"
      rel="stylesheet">

  </head>

  <body>
    <h1>Jan Kowalski</h1>

    <div class="prawaramka">
    <br>
    Moje zdjęcie z wakacji 2004.
    </div>

    <p>Jestem studentem III roku <a href="">Wydziału Geografii</a>
    <a href="http://www.uw.edu.pl">Uniwersytetu Warszawskiego</a>.
    Tematem mojej pracy licencjackiej jest <cite>Opis zasobów
    wodnych Sahary</cite>. Gdy tylko ją ukończę, zamieszczę na tej
    stronie link do pełnego tekstu!

    <p>Zainteresowani mogą zapoznać się z moim <a
    href="cv.pdf">życiorysem naukowym</a> w formacie PDF.

    <h2>Zainteresowania</h2>

    <ul>

    <li><strong>Muzyka poważna</strong>

    <li><a href="szachy.html">Szachy</a> (prowadzę specjalną stronę
    internetową <a href="szachy.html">klubu szachowego</a>,
    zapraszam!)

    <li>Górskie wycieczki (mam uprawnienia przewodnickie i jestem
    członkiem
    <a href="http://www.republika.pl/skgw">SKGW</a>)

    </ul>

    <h2>Kontakt</h2>

    <dl>
      <dt>E-mail</dt>

      <dd>
```

```

        <a href="mailto:jan.kowalski@students.uw.edu.pl">
            jan.kowalski@students.uw.edu.pl
        </a>
    </dd>

    <dt>Gadu–Gadu</dt>

    <dd>janek2005</dd>

    <dt>Telefon</dt>

    <dd>88 998 7777</dd>

</dl>

<p>Jeśli chcesz szyfrować pocztę do mnie, użyj poniższego klucza
PGP: <kbd>1024D/28C9E826</kbd>.</p>

<h2>Publikacje</h2>

<p>Pisuję trochę do niszowych fan–zinów internetowych, głównie o
muzyce. Jeśli chcesz, możesz przeczytać moje artykuły, nie tylko
on–line:</p>

<ol>
    <li><a
        href="http://www.fanzin.pl/908070"><cite>Inspiracje
        speed–metalowe</cite></a> – rzecz o związkach muzyki
        metalowej i organowej

        <li><cite>Śpiewnik beskidzki na bieszczadzkiej trasie</cite>
            –
            wspomnienia zamieszczone w <cite>Kwartalniku
            Piechura</cite>, 3/2004

</ol>

<hr>

<script type="text/javascript"
    src="http://students.uw.edu.pl/~jan.kowalski/lastmod.js">
</script>

</body>
</html>

```

Powyższa strona korzysta z dwóch zewnętrznych plików: `lastmod.js` i `styl.css`. Skrypt JavaScriptu `lastmod.js` jest opisany w rozdziale 14.6.1. Natomiast plik `styl.css` mógłby wyglądać następująco (właściwie jedyną ważną pozycją w nim jest sposób formatowania znaczników klasy `prawaramka`):

```

body {
    background–color: #ffffef;
    color: black;
    font–family: sans–serif;
    margin–left: 1em;

```

```

padding-left: 0.5em;
border-left-color: orange;
border-left-width: thick;
border-left-style: solid;
}

h1 {
background-color: navy;
color: white;
padding-left: 0.5em;
border-bottom-style: solid;
border-bottom-width: 3px;
border-bottom-color: orange;
clear: both;
}

h2 {
color: navy;
}

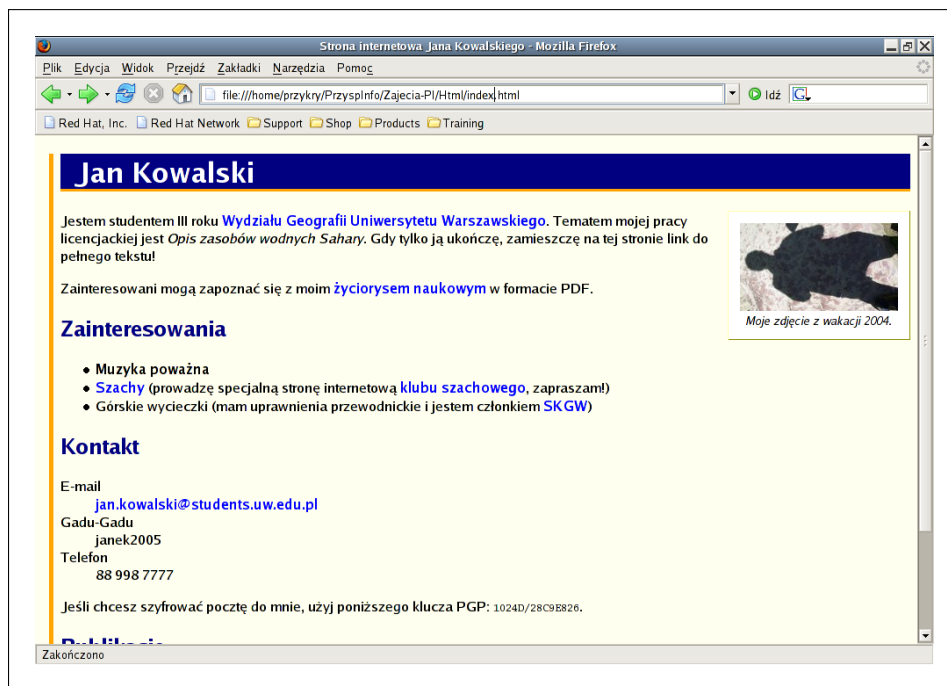
a {
font-weight: bolder;
text-decoration: none;
color: blue;
}

a:hover {
background-color: yellow;
}

.prawaramka {
float: right;
background-color: white;
font-size: smaller;
font-style: italic;
text-align: center;
margin-left: 1em;
padding: 1em;
border-style: outset;
border-width: thin;
border-color: yellow;
}

div.aktualizacja {
font-size: xx-small;
font-style: italic;
}

```



Wygląd zaprojektowanej powyżej strony domowej Jana Kowalskiego.

Strona domowa nie powinna być przesadnie długa. Jeśli np. autor ma na koncie setkę publikacji, powinien dla nich utworzyć osobną stronę, a na stronie domowej zamieścić jedynie link.

Ćwiczenie 14.3.1. Zaprojektuj swoją stronę domową zgodnie z podanymi regułami. Używaj znaczników wyłącznie adekwatnych do treści i sensu. Unikaj bezpośredniego formatowania wizualnego. Zastosuj ascetyczny, lecz wysmakowany arkusz CSS.

14.4 Czego *nie należy* publikować na WWW?

Nie chcemy poruszać tu kwestii ewentualnej *niestosowności treści* stron WWW, choć jest to — oczywiście! — temat bardzo ważny: w niniejszym rozdziale zajmiemy się sprawą bardziej techniczną, a mianowicie *formatem* publikowanych dokumentów. Już wiemy, że zalecany formatem publikacji dokumentów na WWW jest HTML. W uzasadnionych przypadkach, gdy dokument ma bardzo skomplikowaną formę, którą chcemy zachować, używamy do publikacji formatu PDF. Warto więc teraz zadać sobie pytanie: czy są formaty dokumentów, których *nie powinno się* publikować na WWW?

Okazuje się, że tak rzeczywiście jest. Niepisana umowa użytkowników sieci WWW głosi, że

Nie należy publikować na WWW dokumentów, do wyświetlenia których wymagane jest komercyjne lub trudno osiągalne oprogramowanie. Niemile widziane są także dokumenty w zamknię-

tych (niejawnych) formatach, takich jak np. format dokumentów edytora Word.

A więc **nie publikujemy w sieci:**

- tekstów w Wordzie,
- arkuszy kalkulacyjnych w Excelu,
- prezentacji w PowerPointcie,
- dokumentów w formatach Microsoft Office lub OpenOffice,

ani żadnych innych formatów, do poprawnego obejrzenia których wymagana jest tylko jedna, albo komercyjna, albo o specyficznych wymaganiach sprzętowych lub systemowych, przeglądarka. Zamieszczanie tego rodzaju dokumentów od razu wskazuje na nieprofesjonalizm autora strony i jest poniekąd oznaką lekceważenia czytelnika. Gdy widzę taką stronę, to jakbym słyszał autora:

Nie umiesz odczytać mojej recenzji w najnowszej wersji Worda?
To sobie ją kup, bo ja, jak widzisz, mam ją!

albo:

No, przecież nie będę tracić czasu na konwersję tabeli z Excela!
Kto zechce, na pewno sobie jakoś poradzi!

W sytuacjach wyjątkowych, gdy swoje strony adresujemy do bardzo wąskiej grupy odbiorców, możemy złamać powyższe reguły: na przykład, prowadzimy kurs obsługi Excela i chcemy zamieścić na swojej stronie przykładowe pliki, albo nauczyciel umawia się z grupą studentów, że będzie im przedstawiał testy do wypełnienia w domu w postaci plików `.odt`⁹.

Jeśli chcemy udostępnić do przeglądania dokumenty, których oryginały są zapisane w formacie nie nadającym się do publikacji na WWW, powinniśmy je wyeksportować do jednego z dopuszczalnych: HTML lub PDF. Większość współczesnych aplikacji (w tym wszystkie wymienione powyżej) ma opcję eksportu dokumentu do formatu HTML, lecz nie należy z niej korzystać, poza wyjątkowymi sytuacjami. Zazwyczaj znacznie lepszym rozwiązaniem bywa eksport do formatu PDF i zamieszczenie pliku PDF na stronie WWW. O konwersji dokumentów z popularnych formatów do HTML i PDF piszemy w rozdziałach [14.6.5](#) i [14.6.6](#)

14.5 Dynamiczne strony WWW

Ten rozdział nie jest obowiązkowy.

⁹Jest to format dokumentów tekstowych OpenOffice.

Dokumenty HTML udostępniane na naszych stronach nie muszą wcale mieć całkowicie zafiksowanej treści. Przykładowo, byłoby miło, gdyby

- Nasza strona „*sama*” rozpoznawała datę swojej aktualizacji i wyświetlała ją w stosownym miejscu (np. w stopce)
- Albo, by dowolny link w tekście został podświetlony po umieszczeniu nad nim kursora myszki
- Albo, żeby można było niezauważalnie do naszej strony *wklejać bieżącą* zawartość jakiegoś pliku (np. swoje dane adresowe, albo krótką notatkę tekstową).
- Albo, żeby można było uzależnić wyświetlaną treść od tego, kto, z jakiego adresu internetowego (np. jeśli nie z Polski, to wyświetlić wersję po angielsku) i jakiej przeglądarki używa do oglądania naszej strony!

To wszystko jest możliwe, a nawet proste, ale wymaga zazwyczaj bardziej skomplikowanej technologii niż tylko HTML.

Możemy także realizować niezbyt mądre pomysły, np. wyskakujące dodatkowe okienka, albo wyświetlanie dzisiejszej daty i imienin (mam tylko nieśmiałe pytanie: po co?!).

14.5.1 CSS

Tak, już CSS daje pewne możliwości dynamicznej zmiany wyglądu strony, m.in. dzięki pseudoklasie `:hover`, zob. przykład 14.2.6.

14.5.2 JavaScript

Wszystkie wiodące przeglądarki mają wbudowaną obsługę JavaScriptu: prostego języka służącego zmianie zawartości strony w zależności od m.in. działań użytkownika (na przykład umieszczenie kursora nad obiektem lub używanie konkretnej przeglądarki), lub informacji dostarczanych przez serwer WWW wraz ze stroną (takich jak np. data modyfikacji strony).

HTML daje mocną integrację z JavaScriptem (zob. <http://www.w3.org/TR/html4/interact/scripts.html>) (znacznik `<script>` oraz elementy znaczników dotyczące zdarzeń takich jak `onmouseover`, tzn. najechanie myszką na oznakowany element) i umożliwia sztuczki takie jak dynamiczna podmiana obrazka, gdy najdziemy nań myszką.

JavaScript przydaje się bardzo np. do weryfikacji danych wprowadzanych do formularzy HTML. W naszym kursie nie omawiamy formularzy HTML, gdyż wymagają one dodatkowego oprogramowania po stronie serwera.

Pamiętajmy, że najczęściej zwykły twórca stron HTML (taki jak np. typowy uczestnik niniejszego kursu) nie będzie miał większej potrzeby korzystania z JavaScriptu.

14.5.3 PHP

PHP, czyli PHP Hypertext Processor (zob. <http://www.php.net>), jest znakomitym narzędziem do tworzenia stron WWW "w locie", po stronie serwera. Strona stworzona z wykorzystaniem PHP to zwykły dokument HTML uzupełniony o niezwykle znaczniki, w których zamieszczamy kod PHP. Najczęściej pliki zawierające kod HTML uzupełniony o instrukcje skryptowe PHP noszą rozszerzenie `.php`.

Serwer — pod warunkiem, że obsługuje¹⁰ PHP — zanim udostępni czytelnikowi taką stronę, wykona instrukcje PHP i dopiero na tej podstawie utworzy stronę HTML, którą wyśle w końcu do przeglądarki czytelnika (dlatego czytelnik nie widzi w źródle żadnego kodu PHP!)

To, czy będziemy mogli korzystać z potęgi PHP, zależy od możliwości serwera, na którym znajduje się nasza strona. Przykładowo, studencki serwer (zob. <http://students.mimuw.edu.pl>) Wydziału MIM UW bez trudu obsłuży wszystkie instrukcje PHP zawarte na zamieszczonych tam stronach, wszelako pod jednym warunkiem, że strony wykorzystujące PHP będą miały rozszerzenie `.php` — np. `index.php`.

Wspaniale jest wzbogacić strony WWW korzystając z możliwości oferowanych przez PHP (na przykład cały serwis internetowy Wydziału Matematyki (zob. <http://www.mimuw.edu.pl>) jest zbudowany w oparciu o mechanizmy PHP), ale musimy pamiętać, że korzystanie z PHP oznacza zajęcie się — całkiem na poważnie — programowaniem w tym języku skryptowym, a więc dużą inwestycję własnego czasu.

14.6 Kilka tricków przydatnych w życiu

Ten rozdział nie jest obowiązkowy.

14.6.1 Data aktualizacji strony

Ponieważ strony WWW można nieustannie modyfikować, przydałby się prosty sposób automatycznego wstawiania do dokumentu daty jego ostatejnej aktualizacji (zwykle umieszcza się ją w stopce strony). Można to zrobić na wiele sposobów, najprostszy jest przy użyciu JavaScriptu:

```
Aktualizacja: <script
type="text/javascript">
  document.write(
  document.lastModified)
</script>
```

Aktualizacja: 28 listopada
2006

¹⁰Część, zwłaszcza darmowych, serwerów WWW blokuje tę usługę, niestety, rezerwując ją dla kont płatnych.

Odrobinę bardziej skomplikowany kod da nam większą odporność na ewentualne późniejsze modyfikacje:

```
<script type="text/javascript"
src="lastmod.js" > </script>
```

*Aktualizacja: 28 listopada
2006*

W pliku `lastmod.js` osobno zamieszczamy krótki kod w JavaScriptcie:

```
document.write('<div class="aktualizacja"> Aktualizacja: '
+ document.lastModified + '</div>');
```

Łatwo domyślić się znaczenia powyższych instrukcji JavaScriptu: funkcja `document.write(...)` ma coś wpisać do kodu źródłowego naszej strony HTML. Tym czymś są trzy napisy: kilka znaczników HTML `<div class="aktualizacja"> Aktualizacja:`, następnie wartość parametru `document.lastModified`, czyli data ostatniej aktualizacji dokumentu, którą podaje serwer przy okazji wysyłania strony do Czytelnika. Na zakończenie wypisywany jest znacznik zamykający: `</div>`.

14.6.2 Niezauważalne wklejenie zawartości zewnętrznego pliku

Najprościej zrobić to, korzystając ze skryptu PHP i wpisując w odpowiednim miejscu dokumentu HTML:

```
<body>
....

  <div>
    <?php
      include("przyklady/notka.txt");
    ?>
  </div>

....
</body>
```

Aby to zadziało, Twój serwer musi obsługiwać PHP! Dowiedz się u lokalnego administratora systemu, jak mają się sprawy i np. czy pliki HTML wykorzystujące w swej treści PHP nie muszą mieć rozszerzenia `.php`.

14.6.3 Formatowanie dokumentu do wydruku

Zapewne nie raz zdenerwowało Cię, że pięknie wyglądające strony WWW zazwyczaj kiepsko się drukują: są albo zbyt rozwlekłe, albo zbyt kolorowe, albo po prostu fatalnie rozplanowane. No i te denerwujące dodatki nawigacyjne! (chciałoby się wydrukować treść główną strony, bez pomocniczych menu i innych ozdóbek).

Okazuje się, że możesz łatwo sprawić, by przynajmniej Twoje strony drukowały się jak należy! Znow wszystko daje się załatwić przy pomocy CSS, gdyż można podać specyficzne reguły formatowania **w zależności od użytego**

medium, to znaczy: inne dla przeglądarki graficznej, inne dla dźwiękowej, i wreszcie: inne dla *wydruku* (zob. rozdział 14.2.8).

Zobaczmy, co można zrobić, żeby Twoje strony WWW lepiej¹¹ się drukowały. Oto stosowny fragment pliku CSS:

```
@media print {
  A:after {
    content: " [" attr(href) "]" ; font-style: italic;
  }
  body {
    border-style: none;
    font-family: Times, serif;
    font-size: 12pt;
  }
  div.menu {
    visibility: collapse;
  }
}
```

Powyższy fragment najlepiej umieścić na końcu pliku CSS.

1. W pierwszej linii wskazałem, że zawarte instrukcje formatowania będą odnosić się wyłącznie do wydruku.
2. Następnie zażądałem, by — na wydruku — po każdym linku był dodawany (kursywą i w nawiasach kwadratowych) jego adres internetowy. Może się przydać, gdy jedynie *czytamy z kartki* taki dokument, prawda? Niestety, nie wszystkie przeglądarki (w tym mój Internet Explorer) potrafią dokonać takiego cudu! Na szczęście, nie ma powodu do obaw: jeśli przeglądarka czegoś nie potrafi zrobić, to po prostu tego nie robi i idzie dalej!
3. Potem usunąłem z wydruku niepotrzebny ozdobnik w postaci ciągłej kreski po lewej stronie dokumentu. Rzeczywiście, po co nam coś takiego na wydruku?
4. Zażądałem też, by dokument był drukowany (dość dużą) czcionką wielkości 12 punktów, w dodatku szeryfową (ładniejszą na papierze), najlepiej Times'em.
5. Na zakończenie zażądałem, by zawartość wszystkich znaczników `<div class="menu">...</div>` zniknęła z wydruku. Na moich stronach, w takie znaczniki ujęte są wszystkie elementy menu nawigacyjnego. Tym samym, menu nawigacyjne nie pojawi się na wydruku, ku radości (i oszczędności papieru i tuszu) czytelnika!

Musimy jednak pamiętać, że im bardziej skomplikowana strona, tym trudniej będzie nam ją pięknie wydrukować (wystarczy, że pomieszymy w niefortunny sposób piksele z punktami podczas oznaczania rozmiarów bloków:

¹¹Przynajmniej, w przeglądarkach dobrze implementujących CSS 2.0, np. Firefoxie.

na ekranie będzie dość ładnie (chyba, że drastycznie zmienimy rozdzielczość), ale na papierze... zwykle będzie nieciekawie). Jeszcze jeden powód, by samoograniczyć się do form być może niezbyt ambitnych, ale za to klarownych. Dokumenty o wielkim stopniu złożoności lepiej od razu tworzyć w formacie PDF, mając w ten sposób gwarancję (niemal) pełnej powtarzalności wyglądu dokumentu niezależnie od środowiska.

Zwłaszcza strony z ramkami (<frameset>) mogą przysporzyć wielu czytelnikom trudności, także podczas wydruku; jest to argument za tym, by ograniczyć stosowanie ramek na rzecz dynamicznie tworzonych dokumentów, np. przy użyciu PHP.

Pamiętajmy także, że różne przeglądarki (bardzo) różnie sobie radzą z obsługą CSS i w konsekwencji także z drukowaniem stron HTML.

14.6.4 Szablony stron

Wyprodukowawszy już kilka stron WWW zauważamy, że wszystkie mają wiele stałych elementów. Byłoby wielką oszczędnością czasu, gdyby można było oszczędzić sobie wpisywania ich za każdym razem.

Jest kilka możliwych rozwiązań tego problemu:

- Zrobić sobie plik `szablon.html` zawierający „pustą” stronę, a potem tylko wypełniać go treścią, zapisując oczywiście za każdym razem w pliku o innej nazwie!
- Zrobić program, który automatycznie — w trakcie wyświetlania! — uzupełnia stronę o stałe elementy; najczęściej taki program będzie skryptem w języku PHP.

Każde z tych rozwiązań ma wady i zalety.

Plik `szablon.html`

Jest to rozwiązanie jakie stosuję na co dzień. Na przykład, strony internetowe wspierające niniejszy kurs przygotowałem, używając następującego szablonu:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD_HTML_4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd" >
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html;_
    charset=iso-8859-2" >
  <title> </title>
  <meta name="Author" content="Piotr_Krzyzanowski" >
  <link href="styl.css" rel="stylesheet" >
</head>
<body>
```

```

<div class=" menu" >
<a href="" class=" left">Poprzednia lekcja</a>
<a href="" class=" up">Spis treści</a>
<a href="" class=" right">Następna lekcja</a>
</div>

<h1>
Szablon: uzupełnij tytuł strony!
</h1>

<p>

<div class=" pamietaj" >
<h2> Warto zapamiętać</h2>
<p>

</div>

<div class=" menu" >
<a href="" class=" left">Poprzednia lekcja</a>
<a href="" class=" up">Spis treści</a>
<a href="" class=" right">Następna lekcja</a>
</div>

<script type=" text/javascript" src=" lastmod.js" >
</script>

</body>
</html>

```

Zaletą takiego podejścia jest to, że — gdy już opracujemy elegancki szablon — będzie nam bardzo łatwo „produkować” kolejne strony.

Jego zasadniczą wadą jest to, że *jakakolwiek* zmiana szablonu będzie od nas wymagała żmudnej, ręcznej edycji **wszystkich** (sic!) dotychczas utworzonych stron — w przypadku, gdy zechcemy zachować ich jednolity wygląd.

Szablon oparty na ramkach

Częściowym (i najczęściej zupełnie zadowalającym) rozwiązaniem powyższego problemu jest stworzenie doprawdy minimalistycznego szablonu i zawarcie stałych elementów strony w ramach. Wadą jest to, że strony z ramkami są w pewnym sensie zbyt złożone, co zwykle powoduje np. pewne niedogodności podczas drukowania. Dlatego dotychczas w ogóle nie omawialiśmy znaczników (<frameset>) pozwalających definiować strony złożone z ramek — osobnych plików HTML zamieszczanych w wyznaczonych miejscach na specjalnie spreparowanej stronie HTML.

Nie będziemy tutaj omawiać dłużej tego rozwiązania, ponieważ — chociaż wygodne dla autora stron WWW — wiąże się z kilkoma niedogodnościami dla czytelnika (m.in. drukowanie takich stron i wyszukiwanie w nich może sprawić kłopoty czytelnikom nieobeznanym z tą technologią).

Szablon w PHP

Wykorzystanie języka skryptów PHP pozwala nam tworzyć w pełni dynamicznie konstruowane dokumenty HTML. Skrypty PHP są przetwarzane po stronie serwera (czyli tam, gdzie została zamieszczona strona WWW), a czytelnik dostaje gotową, „obrobioną” stronę. Na przykład, bez trudu spowodujemy, żeby, przed wyświetleniem dokumentu u czytelnika, nasz tekst HTML został uzupełniony o fragmenty innych plików HTML - dzięki czemu zamiast używać ramek, nasz serwer wysłałby gotową złożoną (jedną!) stronę HTML, na której z oddzielnego pliku zostałby wcześniej wklejony tekst HTML zawierający menu.

Nauka języka PHP wykracza poza zakres niniejszego kursu. Z drugiej strony, osoby pragnące tworzyć, a następnie utrzymywać w dobrej kondycji dużo stron WWW powinny rozważyć opcję wykorzystania przynajmniej części jego możliwości. PHP ma tę zaletę, że na początku można zacząć od drobnych wstawek kodu PHP do treści „zwykłej” strony HTML i w miarę rozwoju własnych umiejętności potem go komplikować.

Prowadzenie ankiet on-line, wyświetlanie statystyk odwiedzin, itp. jest proste właśnie przy użyciu skryptów PHP, jednakże sam język wymaga opanowania, a więc — dodatkowego nakładu pracy, co nie zawsze jest najbardziej ekonomicznym rozwiązaniem.

Ćwiczenie 14.6.1. Przygotuj własny szablon stron WWW. Pamiętaj, im więcej parametrów szablonu będzie zależało od zewnętrznych plików, tym łatwiej będzie Ci później dokonywać konsystentnych zmian w wyglądzie swoich stron opartych na tym szablonie.

Ćwiczenie 14.6.2. Uzupełnij arkusz CSS z którego korzystają Twoje strony o sformatowanie wydruku. Pamiętaj, że nie wszystkie przeglądarki równie dobrze obsługują CSS — sprawdź więc efekt w kilku wiodących.

Ćwiczenie 14.6.3. Wstaw, korzystając z JavaScriptu, daty aktualizacji na swoich stronach WWW. Użyj ładnego formatowania dla `<div class="aktualizacja">`.

14.6.5 Konwertery HTML

Nie wszyscy umieją sami tworzyć pliki w formacie HTML, ale niewątpliwie wszyscy chcieliby mieć wygodę zamieszczania dowolnych swoich dokumentów na stronach WWW. Dla nich większość aplikacji biurowych ma możliwość eksportu dokumentów do formatu HTML. Niestety — w większości wypadków — choć efekt wizualny może być pozornie pozytywny, w rzeczywistości rzadko kiedy taka konwersja jest naprawdę udana. Dlatego, gdyby ktoś miał rzeczywiście taką *konieczność*, raczej polecałbym w takich wypadkach eksport do formatu PDF, o czym poniżej.

Dla porządku jednak zobaczymy, jakie są możliwości eksportu do HTML:

Word (pliki typu `.doc`, `.rtf`) Poczynając od Word 2000, mamy możliwość

bezpośredniego zapisu do formatu HTML, ale jakość tej konwersji jest najczęściej **fatalna!** Po intensywnym ręcznym retuszu uzyskanego kodu, ostateczny efekt może być jednak zadowalający. Natomiast najnowsza wersja Microsoft Word 2003 ma opcję zapisu do formatu HTML w wersji nieco oczyszczonej: Plik → Zapisz jako → Strona sieci Web, przefiltrowana. Faktycznie, plik HTML ma mniej znaczników, ale wciąż jego jakość pozostawia wiele do życzenia. Strony HTML generowane przy użyciu innej aplikacji wchodzącej w skład pakietu Microsoft Office: Publisher 2003 są, moim zdaniem, nieakceptowalne — o ile nie chcemy ograniczać grona czytelników wyłącznie do posiadaczy przeglądarki Microsoft Internet Explorer.

PowerPoint (pliki typu .ppt) Prezentacje PowerPointa nie nadają się do eksportu do HTML. Owszem PowerPoint daje możliwość zapisu „do HTML”, ale jako zestaw stron złożonych z samych obrazków! To rozwiązanie ma, moim zdaniem, same wady. Znacznie lepiej jest eksportować prezentację PowerPointa do pliku PDF.

Excel (pliki typu .xls) Efekty eksportu nie są najlepsze, zwłaszcza jeśli arkusz zawierał dużo formatowania (czcionki większe i mniejsze, itp). Odradzam konwersję dużych tabel poprzez eksport do HTML, gdyż zazwyczaj prowadzą do bardzo ciężkich stron¹² WWW. Duże tabele najlepiej eksportować albo do pliku PDF, albo (tracąc formatowanie) do pliku tekstowego, a następnie zamieścić go w pliku HTML otoczywszy znacznikami `<pre>` `</pre>`. Nie będą piękne, ale będą dość czytelne i — co najważniejsze! — nie będą się ładować do przeglądarki godzinami...

OpenOffice (pliki .odt, .ods, .odp, .sxw, .sxc, .sxi, .doc, .xls, .ppt) W zasadzie zawsze konwersja do HTML jest trochę lepsza niż analogicznych aplikacji Microsoft Office, choć jednocześnie należy dodać, że również (w zasadzie) zawsze ręczny (lub zautomatyzowany) retusz (zmiana, lub wręcz usunięcie stylu CSS z nagłówka, a także likwidacja mnóstwa zbędnych znaczników) bardzo pomaga. Wady są podobne jak w analogicznych aplikacjach Microsoft Office.

LaTeX Nie ma wystarczająco dobrych narzędzi (co zadziwiające, bo podejście do struktury dokumentu jest bardzo podobne). Najlepszy obecnie jest tex4ht (zob. <http://www.ctan.org/tex-archive/help/Catalogue/entries/tex4ht.html>).

14.6.6 Tworzenie dokumentów PDF

Jedyny sensowny sposób to zapis do formatu PDF z innych programów.

OpenOffice Eksportuje wprost do PDF. Daje możliwość m.in. doboru stopnia kompresji grafik włączanych do dokumentu.

¹²To znaczy plików o dużym rozmiarze.

Word i inne aplikacje Windows Microsoft Office nawet w wersji 2003 nie daje bezpośredniej możliwości zapisu w formacie PDF. Na szczęście, wystarczy zainstalować „drukarkę” w Windows, która drukuje do pliku w formacie PDF. Większość tego typu narzędzi (np. PDFCreator (zob. <http://sourceforge.net/projects/pdfcreator/>)) korzysta z Ghostscriptu dla wykonania zadania.

LaTeX Doskonale efekty, łącznie z obsługą hiperłączy (także wewnątrz dokumentu — spisy treści, bibliografia, przypisy), daje pdfTeX (zob. <http://www.tug.org/applications/pdftex/>), który generuje pliki PDF bezpośrednio z kodu źródłowego L^AT_EX’a.

Dodajmy na marginesie, że LaTeX i wspomagające pakiety: Beamer, ConTeXt lub Pdfscreen, dają możliwości tworzenia znakomitych prezentacji (zob. <http://latex-beamer.sourceforge.net/beamerexample5.pdf>) od razu w PDF. Jako ciekawostkę podaję, że równie oszałamiające prezentacje można tworzyć wprost(?) w HTML, wykorzystując możliwości CSS (zob. <http://www.meyerweb.com/eric/tools/s5/>).

14.7 Przyszłość HTML

Standard dokumentów dla stron WWW nieustannie ewoluje. Nauczyliśmy się dobrze posługiwać się formatem HTML 4.01, obsługiwanym przez wszystkie współczesne przeglądarki. Następnym krokiem w ewolucji HTML jest format XHTML, bardzo podobny do swego przodka, ale o bardziej przemyślanej składni. Między innymi, znaczniki takie jak `<p>` tym razem *muszą* mieć znacznik zamykający `</p>`, a znaczniki nie posiadające znacznika zamykającego, np. ``, muszą przyjąć postać ``.

Przyszłością formatu stron WWW jest bez wątpienia format XML, o niebo bardziej elastyczny od HTML (z regułami jego interpretacji zapisywanymi w oddzielnym arkuszu XSL¹³).

Będzie on prawdopodobnie tym formatem, który umożliwi stworzenie uniwersalnego środowiska składu tekstów, pozwalającego na to, by jeden tekst bazowy został właściwie sformatowany dla dowolnego medium: wydruku na papierze, wyświetlenia na ekranie komputera w przeglądarce WWW, czy też (powiedzmy) odczytania z pomocą syntezy mowy.

Naturalnie, kluczem do sukcesu będzie środowisko tworzenia dokumentów narzucające właściwą (lecz bardziej elastyczną od HTML) strukturę logiczną dokumentom. Wydaje się, że L^AT_EX ma duże szanse rozwinąć się właśnie w tym kierunku.

□ **Ćwiczenie 14.7.1.** Jeśli dalej czujesz niedosyt — bierz się do samodzielnej pracy! □

¹³W pewnym sensie uogólnienie dualizmu HTML — CSS.

Rozdział 15

Prezentacje

15.1 Wprowadzenie

Jednym z ważnych elementów pracy biurowej jest tworzenie prezentacji. Na potrzeby prostych prezentacji wystarczy posłużyć się kreatorem, jednak w przypadkach bardziej złożonych warto poznać pełne możliwości środowiska takie jak praca w poszczególnych widokach oraz definiowanie animacji. Bardziej zaawansowani użytkownicy skorzystają z możliwości definiowania wzorca strony czy modyfikacji stylów.

Niezależnie od stopnia zaawansowania użytkownika, istotne jest stosowanie dobrych praktyk tworzenia prezentacji, a w szczególności wcześniejsze ich planowanie.

Niniejszy moduł pozwoli *aktywnie studiującej* osobie osiąść umiejętności pozwalające na:

- tworzenie prostych prezentacji za pomocą kreatora,
- pracę w różnych widokach,
- definiowanie animacji,
- edycję wzorca strony,
- modyfikację stylów.

15.1.1 Przygotowywanie prezentacji

- Najpierw planowanie
 - harmonogram
 - rysunki
- Podstawowe błędy przy przygotowywaniu slajdów
 - na slajdach nie umieszcza się notatek

– nie wolno przesadzać z wyróżnieniami

- Przynajmniej jedna próba przed właściwą prezentacją

Najpierw planowanie

Przygotowywanie prezentacji należy zacząć od planowania. Pierwszy konspekt warto sporządzić tradycyjnymi metodami — przy pomocy kartki i ołówka. Należy określić kolejność omawianych zagadnień oraz wykonać pierwsze przymiarki co do liczby i zawartości slajdów. Oczywiście w trakcie późniejszej pracy nad slajdami konspekt będzie nieustannie ewoluował, ale to nie znaczy, że można się nie przykładać.

Dobre rysunki są w stanie przekazać tysiąc słów, więc warto się zastanowić, czy i które slajdy o nie wzbogacić. Jeżeli nie dysponujemy gotowymi rysunkami i dopiero trzeba je przygotować, pierwsze projekty również wykonujemy na kartce.

Podstawowe błędy przy przygotowywaniu slajdów

Zazwyczaj slajdy zawierają przede wszystkim tekst. Nie należy ich jednak traktować jako notatek. Osoby uczestniczące w prezentacji przeczytają tekst na slajdach szybciej niż my to zrobimy na głos i przestaną się interesować tym co mówimy. Jeżeli chcemy, żeby nas słuchano, bądźmy zwięzli i stosujmy wypunktowania.

Przygotowując slajdy można rozważyć różne sposoby wyróżnienia pewnych elementów, np. nadanie im innego koloru, użycie ciekawej czcionki czy powiększenie. Do przyciągnięcia uwagi słuchaczy doskonale nadają się również animacje. Należy przy tym zachować umiar, gdyż nadużywanie wyróżnień na jednym slajdzie rozprasza.

Dobre slajdy to nie wszystko

Przed prezentacją należy przeprowadzić przynajmniej jedną próbę. Osoby, które tego nie zrobią, łatwo poznać, gdyż niektóre zagadnienia omawiają po kilka razy.

15.1.2 OpenOffice Impress

OO Impress to narzędzie do przygotowywania pokazów slajdów, które łączy cechy:

- edytora tekstu (tekst, listy wyliczeniowe i wypunktowania, tabele),
- programu do rysowania (grafika wektorowa, obiekty 3D),
- arkusza kalkulacyjnego (arkusze kalkulacyjne, wykresy),

- edytora formuł matematycznych oraz
- programu do tworzenia prostych animacji (przejścia między slajdami, animacje elementów na slajdach).

Impress jest ściśle zintegrowany z pozostałymi składnikami pakietu *OpenOffice*. Tekst na slajdach umieszcza się na przykład w polach tekstowych, które można wypełniać tak samo jak dokumenty w programie *Write*, włączając w to:

- wprowadzanie tekstu,
- formatowanie tekstu,
- tworzenie tabel,
- używanie stylów, oraz
- automatyczną kontrolę pisowni.

Pozostałe składniki pakietu *OpenOffice*, z których funkcji pośrednio korzysta się tworząc prezentacje przy użyciu *Impress* to:

- *Draw* — narzędzie do tworzenia grafiki wektorowej i prostych efektów 3D,
- *Calc* — narzędzie do tworzenia arkuszy kalkulacyjnych oraz generowania na ich podstawie wykresów,
- *Math* — narzędzie do tworzenia wzorów matematycznych.

Dla narzędzi *Draw* i *Calc* omówiona zostanie jedynie podstawowa funkcjonalność.

15.2 Kreator prezentacji

Po włączeniu programu *Impress* uruchamia się kreator prezentacji. Jego celem jest stworzenie początkowego dokumentu. Kreator uruchamia się również jeżeli w oknie głównym aplikacji wybierzemy z menu głównego polecenie

Plik ► Nowy ► Prezentacja

lub użyjemy kombinacji klawiszy *Ctrl-N*. Kreator można też w każdej chwili przywołać z menu głównego przy pomocy polecenia

Plik ► Kreatory ► Prezentacja

Aby kreator nie uruchamiał się więcej automatycznie należy zaznaczyć pole wyboru *Nie pokazuj więcej tego okna dialogowego*. Przywrócenie automatycznego uruchamiania kreatora następuje po wybraniu z menu głównego polecenia

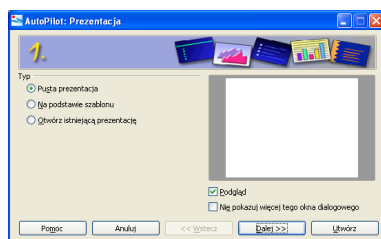
Narzędzie ►Opcje

, wybraniu pozycji

Impress ►Ogólne

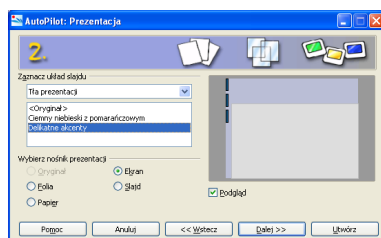
z listy *OpenOffice.ux.pl* oraz zaznaczeniu pola wyboru *Rozpoczynaj pracę z kreatorem*.

15.2.1 Kreator prezentacji — krok 1



Praca kreatora odbywa się w kilku krokach. Ich liczba zależy od typu prezentacji, który wybrano w kroku 1. W każdym momencie można pominąć pozostałe kroki wybierając przycisk *Utwórz*, co jest jednoznaczne z zaakceptowaniem wartości domyślnych. Dalej opiszemy proces tworzenia nowej prezentacji.

15.2.2 Kreator prezentacji — krok 2

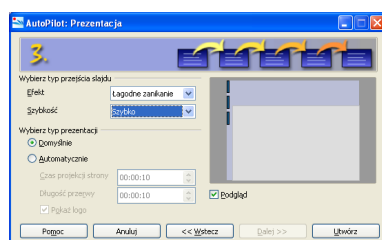


W drugim kroku wybierany jest *Wzorzec stron* (uwaga: kreator zawiera błędną odpowiedź, iż chodzi o *układ slajdu*). Wzorzec określa domyślne elementy i style (m.in. tła i czcionki) dla wszystkich slajdów. Wszystkie slajdy w prezentacji powstają poprzez dodawanie elementów do wzorca stron.

Używana przez nas wersja *Impress* zawiera pięć gotowych wzorców: domyślny dla nowej prezentacji, *Ciemny z niebieskim pomarańczowym*, *Delikatne akcenty*, *Przedstawienie nowego produktu* i *Rekomendacja strategii*. Dwa ostatnie stają się dostępne po zamianie *Prezentacje* na *Tła prezentacji* w liście wyboru.

W tym kroku można również wybrać nośnik prezentacji. Zazwyczaj będzie to ekran (ta opcja obejmuje również projektor multimedialny). Pozostałe nośniki różnią się proporcjami slajdów oraz mogą zawierać marginesy wydruku.

15.2.3 Kreator prezentacji — krok 3

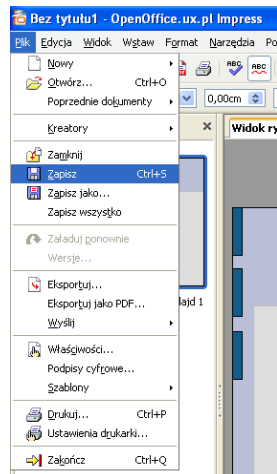


W trzecim kroku określany jest efekt graficzny towarzyszący zmianie wyświetlanego slajdu oraz typ prezentacji:

- domyślnie — zmianę slajdu trzeba zainicjować myszką lub klawiaturą,
- automatycznie — każdy slajd wyświetlany jest przez *Czas projekcji strony* (zmiana na następny następuje automatycznie), a cała prezentacja odtwarzana jest cyklicznie z przerwami trwającymi *Długość przerwy*.

Prezentacje automatyczne są często stosowane na różnego rodzaju stoiskach targowych lub na podobnych imprezach, ale slajdy stanowiące uzupełnienie referatu zazwyczaj przełącza się ręcznie.

15.2.4 Nową prezentację należy od razu zapisać

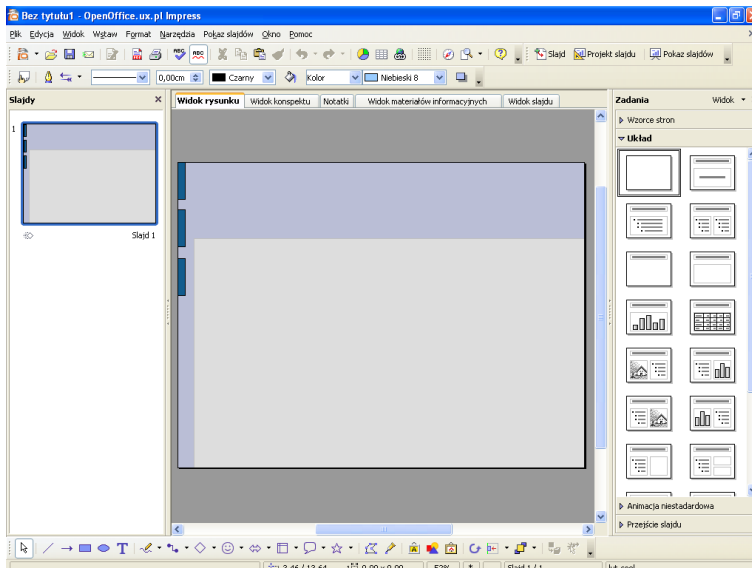


Pierwszą czynnością po utworzeniu nowej prezentacji powinno być zawsze jej zapisanie. Można to uczynić wybierając polecenie

Plik ► Zapisz

lub używając kombinacji klawiszy *Ctrl-S*. Warto wyrobić sobie nawyk częstego zapisywania postępów prac.

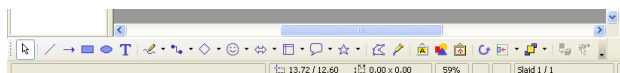
15.3 Podstawowe możliwości OpenOffice Impress



Ekran główny *Impress* podzielony jest na kilka części. Niektóre z pasków narzędzi znajdujących się na górze ekranu ulegają zmianie w zależności od elementów slajdu, które obecnie zaznaczono. Środkową część okna zajmuje

aktualnie edytowany slajd. Narzędzia dostępne na paskach na dole ekranu pozwalają dodawać do aktualnie edytowanego slajdu pola tekstowe i elementy graficzne. Te same narzędzia są również dostępne w programach *Write* i *Draw*.

15.3.1 Dodawanie obiektów do slajdu



Pola tekstowe i obiekty graficzne:

- dodawanie
- przesuwanie
- wprowadzanie tekstu
- skalowanie
- kopiowanie
- menu kontekstowe

Każda z ikon na pasku posiada podpowiedź, która jest wyświetlana po zatrzymaniu nad nią nieruchomo kursora myszy. Obiekty odpowiadające aktywnemu narzędziu umieszcza się zaznaczając przy pomocy myszy obszar na slajdzie, w którym mają się znaleźć. Po prawej stronie niektórych ikon znajduje się czarny trójkąt, który umożliwia wybór innego kształtu z dostępnej palety. Jeżeli którąś z ikon uaktywniono przez pomyłkę, to można dodać obiekt po czym go skasować, bądź zrezygnować z dodawania obiektu uaktywniając ikonę przedstawiającą kursor myszy lub używając klawisza *Esc*. Posługując się przyciskiem *Fontwork – Galeria* można wstawić tekst z interesującymi efektami graficznymi, a używając przycisku *Z pliku* wstawić grafikę zapisaną w pliku dyskowym.

Wstawione obiekty dają się zaznaczać myszką oraz przesuwać przy pomocy myszki (przeciągnij i upuść) lub klawiatury (strzałki). Po wybraniu pola tekstowego w jego obrębie umieszczany jest kursor i można wprowadzać tekst. Podczas edycji tekstu górny pasek narzędziowy dostosowuje się automatycznie dając możliwości znane z programu *Write*. Tekst można również dodawać do obiektów graficznych. Kursor pojawia się w nich po podwójnym kliknięciu.

Zaznaczone obiekty można skalować przeciągając rozmieszczone na ich konturze kwadraciki. Aby skalowany obiekt nie stracił swoich proporcji należy w trakcie rozciągania przytrzymać klawisz *Ctrl*.

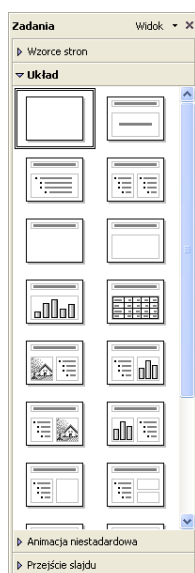
Obiekty można kopiować, wklejać i wycinać (przy pomocy menu *Edycja*, menu kontekstowego oraz skrótów klawiszowych). W obu wypadkach menu kontekstowe daje dostęp do wielu przydatnych właściwości i narzędzi.

Panel slajdy

Po lewej stronie znajduje się panel *Slajdy* zawierający miniaturki slajdów, z których składa się prezentacja. Miniaturka aktualnie edytowanego slajdu jest wyróżniona pogrubioną krawędzią. Zmiana aktualnego slajdu możliwa jest przy pomocy myszki lub klawiszy *PageUp* i *PageDown*. Korzystając z menu kontekstowego dostępnego pod prawym klawiszem myszki można dodawać nowe slajdy lub usuwać już istniejące. Analogicznie można też całe slajdy kopiować, wycinać i wklejać. Prosty sposób na zmianę kolejności slajdów jest przeciągnięcie miniaturki lub grupy miniaturek na inne miejsce.

Slajdy można również dodawać i duplikować za pośrednictwem menu *Wstaw*. Dodatkowo w prawym górnym rogu ekranu znajduje się przycisk *Slajd*, który umożliwia dodawanie slajdów na jeszcze jeden sposób.

15.3.2 Gotowe układy slajdu



Domyślnie aktywna zakładka *Układ* znajdującego się po prawej stronie panelu *Zadania* zawiera miniaturki z najczęściej używanymi układami slajdów. Wybranie jednej z miniaturek powoduje dodanie do aktualnie edytowanego slajdu wszystkich elementów układu jeszcze się na tym slajdzie nie znajdujących. Warto zwrócić uwagę, że ustawienia stylu obiektów dodawanych w ten sposób (np. pól tekstowych) są określone przez wzorzec stron użyty w danej prezentacji. Oprócz pól tekstowych, dostępne układy pozwalają umieścić na slajdzie grafikę zapisaną w pliku dyskowym, wykresy i arkusze. Zazwyczaj slajdy wypełnia się właśnie używając tych układów.

15.3.3 Dostosowywanie paneli do swoich potrzeb

Panele *Slajdy* i *Zadania* można ukryć i przywołać używając kropczek znajdujących się w połowie wysokości ramki, która odgranicza je od środkowej

części okna z edytowanym slajdem. Chwytając ramkę odgraniczającą poza kropczkami można również rozciągnąć lub zwięzić panele.

15.3.4 Pokaz slajdów, zapisywanie i eksportowanie

- Uruchamianie pokazu slajdów
- Zapisywanie i wczytywanie pokazu slajdów
- Eksportowanie slajdów
 - PDF
 - HTML
 - pliki graficzne

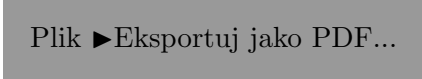
Pokaz slajdów uruchamia się poleceniem



Pokaz slajdów ► Pokaz slajdów

bądź jednym z klawiszy *F5* lub *F9*. Dodatkowo w prawym górnym rogu ekranu znajduje się przycisk *Pokaz slajdów*, który umożliwia rozpoczęcie pokazu na jeszcze jeden sposób.

Pokazy slajdów zapisywane są w plikach OpenDocument (ODP). Od maja 2006 ten format plików jest standardem International Standards Organization (ISO). Nie należy się jednak w najbliższym czasie spodziewać, że dokumenty zapisane w ten sposób będzie można otworzyć na komputerach, na których nie zainstalowano *OpenOffice* w wersji co najmniej 2.0. Dlatego dość popularne jest eksportowanie pokazów slajdów do dokumentów Portable Document Format (PDF), które można odczytać przy pomocy popularnego programu Adobe Reader. Zapisanego w ten sposób pokazu slajdów nie można jednak już więcej modyfikować. Podczas eksportowania traczone są ponadto niektóre elementy multimedialne prezentacji, jak animacje obiektów na slajdach czy efekty muzyczne. Aby wyeksportować pokaz slajdów do formatu PDF należy użyć ikony *Eksportuj bezpośrednio jako PDF* ze *Standardowego* paska narzędzi, wybrać polecenie



Plik ► Eksportuj jako PDF...

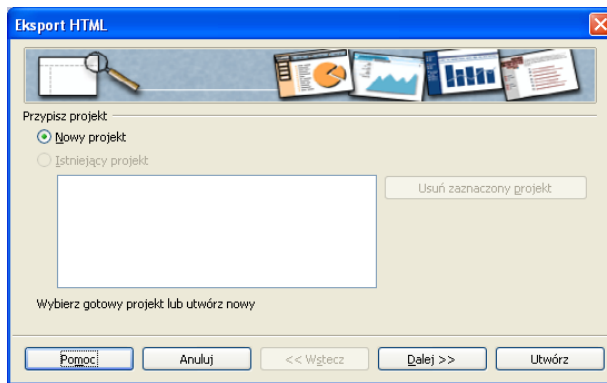
albo



Plik ► Eksportuj...

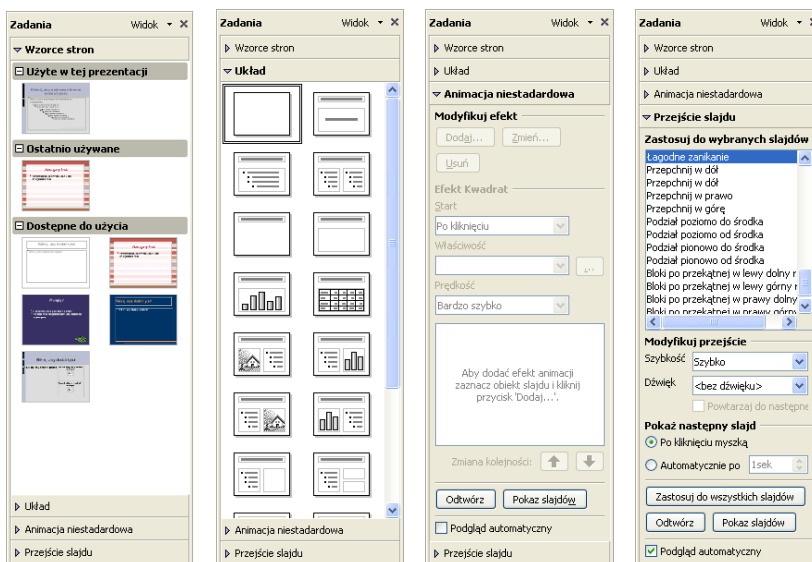
i określić PDF jako format pliku.

Ciekawą możliwością jest również wyeksportowanie pokazu slajdów jako dokumentu HTML, bądź jako zbioru grafik przedstawiających poszczególne slajdy (np. w formatach PNG lub JPG).



15.4 Dalsze możliwości OpenOffice Impress

15.4.1 Panel zadania



Znajdujący się po prawej stronie okna głównego panel *Zadania* posiada cztery zakładki. Domyślnie aktywna jest zakładka *Układ*.

Wzorce

Zakładka *Wzorce* umożliwia zmianę wzorca stron dla wszystkich slajdów znajdujących się w prezentacji. Aby zmienić wzorec tylko dla aktywnego slajdu, należy z menu *Format*, menu kontekstowego miniaturki slajdu w panelu *Slajdy* lub podmenu *Slajd* menu kontekstowego aktywnego slajdu wybrać polecenie *Projekt slajdu* i wskazać jedną z miniaturki przedstawiających wzorce używane w prezentacji bądź przy pomocy przycisku *Zaladuj...* wybrać nowy wzorec.

Animacja niestandardowa

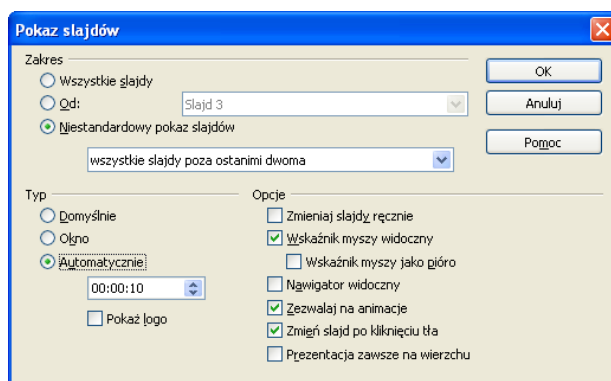
Zakładka *Animacja niestandardowa* pozwala przypisywać animacje do obiektów znajdujących się na slajdzie (przytrzymując klawisz *Ctrl* można zaznaczyć kilka obiektów naraz) lub do zaznaczonych fragmentów tekstu. Należy w tym celu zaznaczyć obiekt/y (np. pole tekstowe) lub fragment tekstu, użyć przycisku *Dodaj* i z okienka dialogowego, które się pojawi wybrać rodzaj animacji (okienko ma cztery zakładki). Dla każdego slajdu można zdefiniować wiele animacji niestandardowych. Tak samo dla pojedynczego obiektu lub fragmentu tekstu. Domyślnie, podczas pokazu slajdów, animacje będą pokazywane w kolejności ich dodawania, a każda kolejna będzie rozpoczynana po kliknięciu. Kolejność, sposób rozpoczęcia (po kliknięciu, razem z poprzednim oraz po poprzednim) oraz moment startu dla każdej animacji można określić korzystając z pozostałych elementów zakładki *Animacja niestandardowa*, które stają się aktywne po wskazaniu animacji na liście znajdującej się po środku zakładki.

Uwaga: Animacje doskonale nadają się do przyciągnięcia uwagi, ale użyte w nadmiarze mogą również rozpraszać.

Przejście slajdu

Zakładka *Przejście slajdu* umożliwia określenie efektu graficznego towarzyszącego zmianie wyświetlanego slajdu. Efekt zaznaczony na liście staje się od razu przypisany dla aktywnego slajdu. Przy pomocy przycisku *Zastosuj do wszystkich slajdów* można dany efekt przypisać do wszystkich slajdów w prezentacji. Efekt można też przypisać do kilku wybranych slajdów. Trzeba w tym celu przytrzymując wciśnięty klawisz *Ctrl* zaznaczyć kilka miniatur w panelu *Slajdy* i wskazać efekt na liście. Poza rodzajem efektu graficznego można też ustalić jego szybkość, skojarzyć z przejściem efekt dźwiękowy oraz ustalić kiedy następuje przejście slajdu: po kliknięciu czy po upływie zadanego czasu.

15.4.2 Ustawienia pokazu slajdów



Polecenie *Ustawienia pokazu slajdów...* z menu *Pokaz slajdów* przywołuje

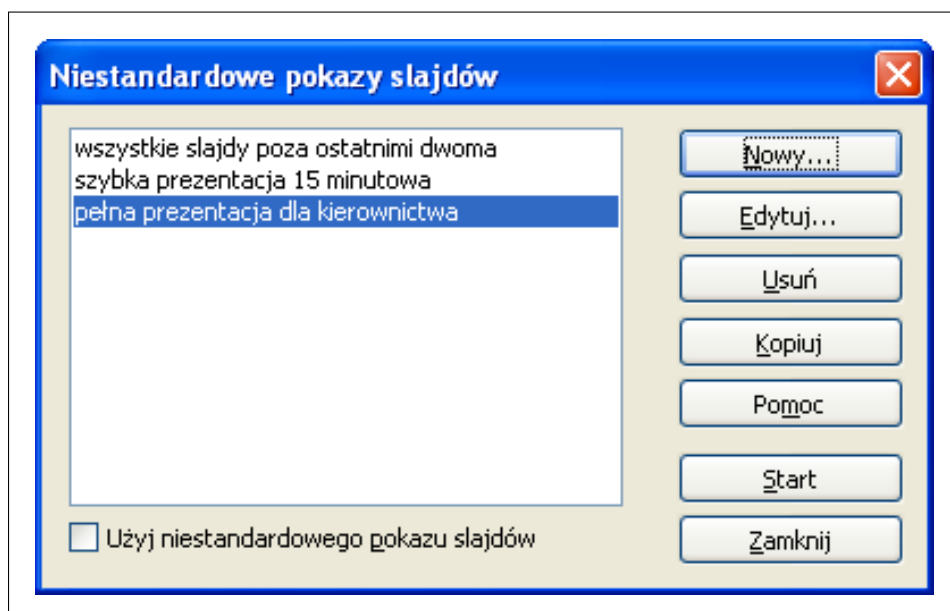
okno, w którym można zmienić ustawienia pokazu slajdów. Do wyboru są trzy typy pokazu:

- *Domyślnie* — slajdy wyświetlane są na całym ekranie, a prezentacja kończy się po opuszczeniu ostatniego slajdu,
- *Okno* — podczas wyświetlania slajdów widoczne jest menu główne *Impress* oraz paski narzędziowe, a prezentacja kończy się po opuszczeniu ostatniego slajdu,
- *Automatycznie* — slajdy wyświetlane są na całym ekranie, a po opuszczeniu ostatniego slajdu i odczekaniu zadanego czasu pokaz zaczyna się od nowa.

Zaznaczając lub odznaczając odpowiednie pola wyboru można między innymi ustalić:

- czy podczas pokazu będzie widoczny kursor myszy,
- czy użyć kursora myszy w roli piórka, którym daje się rysować po wyświetlanych slajdach,
- czy wyświetlane będą animacje niestandardowe,
- czy okienkom innych programów wolno przykryć wyświetlane slajdy.

Niewyświetlanie wszystkich slajdów podczas pokazu



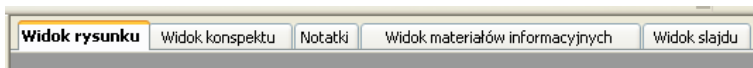
Niestandardowe pokazy slajdów

Często jeden zestaw slajdów jest używany na wielu pokazach. W zależności od ilości czasu przeznaczanego na prezentację oraz docelowej grupy słuchaczy można zdecydować się na pokazanie jedynie podzbioru slajdów. Wybór slajdów do pokazania odbywa się w części *Zakres* okienka *Ustawienia pokazu slajdów*. Najciekawsza jest ostatnia z trzech dostępnych opcji *Niestandardowy pokaz slajdów*, którą można wybrać dopiero po zdefiniowaniu przynajmniej jednego niestandardowego pokazu. W tym celu należy się posłużyć oknem dialogowym przywoływanym z menu głównego poleceniem *Pokaz slajdów ▶ Niestandardowy pokaz slajdów*. Domyślnie nazwy slajdów odzwierciedlają ich kolejność w prezentacji. Zmiana kolejności istniejących slajdów lub wstawienie nowych slajdów pomiędzy już istniejące spowoduje przemianowanie slajdów z domyślnymi nazwami. Zmiana nazw slajdów nie ma na szczęście wpływu na zawartość pokazów niestandardowych (zdefiniowane pokazy będą zawierały te same slajdy co wcześniej pomimo, że ich nazwy się zmieniły) ale dobrym pomysłem jest nadanie slajdom nazw odzwierciedlających ich zawartość, gdyż ułatwi to definiowanie i modyfikowanie pokazów niestandardowych. Nazwę slajdu można zmienić odnajdując jego miniaturkę w panelu *Slajdy* i wybierając z menu kontekstowego polecenie *Zmień nazwę slajdu*.

W programie *Impress* istnieje jeszcze jeden mechanizm pozwalający nie pokazywać wszystkich slajdów — ukrywanie. W celu ukrycia slajdu należy odnaleźć jego miniaturkę w panelu *Slajdy* i z menu kontekstowego wybrać polecenie *Pokaż/ukryj slajd*. Znajdujący się obok lewego górnego rogu miniaturki numerki slajdu zostanie wtedy otoczony ramką i przekreślony, a slajd nie będzie wyświetlany, nawet będąc wskazanym w ustawieniach pokazu slajdów. Przywrócenie ukrytego slajdu odbywa się dokładnie w ten sam sposób.

Ponieważ przy edycji slajdów często sprawdza się efekty prac (np. wygląd animacji niestandardowych), pokaz slajdów rozpoczyna się od slajdu aktualnie edytowanego. Jest to duże udogodnienie, gdyż nie trzeba przeglądać za każdym razem wszystkich wcześniejszych slajdów. Jeżeli z ustawień pokazu slajdów wynika, że aktualnie edytowany slajd ma nie być pokazywany, to pokaz zacznie się od pierwszego slajdu zakresu występującego po slajdzie aktualnie edytowanym. Jeżeli natomiast aktualnie edytowany slajd jest ukryty, to pokaz rozpocznie się od początku.

15.4.3 Widoki

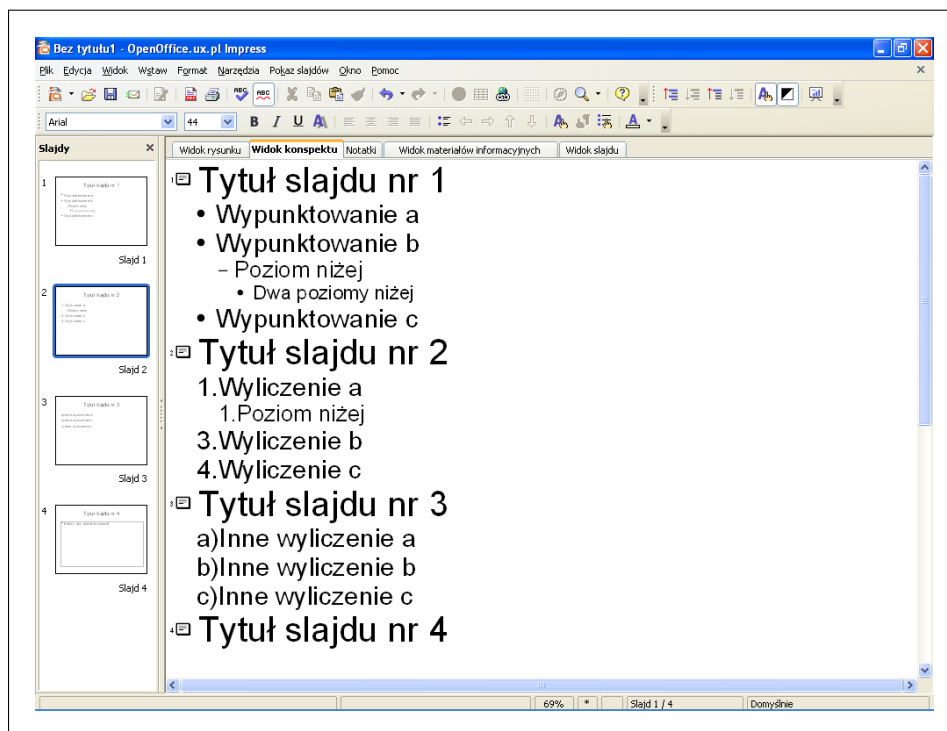


- Widok konspektu
 - poziom listy
 - wypunktowania i numerowanie
 - wyłączanie formatowania i widok czarno-biały
- Notatki

- drukowanie
- Widok materiałów informacyjnych
 - zmiana układu
 - drukowanie
- Widok slajdu
 - zmiana kolejności slajdów
 - ukrywanie slajdów
 - określanie efektów przejść slajdów

Pracując w programie *Impress* można wybrać jeden z pięciu widoków. Każdy z widoków jest zaprojektowany tak, by ułatwić wykonywanie pewnych zadań. Do formatowania slajdów, dodawania tekstu, grafiki oraz animacji służy *Widok rysunku*. Jest to domyślny widok po utworzeniu nowej prezentacji i to on dotychczas był omawiany w tych materiałach. Aby przełączyć się między widokami należy użyć zakładek znajdujących się nad obszarem edycji aktualnego slajdu, bądź menu poleceń z menu głównego *Widok*.

Widok konspektu

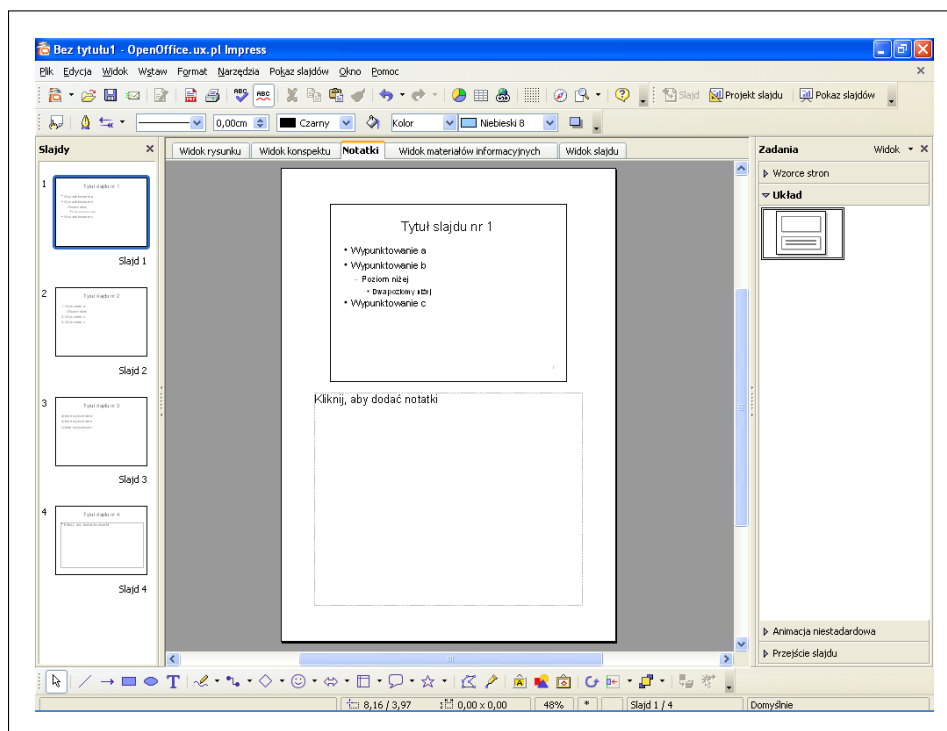


Widok konspektu

Widok konspektu pozwala wygodnie wprowadzać tytuły slajdów oraz tworzyć na poszczególnych slajdach wypunktowania i wyliczenia. Łatwo można

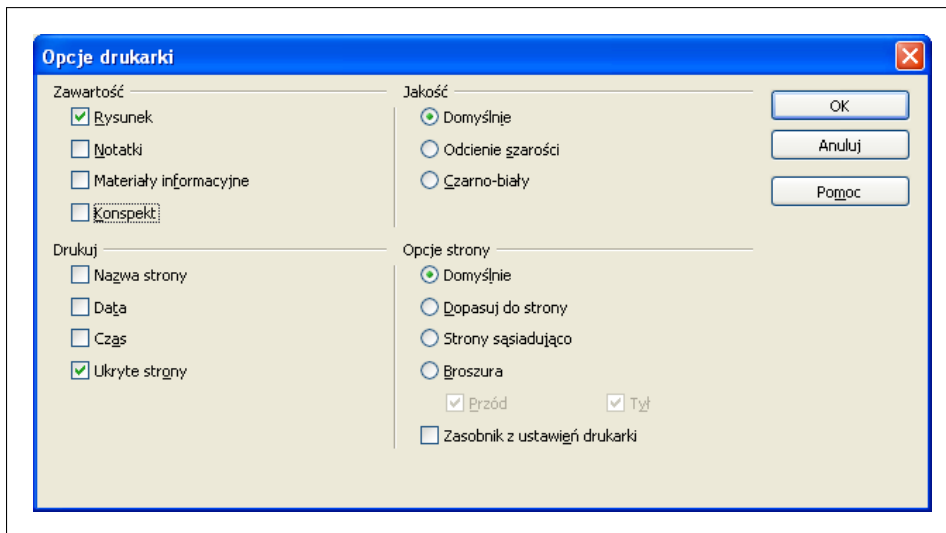
również dodawać nowe slajdy, zmieniać kolejność slajdów oraz zmieniać kolejność elementów wypunktowań i wyliczeń. Przy pomocy przycisków *Poziom wyżej* i *Poziom niżej* z górnego paska narzędzi można zmieniać poziom listy, a przy pomocy przycisków *Przesuń w górę* i *Przesuń w dół* można przesuwać elementy względem siebie. Jeżeli, ze względu na style wybranego wzorca stron, tekst jest mało czytelny, można skorzystać ze znajdujących się w prawym górnym rogu przycisków *Wyłącz/Włącz formatowanie* oraz *Widok czarno-biały*. Ich użycie nie wpływa na zawartość slajdów, a jedynie na sposób wyświetlania tekstu w tym widoku.

Notatki



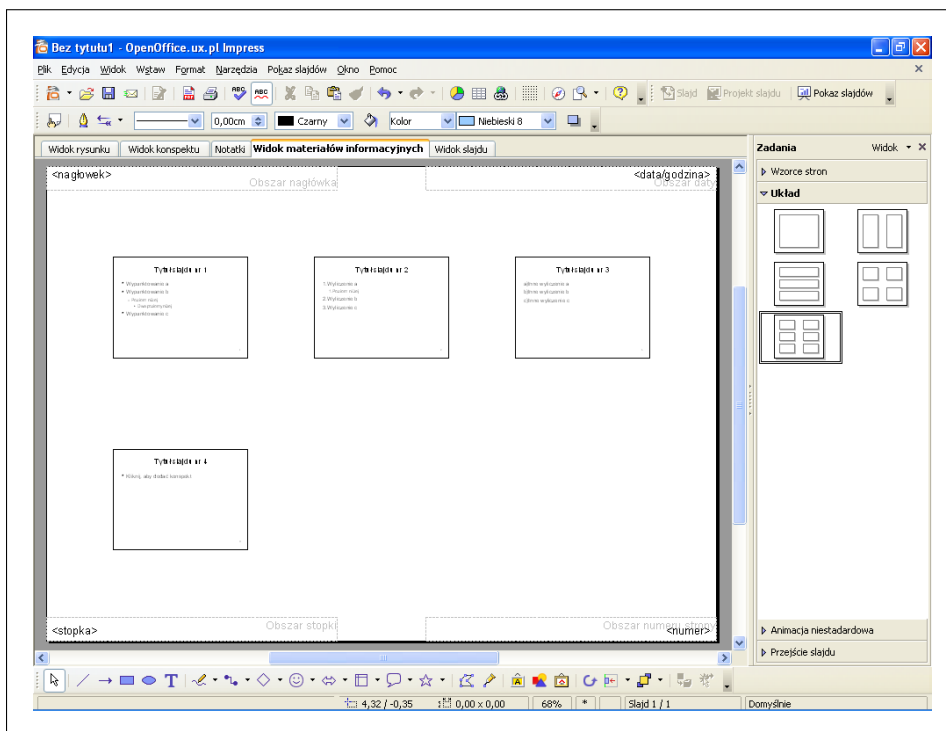
Widok notatki

Widok notatki służy do dodawania notatek, które nie są wyświetlane podczas pokazu slajdów. Sporządzone w ten sposób notatki można przejrzeć przed prezentacją lub wydrukować. Aby wydrukować notatki należy przy pomocy przycisku *Opcje* z dialogu drukowania przywołać okienko *Opcje drukarki* i zaznaczyć odpowiednie pole wyboru.



Drukowanie

Widok materiałów informacyjnych

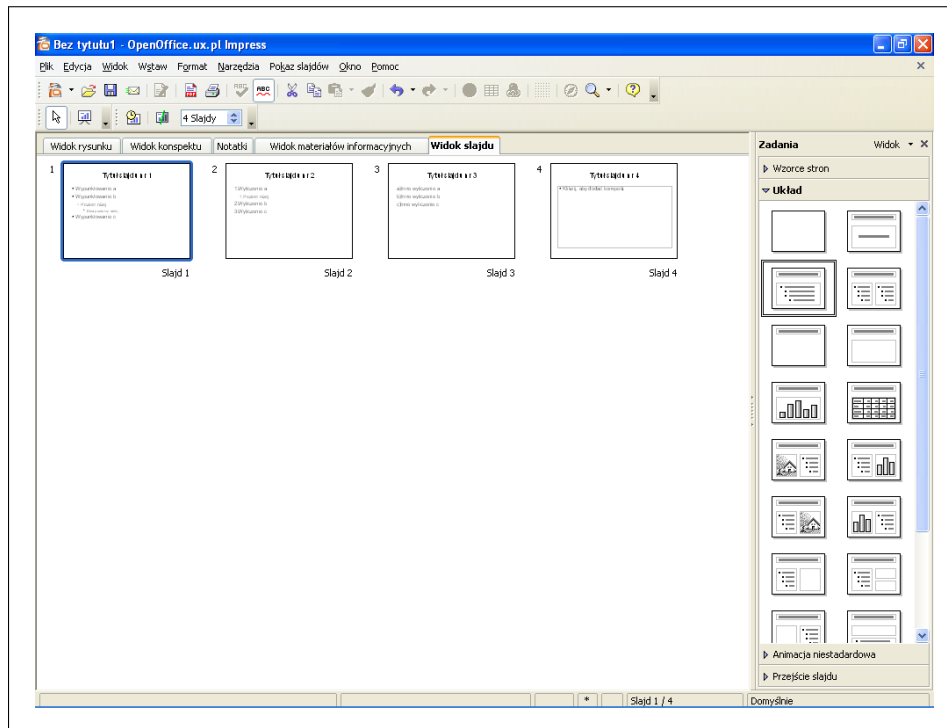


Widok materiałów informacyjnych

Widok materiałów informacyjnych pozwala zaprojektować, w jaki sposób ze slajdów przygotować gotowe do wydruku materiały informacyjne. Do wyboru jest pięć układów, które można modyfikować przeciągając slajdy przy pomocy myszki. Aby wydrukować materiały informacyjne należy przy pomocy przycisku *Opcje* z dialogu drukowania przywołać okienko

Opcje drukarki i zaznaczyć odpowiednie pole wyboru.

Widok slajdu



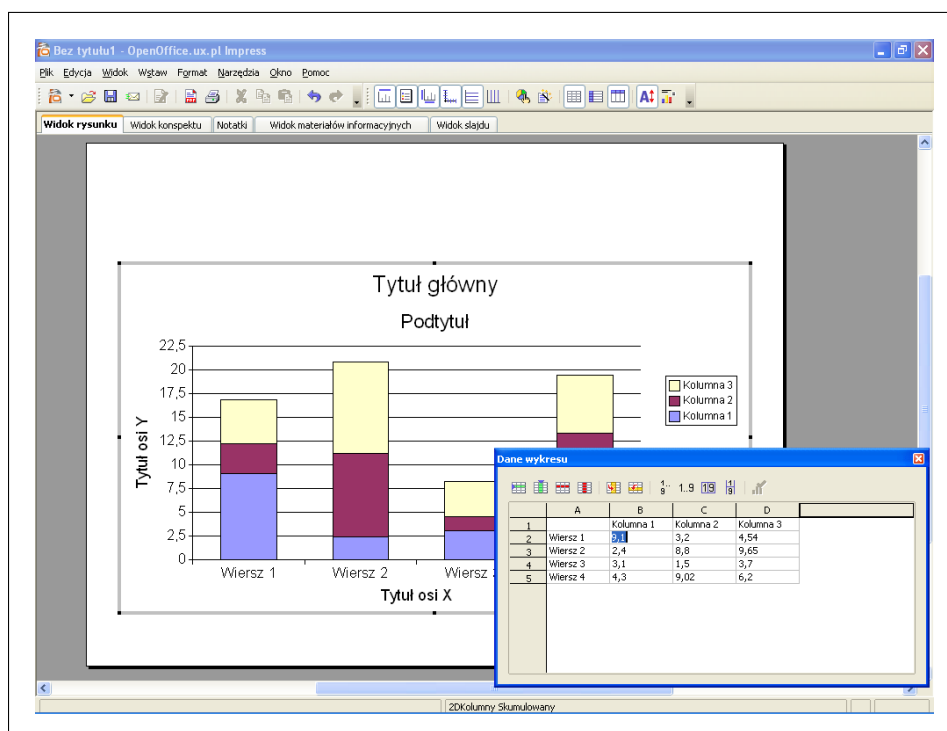
Widok slajdu

Widok slajdu przedstawia wszystkie slajdy w postaci miniatur. Liczbę miniatur wyświetlanych w rzędzie można zmienić przy pomocy licznika znajdującego się na górnym pasku narzędziowym. Pracując w tym widoku, łatwo można zmieniać kolejność slajdów oraz ukrywać i przywracać slajdy. Jest on również zalecany do ustawiania przejść między slajdami. Przy pomocy przycisku *Z pomiarem czasu* można uruchomić próbny pokaz slajdów, w którym Impress będzie odliczał czas wyświetlania każdego slajdu. Ta funkcja jest szczególnie przydatna przy szacowaniu czasu wyświetlania poszczególnych slajdów, jeżeli przejścia między slajdami mają następować automatycznie.

15.4.4 Wstawianie wykresów

- Wstawianie wykresów
- Edytowanie danych
- Typy wykresów
- Formatowanie elementów wykresów

Niektóre z predefiniowanych układów zawierają obszar wykresu. Po podwójnym kliknięciu w obszarze wykresu *Impress* przełącza się tryb edycji wykresu.



Tryb edycji wykresu

Aby opuścić tryb edycji wykresu wystarczy kliknąć gdzieś na slajdzie poza obszarem wykresu lub użyć klawisza *Esc*. Początkowo obszar wykresu jest niewypełniony, ale przy pierwszej próbie edycji zostanie zainicjowany przykładowym wykresem kolumnowym.

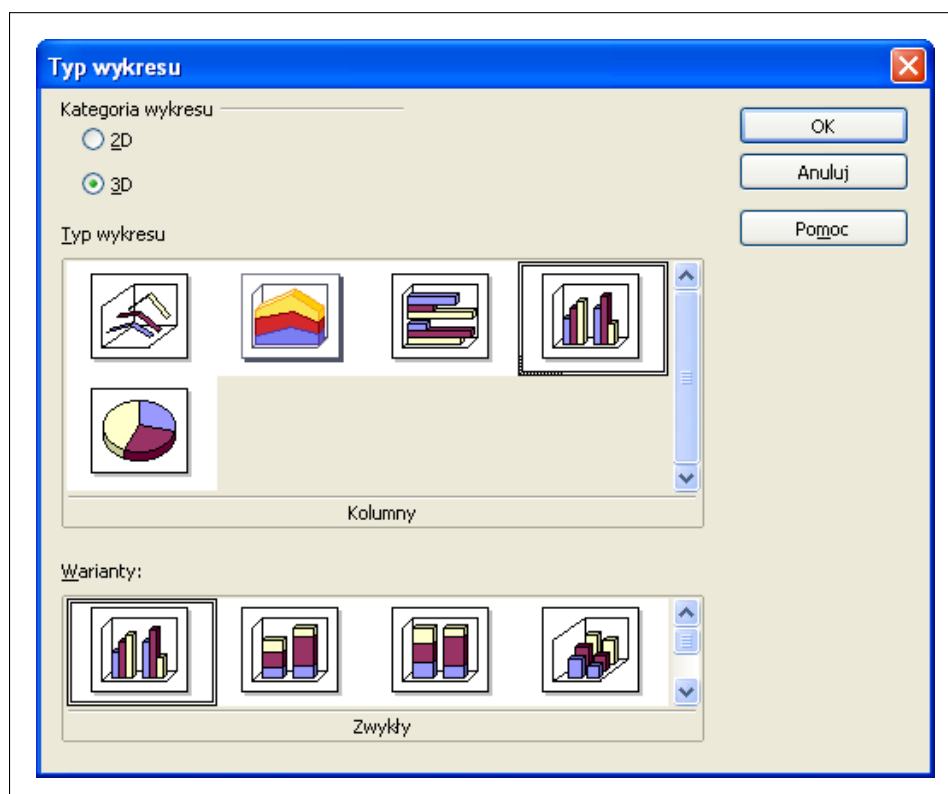
Dane wykresu

Przy pomocy okna dialogowego *Dane wykresu* można edytować dane, które przedstawia wykres. Okno to daje się przywołać w trybie edycji wykresu przy pomocy jednego z przycisków na górnym pasku narzędzi, menu kontekstowego obszaru wykresu bądź polecenia z menu głównego *Edycja*.

Ustawienia wykresu

Przyciski umieszczone na górnym pasku narzędzi pozwalają zdecydować czy wykres ma zawierać niektóre opcjonalne elementy, takie jak tytuł, legendę, opisy osi, legendę osi, siatkę poziomą czy siatkę pionową. Edycja tekstu w tytule, legendach i opisach jest możliwa po wybraniu tych obiektów w obszarze wykresu. Nazwy wierszy i kolumn można zmienić w oknie dialogowym *Dane wykresu*.

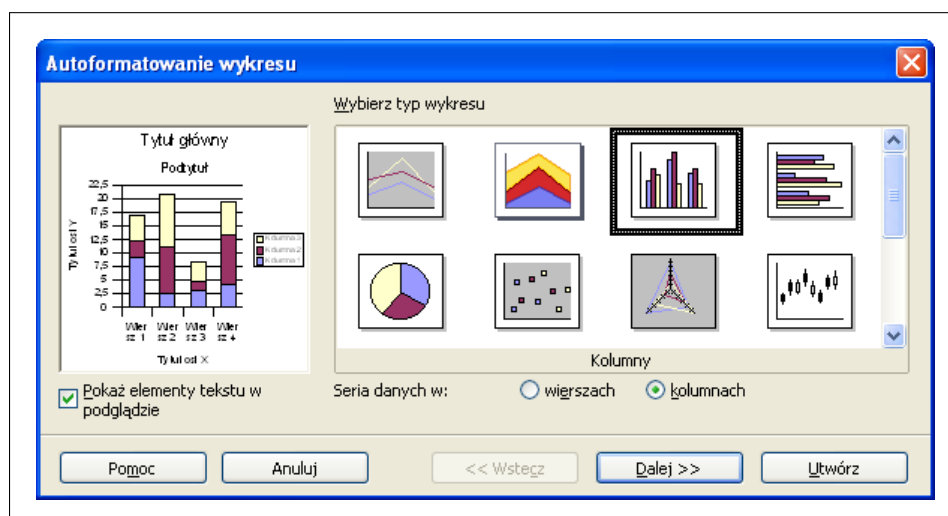
Typy wykresów



Typy wykresów

Przy pomocy jednego z przycisków na górnym pasku narzędzi, menu kontekstowego obszaru wykresu bądź polecenia z menu głównego *Edycja* można przywołać okno dialogowe *Typy wykresów*. Poza domyślnym typem kolumnowy do wyboru jest jeszcze siedem typów dwuwymiarowych oraz pięć typów trójwymiarowych. Wykresy każdego typu występują w kilku wariantach.

Autoformatowanie



Autoformatowanie wykresów

W ten sam sposób, co okno dialogowe *Typy wykresów*, można przywołać kreator *Autoformatowanie wykresu*. Przy jego pomocy można w wygodny sposób dokonać omówionych dotychczas ustawień, takich jak typ i wariant wykresu. Można też zdecydować, czy wyświetlać opcjonalne elementy slajdu, a jeżeli tak, to jaki powinny zawierać tekst.

Ręczne wstawianie wykresów

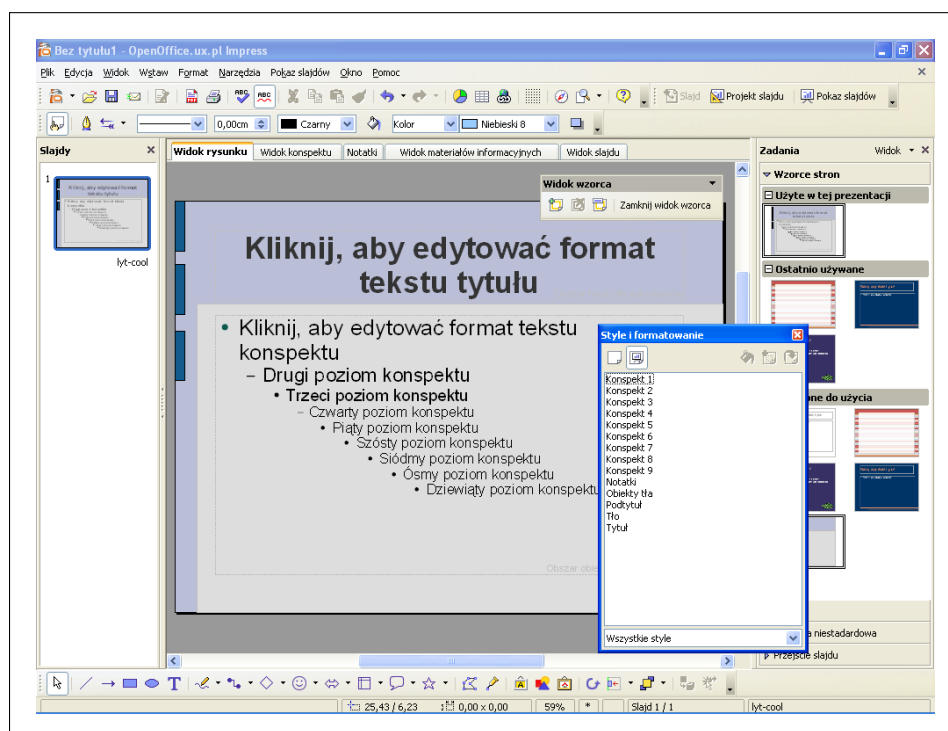
Obszar wykresu można też umieścić na slajdzie bez użycia predefiniowanych układów. Służy do tego polecenie *Wstaw ► Wykres* oraz przycisk *Wykres* z górnego paska narzędzi.

15.4.5 Edycja wzorca stron

- Edycja wzorca
- Edycja stylu
- Wstawianie daty, godziny i numeru slajdu
- Nagłówki i stopki
- Zapisywanie i używanie szablonów

W obecnej wersji *Impress* zawiera jedynie pięć predefiniowanych wzorców stron, w tym jeden domyślny dla pustej prezentacji. Jak je wybrać dla wszystkich, bądź dla pojedynczego slajdu wyjaśniono w podrozdziale 15.4.1. Jest jednak możliwość tworzenia nowych wzorców, edycji już istniejących oraz używania wzorców przygotowanych przez inne osoby.

Edycja wzorca stron



Edycja wzorca

Edycja wzorca stron odbywa się w specjalnym widoku, na który można się przełączyć przy pomocy polecenia

Widok ► Wzorzec ► Wzorzec slajdu

. Do opuszczenia *Widoku wzorca* służy przycisk *Zamknij widok wzorca* na pływającym pasku narzędzi, który jest widoczny w tym widoku. Wzorzec slajdu zawiera definicje stylów oraz obiekty powtarzane na każdym slajdzie. Umieszczając we wzorcu jakiś obiekt, np. pole z tekstem, sprawimy, że będzie on widoczny na wszystkich slajdach.

Edycja stylów

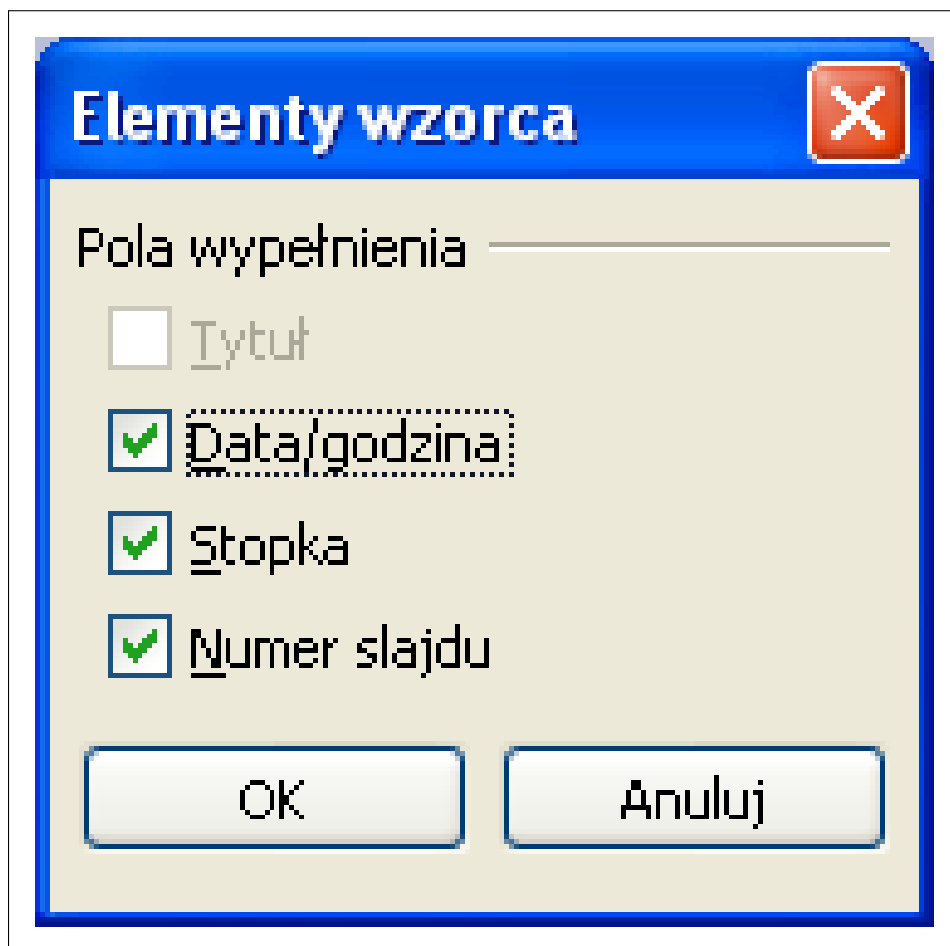
Format tekstu tytułu, tekstu konspektu oraz ustawienia tła dla każdego slajdu są kopiowane ze wzorca stron. Modyfikacje formatów dokonywane w *Widoku rysunku* dotyczą aktualnie wybranego slajdu, ale pozostałych nie. Modyfikacje dokonywane w *Widoku wzorca* dotyczą wszystkich slajdów, dla których wybrano dany wzorzec.

Ustawienia tła można zmieniać w zakładce *Tło* okienka *Ustawienia strony*, które przywołuje się poleceniem

z menu kontekstowego obszaru, w którym wyświetlany jest slajd/wzorzec. *Impress* oferuje pięć możliwych ustawień tła, między którymi można się przełączać przy pomocy rozwijanej listy wyboru. Po zmianie ustawień tła dla pojedynczego slajdu otrzymujemy możliwość zaakceptowania ustawień dla wszystkich slajdów.

Zmieniając format tekstu tytułu i tekstu konspektu w *Widoku wzorca* naprawdę modyfikujemy style używane w prezentacji. Style można również modyfikować przy pomocy okienka *Style i formatowanie* dostępnego we wszystkich programach pakietu *OpenOffice*. Okienko to można przywołać dla dowolnym widoku klawiszem *F11*.

Daty, godziny i numery slajdów



Elementy wzorca

Wzorzec strony zawiera pięć obiektów o specjalnym znaczeniu. Są to: pola z tekstem tytułu slajdu, z tekstem konspektu, z datą i godziną, z tekstem

stopki i numerem slajdu. Obiektów tych nie można usunąć, jednak pola data/godzina, stopka i numer można ukryć odznaczając odpowiadające im pola wyboru w okienku *Elementy wzorca*. Okienko to przywołuje się poleceniem

Widok ► Wzorzec ► Elementy wzorca...

z menu głównego.

Jeżeli pola data/godzina, stopka, i numer nie są ukryte we wzorcu, można zdecydować o ich wyświetlaniu na poszczególnych lub wszystkich slajdach oraz uzupełnić ich treść przy pomocy okna przywoływanego poleceniem *Widok ► Nagłówek i stopka*.

W stopce można umieścić informacje o autorze prezentacji, a widoczny numer slajdu ułatwia robienie notatek osobom uczestniczącym w prezentacji, dlatego te pola są często używane w prezentacjach. Ukrycie tych pól można rozważyć na slajdach zawierających grafikę, bądź wypełnionych zawartością aż po krawędzie.

Efekty analogiczne do użycia obiektów specjalnego znaczenia można uzyskać wstawiając obiekty dostępne w menu głównym *Wstaw ► Pola*. Obiekty te można używać zarówno w *Widoku rysunku* jak i *Widoku wzorca*, oczywiście z różnym efektem.

Zapisywanie i używanie szablonów

Osoby i organizacje przygotowujące wiele prezentacji powinny rozważyć nadanie im wspólnego wyglądu. Należy w tym celu przygotować szablon, który będzie stanowił punkt wyjściowy dla tworzonych prezentacji. Przygotowanie szablonu polega na utworzeniu nowej prezentacji, zmodyfikowaniu ustawień wzorca stron i zapisaniu efektów w pliku typu *Szablon prezentacji OpenDocument (.otp)*. Tak przygotowany szablon można zaimportować do kategorii tła prezentacji przy pomocy okienka *Zarządzanie szablonami*, które przywołuje się poleceniem

Plik ► Szablony ► Zarządzaj...

z menu głównego. Zaimportowany szablon będzie dostępny do użycia w kreatorze prezentacji oraz okienku *Projekt slajdu*, a po pierwszym użyciu w danej prezentacji również w panelu *Wzorce stron*.

Wiele gotowych szablonów prezentacji można bezpłatnie ściągnąć ze strony <http://ooextras.sourceforge.net/downloads/simpres/>. Pomimo, iż są zapisane w pochodzących z poprzedniej wersji OpenOffice plikach o rozszerzeniu *(.sti)*, można je zaimportować tą samą metodą.

Rozdział 16

Wprowadzenie do systemu USOS. Zasoby BUW dostępne drogą elektroniczną

16.1 Wprowadzenie

Uniwersytecki System Obsługi Studiów jest aplikacją, która powstała i jest rozwijana na Wydziale Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego. W dalszej części rozdziału w odniesieniu do ww. systemu będzie stosowana skrócona (powszechnie stosowana) nazwa — USOS. System USOS umożliwia całkowitą obsługę procesu kształcenia na dowolnym wydziale dowolnej uczelni.

W niniejszym rozdziale zostaną również omówione zagadnienie związane z dostępem do zasobów Biblioteki Uniwersytetu Warszawskiego drogą elektroniczną.

Niniejszy moduł pozwoli *aktywnie studiującej* osobie osiąść umiejętności pozwalające na:

- korzystanie z zasobów internetowych Uniwersytetu Warszawskiego
 - USOS
 - BUW

16.2 Uniwersytecki System Obsługi Studiów

16.2.1 Architektura systemu USOS

System USOS składa się z:

- Głównej bazy danych

- Aplikacji komputerowej dla dziekanatów
- Aplikacji web dla studentów oraz nauczycieli akademickich — USOSweb

W niniejszym rozdziale będzie mowa jedynie o części webowej USOSa tzw. systemie USOSweb w wersji używanej na Uniwersytecie Warszawskim. (USOS jest wykorzystywany na wielu polskich uczelniach wyższych). Niniejszy rozdział będzie pomocny dla użytkownika systemu USOSweb będącego studentem. Nie zostaną tu opisane elementy interfejsu będące częścią systemu przeznaczoną dla nauczyciela akademickiego. Należy wyjaśnić, iż wygląd oraz dostępne opcje interfejsu USOSweba zależą od tego, kto jest aktualnie zalogowany. Jednak zarówno student, jak i nauczyciel korzystają z tej samej aplikacji web (w przypadku Wydziału Matematyki, Informatyki i Mechaniki dostępny pod adresem:

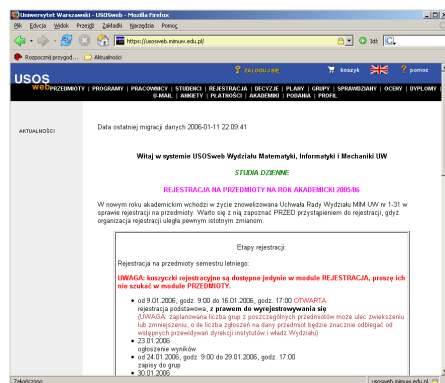
<http://usosweb.mimuw.edu.pl>

16.2.2 Dostęp do aplikacji USOSweb

- Aby wyświetlić aplikację USOSweb wpisujemy w przeglądarce jej adres, np: <http://usosweb.mimuw.edu.pl>
- Adres ten obowiązuje jedynie dla Wydziału Matematyki, Informatyki i Mechaniki
- Np. dla Wydziału Fizyki jest to adres: <http://usosweb.fuw.edu.pl>
- Systemu USOSweb można używać z dowolnego miejsca, np. z domu

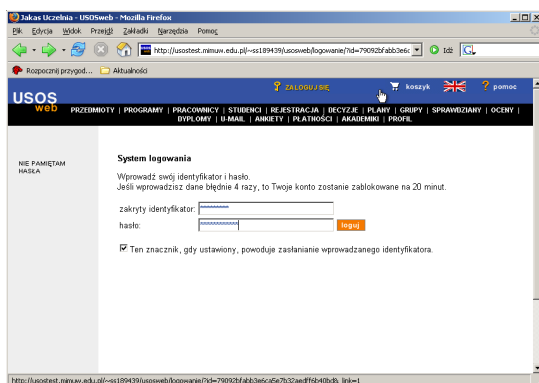
Jak widać, aplikacja USOSweb jest dostępna za pomocą dowolnej przeglądarki internetowej. Informację o adresie URL, pod którym jest dostępna aplikacja USOSweb, można uzyskać na każdym z wydziałów używających systemu USOS.

16.2.3 Główne okno aplikacji USOSweb



- Bez zalogowania, większość pozycji w menu nie jest dostępna
- Ale m.in. można przeglądać ofertę dydaktyczną uczelni (pozycja menu: *Przedmioty*)
- Na głównej stronie zwykle umieszczane są bieżące informacje

16.2.4 Logowanie



- Aby zalogować się, należy kliknąć odnośnik *Zaloguj się*
- W oknie logowania należy użyć numeru PESEL i hasła

Każdy student loguje się do systemu USOSweb używając numeru PESEL, oraz hasła, które wybrał rejestrując się jako kandydat na studia w systemie Internetowej Rejestracji Kandydatów na UW. Wszelkie problemy z dostępem do USOSweba rozwiązują sekcje dziekanatów zajmujące się sprawami studenckimi. USOSweb jest systemem modułowym, każdemu z modułów przypisana jest pozycja w głównym menu aplikacji. W dalszej części rozdziału zostanie opisana funkcjonalność każdego z modułów. Jak już wspomniano, dostęp do większości składowych jest możliwy dopiero po autoryzacji. W dalszej części niniejszego rozdziału, system USOSweb będzie opisywany przy założeniu, że dokonano poprawnego zalogowania.

16.2.5 Moduły systemu USOSweb

W niniejszym rozdziale zostaną przedstawione podsystemy USOSweba, tzw. moduły. Bardzo ważną składową jest *Rejestracja*, umożliwia ona rejestrację na przedmioty, musi więc jej używać każdy student. Niewątpliwie warto temu modułowi przyrzeć się nieco dokładniej.

Moduły systemu USOSweb — podstawowe wiadomości

- Moduły to po prostu składowe systemu USOSweb dostępne przez górne menu

- Na poniższych slajdach zostaną pokazane elementy najważniejszych modułów USOSweba
- Aby zobaczyć moduł w przeglądarce, należy kliknąć jego nazwę w górnym menu systemu

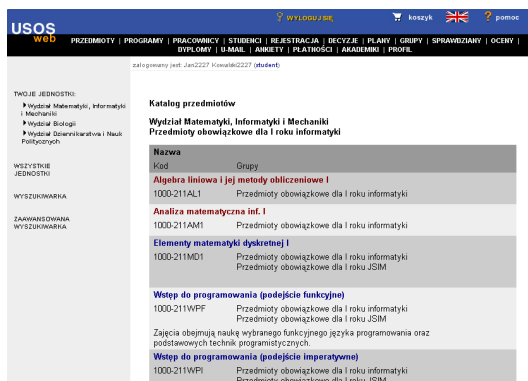
Moduł Przedmioty



- Moduł służy studentom oraz nauczycielom akademickim do przeglądania informacji o oferowanych przez uczelnię przedmiotach
- Są też dostępne wyszukiwarki

Moduł *Przedmioty* służy studentom do przeglądania oferty dydaktycznej uczelni. Można zobaczyć dokładne opisy przedmiotów, sprawdzić kto i kiedy prowadzi zajęcia oraz jakie zajęcia wchodzi w skład przedmiotu. Przedmioty można przeglądać wg oferującej je jednostki dydaktycznej, lub skorzystać z wyszukiwarki, bądź, gdy chcemy podać więcej szczegółów do wyszukiwania — z zaawansowanej wyszukiwarki.

Moduł Przedmioty - wybór przedmiotu



- Wybieramy jednostkę uniwersytecką, a później grupę przedmiotów
- Grupa to np. przedmioty obowiązkowe dla I roku Informatyki

Jeśli klikniemy na nazwę przedmiotu w danej grupie, to zobaczymy szczegóły związane z danym przedmiotem (np. skrócony opis przedmiotu).

Moduł Programy

Przedmioty wymagane

Ta część serwisu prezentuje wymagania przedmiotowe do zaliczenia poszczególnych etapów programu. Zaliczenie warunkowe semestru/roku jest możliwe przy braku zaliczenia przedmiotów oznaczonych zielonym krzyżykiem (pod warunkiem spełnienia wymagań punktowych). Przedmioty te są jednak nadal wymagane i student musi je zaliczyć w ciągu następnego roku studiów.

Informatyka, dzienne magisterskie

Pierwszy semestr informatyki pierwszego roku

nazwa przedmiotu	wymagany do zaliczenia warunkowego
Algebra liniowa i jej metody obliczeniowe I	X
Analiza matematyczna inf. I	X
Elementy matematyki dyskretnej I	✓
Wstęp do programowania (podjęcie imperatywne)	✓
Wstęp do teorii mnogości	X

Drugi semestr informatyki pierwszego roku

nazwa przedmiotu	wymagany do zaliczenia
------------------	------------------------

- Klikamy na *Przedmioty wymagane*, Wybieramy jednostkę uniwersytecką, a później program studiów
- Programem studiów jest np. *Informatyka, dzienne magisterskie*

Moduł *Programy* przeznaczony jest dla studentów i służy im do przeglądania informacji o wymaganiach programów studiów. W tym module można zobaczyć wymagania zaliczeniowe (punktowe i przedmiotowe) dla każdego etapu ustalonego programu. Informacje te dostępne są bez konieczności logowania. Każdy student odbywa studia na pewnym kierunku (np. biologia lub informatyka) zgodnie z pewnym programem studiów. W przypadku ww. kierunków programami są *Biologia, dzienne magisterskie*, oraz odpowiednio *Zawodowe Studia Informatyki, wieczorowe licencjackie*. Może być kilka programów studiów na jednym kierunku np. na Informatyce jest też program *Informatyka, dzienne magisterskie*. Program może być też realizowany na kilku kierunkach. Student może realizować kilka programów studiów, ale wówczas zaliczenia zdobyte w ramach jednego programu, są ignorowane podczas rozliczania pozostałych programów. Z każdym programem wiążą się jego etapy. Student realizuje etap programu w trakcie określonego cyklu dydaktycznego (rok, semestr, trymestr). Wymagania zaliczeniowe wiążą się z etapami studiów w ramach danego programu studiów. Moduł *Programy* pozwala na sprawdzenie, czy może być zaliczony, lub zaliczony warunkowo dany etap studiów wg danego programu studiów. Zielonym znakiem V oznaczone są przedmioty wymagane do zaliczenia warunkowego. Niezaliczenie tak oznaczonego przedmiotu uniemożliwia dalsze studia.

Moduł Programy - wymagane punkty

The screenshot shows the 'Punkty wymagane' section of the USOS WED system. It details the required points for the Informatics program across three semesters. The data is as follows:

semestr	tytuł przedmiotu	zalicza	zalicza warunkowo
Pierwszy semestr informatyki pierwszego roku	Europejski System Transferu i Akumulacji Punktów	30	10
	UH wychowanie fizyczne	2	0
Drugi semestr informatyki pierwszego roku	Europejski System Transferu i Akumulacji Punktów	60	45
	UH wychowanie fizyczne	2	0
Drugi rok informatyki	Europejski System Transferu i Akumulacji Punktów	120	105
	MMiJN fakultatywne informatyczne	1	0
	MMiJN zaliczenie za egzamin II roku informatyki	7	5
	MMiJN zaliczenie za egzamin II roku informatyki	7	5

- Klikamy na *Punkty wymagane*, wybieramy jednostkę uniwersytecką, a później program studiów

Na poprzednim slajdzie były ukazane przedmioty wymagane do zaliczenia warunkowego. Takie zaliczenie jest możliwe pod warunkiem spełnienia wymagań punktowych.

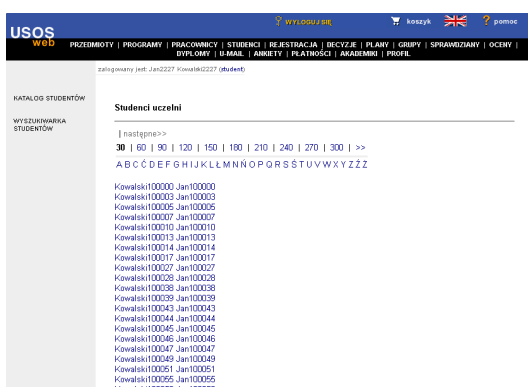
Moduł Pracownicy

The screenshot shows the 'Pracownicy uczelni' page in the USOS WED system. It displays a list of staff members at the Faculty of Mathematics, Informatics and Mechanics of the University of Warsaw. The list includes names and identification numbers, such as Kowalska103580 Ewa103580, Kowalska1940 Ewa1940, and so on, up to Kowalska483 Ewa483.

- Umożliwia przeglądanie listy pracowników uczelni
- Można korzystać z wyszukiwarki
- Klikamy *Katalog pracowników*, wybieramy jednostkę organizacyjną a później program studiów

Moduł *Pracownicy* zawiera aktualną listę pracowników dydaktycznych uczelni. Część danych o pracownikach dostępna jest bez konieczności logowania, lecz np. obejrzenie planu zajęć danego pracownika wymaga zalogowania.

Moduł Studenci



- Umożliwia przeglądanie listy studentów uczelni
- Klikamy *Wyszukiwarka*
- Wpisujemy szukane nazwisko

Przeglądanie listy studentów wymaga zalogowania. Można przeglądać studentów według jednostki organizacyjnej oraz kierunków i programów studiów.

Moduł Rejestracja

- Rejestracja na przedmioty polega na wskazaniu przez studenta przedmiotów, na które chce uczęszczać
- System USOSweb nie rejestruje studenta automatycznie na przedmioty obowiązkowe!
- Znajomość wymagań programu i właściwy dobór przedmiotów jest obowiązkiem studenta
- Zapisywanie się na przedmioty jest możliwe tylko w czasie trwania rejestracji
- W czasie trwania rejestracji można wielokrotnie zmieniać swoje deklaracje dotyczące przedmiotów
- W momencie zakończenia rejestracji, system będzie rozpatrywał tylko stan końcowy deklaracji

Moduł Rejestracja - koszyk



- Klikamy *Rejestracja*
- Okno, które widzimy ukazuje zawartość koszyka

Zawartość koszyka to lista przedmiotów, na które student chce się zarejestrować (złożył prośbę o rejestrację, prośba czeka na rozpatrzenie lub została rozpatrzona). W każdym wierszu listy znajdują się:

- Status rejestracji (A — zaakceptowana, O — odrzucona permanentnie, Z — odrzucona z możliwością ponowienia prośby, P — prośba o rejestrację)
- Kod i nazwa przedmiotu
- Cykl dydaktyczny
- Ikona z rysunkiem koszyka umożliwiająca rezygnację z rejestracji
- Liczba chętnych na przedmiot, tzn. osób których status rejestracji ma wartość A lub P oraz limit miejsc
- Odnośnik *Edycja* pozwalający na przejście do okna umożliwiającego podpinanie przedmiotu do programu studiów
- Pole umożliwiające zaznaczenie kilku przedmiotów w celu zbiorowego podpięcia do programu studiów Po zaznaczeniu zestawu przedmiotów można nacisnąć przycisk *Podepnij* i przejść do okna podpinania

Moduł Rejestracja - wybór przedmiotu

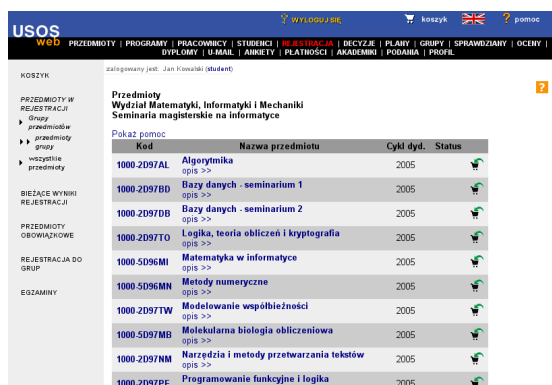


- Klikamy na *Przedmioty w rejestracji*
- Okno, które widzimy ukazuje listę grup przedmiotów
- Można też wyświetlić listę wszystkich przedmiotów, dla których w danym okresie trwa rejestracja

Pokazywane są jedynie grupy przedmiotów, w których znajduje się przynajmniej jeden przedmiot, na który student może się zarejestrować. W przypadku Jana Kowalskiego są to jedynie seminaria magisterskie (akurat w danym okresie trwa rejestracja właśnie na seminaria magisterskie).

Poniżej zostanie pokazany proces rejestracji, w tym przykładzie Jan Kowalski zrezygnował z przedmiotu na który był zarejestrowany (patrz poprzedni slajd) i teraz musi zarejestrować się ponownie.

Moduł Rejestracja - rejestracja na wybrany przedmiot



- Okno, które widzimy ukazuje listę przedmiotów w grupie, którą wybraliśmy
- Aby złożyć prośbę o zarejestrowanie, należy kliknąć koszyk przy wybranym przedmiocie

Dla każdego przedmiotu dostępny jest szczegółowy opis. Aby go zobaczyć, należy kliknąć odnośnik *Opis*.

Kliknięcie na koszyk, nie oznacza zarejestrowania na przedmiot. Oznacza to jedynie złożenie prośby o zarejestrowanie, na której rozpatrzenie trzeba poczekać.

Moduł Rejestracja — podpięcie przedmiotu do etapu lub programu studiów

zalogowany jest: Jan Kowalski (student)

Podpinanie do programu lub etapu

Twoja prośba o rejestrację na przedmiot została przyjęta. Teraz możesz wybrać program albo etap programu, na poczet którego chcesz zaliczać ten przedmiot. Wybór może dotyczyć jedynie tych programów, które nie zostały jeszcze zgłoszone do rozliczenia i dla których nie upłynęła data rozliczenia.

Jeżeli naciśniesz przycisk ZATWIERDŹ przy pustym formularzu, to przedmiot nie zostanie zaliczony na poczet żadnego programu. Jeżeli wcześniej podpiąłeś przedmiot pod inny program, to teraz możesz jedynie złożyć prośbę o podpięcie, która będzie rozpatrzona przez Dziekana.

- Metody numeryczne (1000-SD96MN), Rok akademicki 2005/06 (2005)

	Podpięcie	Prośba	Zgłoszony do rozliczenia	Data rozliczenia
Ekonomia, dzienne magisterskie	<input type="checkbox"/>	<input type="checkbox"/>	nie	2006-09-30
Specjalność informatyka i ekonometria trzeci rok	<input type="checkbox"/>	<input type="checkbox"/>	nie	2006-09-30
Specjalność ekonomia integracji europejskiej czwarty rok	<input type="checkbox"/>	<input type="checkbox"/>	nie	2006-09-30
Informatyka, dzienne magisterskie	<input type="checkbox"/>	<input type="checkbox"/>	nie	2006-09-30
Pięty rok informatyki	<input type="checkbox"/>	<input type="checkbox"/>	nie	2006-09-30

Zatwierdź | Odepnij wszystkie

- Po kliknięciu na koszyk musimy podpiąć wybrany przedmiot do etapu lub programu studiów
- Po zaznaczeniu etapu studiów, należy kliknąć *Zatwierdź*

Jeżeli podepnimy przedmiot pod jakiś etap lub program, możemy też złożyć prośbę o podpięcie pod inny etap lub program. Prośba ta zostanie rozpatrzona. Należy w takim przypadku zaznaczyć pole *Prośba*.

Uwaga! Jeśli naciśniemy przycisk *Zatwierdź* bez wyboru etapu lub programu studiów, przedmiot nie zostanie zaliczony na poczet żadnego programu lub etapu.

Moduł Rejestracja — wygląd koszyka po złożeniu prośby o rejestrację

zalogowany jest: Jan Kowalski (student)

Koszyk

Pokaż pomoc

Pokaż wszystkie rejestracje

Status	Kod	Nazwa przedmiotu	Cykl dyd.	Chętni/Limit	Kolejność wyboru	Podpięcie Zbiorowe
prośba o zarej.	(P) 1000-SD96MN	Metody numeryczne	2005	20/8	1	EDYCJA

Podepnij

- Rejestracja Jana Kowalskiego ma status P

W przypadku innych przedmiotów na które student chce się zarejestrować, postępowanie jest analogiczne jak wyżej opisane. Należy jednak wiedzieć, iż system USOSweb nie sprawdza, czy student ma prawo wybrać dany przedmiot (np. czy wybiera przedmiot z grupy przedmiotów do wyboru dla swojego kierunku studiów). Należy także pamiętać, aby zapisać się na wszystkie przedmioty obowiązkowe na danym etapie studiów.

Moduł Rejestracja — przeglądanie przedmiotów obowiązkowych

The screenshot shows the USOSweb interface for a student named Jan Kowalski. The main content area is titled 'Twoje programy' and contains a table of programs. The table has two columns: 'Kod' and 'Opis programu'. The programs listed are:

Kod	Opis programu
DM-JSEM	Jednoczesne Studia Ekonomiczno-Matematyczne, dzienne magisterskie
DM-INF	Informatyka, dzienne magisterskie
DM-EK	Ekonomia, dzienne magisterskie

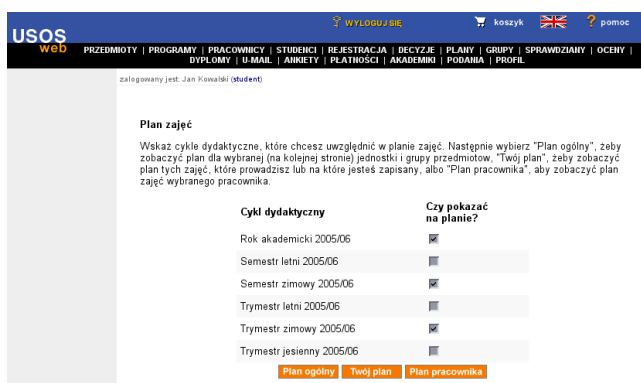
- Odnośnik *Przedmioty obowiązkowe* pozwala wyświetlić przedmioty obowiązkowe
- Ułatwia to wybór przedmiotów obowiązkowych do rejestracji
- Po kliknięciu na *Przedmioty obowiązkowe* należy wybrać program studiów

Moduł Decyzje

Moduł *Decyzje* pozwala na przechowywanie w systemie w postaci elektronicznej informacji o osiągnięciach przedmiotowych studenta, uzyskanych przez niego na innych uczelniach w ramach indywidualnych wyjazdów, przemiesień, programów MOST oraz Socrates/Erasmus. Każda decyzja definiowana jest dla danego studenta w dziekanacie i jest związana z pewnym etapem programu studiów studenta oraz cyklem dydaktycznym.

Ponieważ moduł ten nie jest często używany, nie zostaną w tym miejscu przedstawione szczegóły związane z jego obsługą i funkcjonowaniem.

Moduł Plany

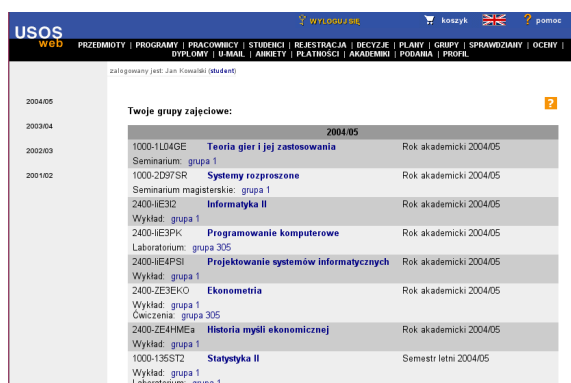


- Aby obejrzeć własny plan, należy kliknąć *Twój plan*
- Można też oglądać plany grup przedmiotów i nauczycieli akademickich

Moduł *Plany* udostępnia studentom i nauczycielom akademickim plany zajęć utworzone na podstawie informacji o zajęciach przechowywanych w bazie danych systemu USOS. W planie zajęć studenta widać zajęcia na które jest zapisany.

Należy pamiętać, że przedmioty są oferowane w cyklach dydaktycznych, np. może być tak, że seminaria, proseminaria są związane z rokiem akademickim, monografie z semestrem, a zajęcia na studiach wieczorowych — z trymestrem. Aby więc na planie uzyskać zajęcia z określonego okresu, należy wskazać wszystkie cykle, które zachodzą na dany okres. Przykładowo, miesiące październik-grudzień zachodzą na cykle: rok akademicki, semestr zimowy i trymestr jesienny, zaś miesiące kwiecień-czerwiec na: rok akademicki, semestr letni i trymestr letni.

Moduł Grupy



- Aby obejrzeć własne grupy ćwiczeniowe, należy wybrać z menu rok akademicki

Moduł *Grupy* służy studentom do przeglądania informacji o grupach zajęciowych. Student może sprawdzić do jakiej grupy został przypisany, kto prowadzi zajęcia w tych grupach i w jakich salach będą się one odbywały. Można też zobaczyć listę wszystkich studentów przypisanych do swojej grupy. Oprócz informacji o grupach, do których student jest aktualnie przypisany, studenci mogą przeglądać informacje o swoich dawnych grupach.

Moduł Sprawdziany

The screenshot shows the USOS interface for a student named Jan Kowalski. It displays the results for 'Kołokwium 1 - wyniki' (Exam 1 - results) for the subject 'Programowanie w logice' (Logic Programming). The exam was held on 2004-11-30. A table shows scores for five tasks: Zadanie 1 (3), Zadanie 2 (5), Zadanie 3 (1), Zadanie 4 (3.5), and Zadanie 5 (13.5), with a total score of 25.5. Below the table, there is a section for 'Zadania sprawdzianu' (Exam tasks) with a table showing the number of questions and the number of correct answers for each task.

Zadanie 1	Zadanie 2	Zadanie 3	Zadanie 4	Zadanie 5	Suma
1	3	5	1	3.5	13.5

nazwa	punkty	statystyki wyników	
Zadanie 1	0-3	wszystkich	tylko niepustych
Zadanie 2	0-4	wszystkich	tylko niepustych
Zadanie 3	0-5	wszystkich	tylko niepustych
Zadanie 4	0-3	wszystkich	tylko niepustych
Zadanie 5	0-5	wszystkich	tylko niepustych

- Aby obejrzeć swoje wyniki ze sprawdzianów należy wybrać przedmiot z listy
- Następnie należy wybrać sprawdzian, w tym przypadku jest to *kołokwium 1*

Moduł *Sprawdziany* daje studentowi dostęp do wyników sprawdzianów oraz ocen uzyskanych w trakcie trwania zajęć. Dostęp możliwy jest tylko wtedy, gdy osoba prowadząca przedmiot zdecydowała się skorzystać z tego modułu.

Moduł Oceny

The screenshot shows the 'Oceny końcowe' (Final grades) section in the USOS system. It displays a list of subjects and their corresponding grades for the academic year 2004/05 and the first semester of 2004/05. The grades are listed in a table format.

Rok akademicki 2004/05			
2400-ZE3EKO	Ekonometria		4 (CW)
2400-ZE4HMEa	Historia myśli ekonomicznej		5
2400-IE3E2	Informatyka II		5
2400-ZE4PRHa	Prawo handlowe		3
2400-IE3PK	Programowanie komputerowe		5
2400-IE4PSI	Projektowanie systemów informatycznych		5 (MYK)
1000-2D97SR	Systemy rozproszone		ZAL
1000-1LD4GE	Teoria gier i jej zastosowania		4,5
Semestr Ietni 2004/05			
2400-IE4EK2	Ekonometria II		5
1000-2PD4LB	JZEE - budowa aplikacji		5
1000-234MRJ	Metody realizacji języków programowania		4
1000-24C8RJ	Metody realizacji języków programowania - laboratorium		5
1000-2M0DWW	Programowanie WWW		5
2400-ZE3RA2	Rachunkowość zarządcza		5
2400-7E3MRTE	Statystyka ekonomiczna		4

- Można oglądać wszystkie swoje oceny

Za pomocą opcji *Oceny końcowe* student może oglądać uzyskane przez siebie oceny, bez podziału na programy studiów, w ramach których będą rozliczane. Opcja *Programy studiów* pozwala oglądać programy studiów i stan rozliczenia etapów w ramach programów.

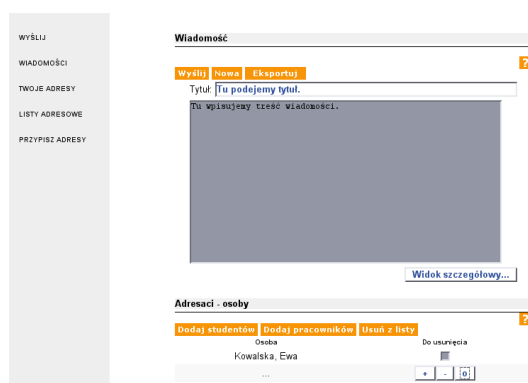
Moduł Dyplomy



- Można przeglądać własne złożone prace lub zatwierdzone tematy
- Lub przeglądać/wyszukiwać prace innych, we wszystkich jednostkach UW

Moduł *Dyplomy* udostępnia informacje o złożonych pracach dyplomowych oraz o zatwierdzonych tematach prac dyplomowych. Istnieje także możliwość wpisania do systemu streszczenia oraz słów kluczowych.

Moduł Umail



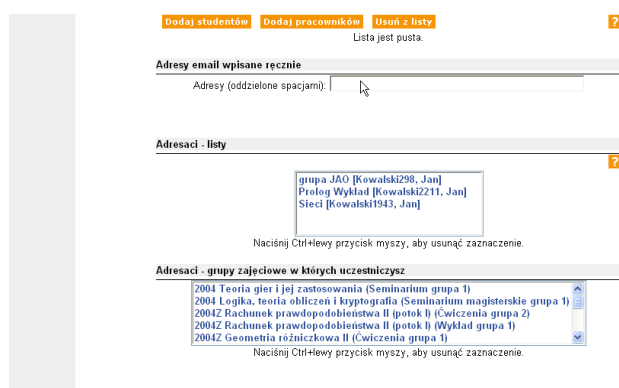
- Aby utworzyć wiadomość należy kliknąć *Wyślij* (menu po lewej stronie)
- Do adresatów (za pomocą *Dodaj Studentów*) została dodana Ewa Kowalska
- Aby wysłać wiadomość należy kliknąć przycisk *Wyślij*

Moduł *Umail* służy do przekazywania wiadomości tekstowych między użytkownikami systemu USOSweb, umożliwia też wysyłanie e-maili do osób, które nie korzystają z USOSweba. Użytkownik może wysyłać wiadomości do wybranych osób lub list adresowych, może też tworzyć własne listy adresowe i nimi zarządzać.

Moduł *Umail* pozwala na łatwe znajdowanie adresatów za pomocą wyszukiwarki studentów oraz pracowników. Należy kliknąć przycisk odpowiednio *Dodaj studentów* lub *Dodaj pracowników*. Można też wpisywać (w polu *Adresy email wpisane ręcznie*) dowolne adresy e-mail.

Można również definiować własne listy adresowe — należy kliknąć *Listy adresowe*. Z tak zdefiniowanych list można później skorzystać podczas wysyłania wiadomości. Należy zaznaczyć odpowiednią listę w polu *Adresaci* — *listy*.

Moduł Umail — standardowe listy adresowe



- Aby wysłać wiadomość do wszystkich w danej grupie zajęciowej wystarczy wybrać jedną lub kilka grup z listy *Adresaci* — *grupy zajęciowe w których uczestniczysz*

W *Umailu* listy wysyłkowe związane z grupami zajęciowymi do których student jest zapisany są dostępne standardowo i nie trzeba ich tworzyć ręcznie. Aby wysłać wiadomość do wszystkich uczestników zajęć w danej grupie, należy jedynie wybrać tę grupę z pola *Adresaci* — *grupy zajęciowe*. Jest to bardzo wygodne.

Moduł Ankiety

Ankieta	Cykl	Przedmiot	Zajęcia
1000-2004/W1	2004/TJ	1000-611WSO	Współczesne systemy operacyjne - Wykład
1000-2004/W1	2004/TJ	1000-612AS2	Algorytmy sekwencyjne II - Wykład
1000-2004/W1	2004/TJ	1000-612OPD	Organizacja i zarządzanie produkcją oprogramowania - Wykład I
1000-2004/W1	2004/TJ	1000-612PSI	Projektowanie systemów informacyjnych - Wykład
1000-2004/W1	2004/TJ	1000-612SJP	Semantyka języków programowania - Wykład
1000-2004/W1	2004/TJ	1000-612SK1	Sieci komputerowe I - Wykład
1000-2004/W1	2004/TJ	1000-612WOP	Wytwarzanie oprogramowania - Wykład
1000-2004/W1	2004/TJ	1000-622SK1	Sieci komputerowe I - laboratorium - Laboratorium
1000-2004/W1	2004/TJ	1000-6M04PM	Pamięci masowe - urządzenia, sieci, archiwizacja i zarządzanie

- Aby zacząć wypełnianie ankiety, należy kliknąć *Wypełnij Ankietę*
- Następnie należy wybrać zajęcia
- Później wybieramy nazwisko prowadzącego

Ankieta służy do oceny zajęć i prowadzących je nauczycieli akademickich. Ankiety dotyczące zajęć mogą wypełniać uczestniczący w tych zajęciach studenci. Ankieta jest anonimowa, żeby ją wypełnić trzeba się co prawda zalogować, ale system nie zapamiętuje związku między konkretną ankietą, a wypełniającym ją studentem.

Jeśli student wyrazi swoje opinie na temat zajęć oraz wystawi ocenę prowadzącemu je nauczycielowi akademickiemu, pomoże mu dostrzec pewne kwestie być może dla niego niezauważalne. Istnieje więc szansa, że w przyszłej edycji zajęć będą one ciekawsze i lepiej prowadzone. Warto więc poświęcić chwilę czasu na wypełnianie ankiet.

Można wypełniać jedynie ankietę *Aktywną*. Ankieta jest *Aktywna* w okresie wyznaczonym przez dziekana.

Wypełnienie ankiety polega na wpisaniu dla każdego pytania liczby z danego zakresu, dopisanie komentarza tekstowego do zajęć oraz zatwierdzeniu wpisów. Każdą ankietę można wypełnić tylko raz — po zatwierdzeniu staje się niedostępna. Nie należy zatem zatwierdzać częściowych wyników. Ankieta przestaje być również dostępna, gdy minie jej termin ważności. Nie należy zatem zbyt długo zwlekać z jej wypełnianiem, bo nawet niewypełniona ankieta staje się niedostępna po upływie określonego czasu.

Po zakończeniu terminu wypełniania ankiet trafia ona do dziekana, który przegląda i „zatwierdza” słowne komentarze. Dziekan rezerwuje sobie prawo do usunięcia zbyt osobistych wypowiedzi. Może również dopisać do tych komentarzy podsumowanie, które także będzie widoczne w USOSweb.

Wreszcie wyniki ankiet trafiają do USOSweb i stają się dostępne dla pracowników wydziału, przy czym każdy widzi jedynie własne. Ankieta jest wówczas w stanie *Zakończona* i tylko taka ankieta jest widoczna dla pracowników.

Moduł Ankiety — wypełnianie ankiety

1000-611WSO Współczesne systemy operacyjne

Ankieta: Ankieta za pierwszy trymestr dla studentów studiów wieczorowych
Jednostka: Wydział Matematyki, Informatyki i Mechaniki
Cykl: Trymestr jesienny 2004/05
Typ zajęć: Wykład

Prowadzący zajęcia: Jan Kowalski

Treść

	Ocena
1. Czy prowadzący jest przygotowany do zajęć?	5
2. Czy zajęcia rozpoczynają się punktualnie, czy prowadzący jest dostępny w godzinach konsultacji?	7
3. Czy wyjaśnienia prowadzącego są zrozumiałe?	5
4. Dostępność notatek/możliwość prowadzenia notatek?	9
5. Czy zajęcia są interesujące?	6

Komentarz

- Dla każdego punktu listy wybieramy ocenę
- Można też napisać komentarz

Moduł Płatności

USOS

WYLOGUJ SIĘ

koszyk

pomoc

WIOD PRZEDMIOTY | PROGRAMY | PRACOWNICY | STUDENCI | REJESTRACJA | RECYZJE | PLANY | GRUPY | SPRAWOZDANIA | OCENY | DIPLOMY | U.MAIL | ANKIETY

PLATNOŚCI | AKADEMIA

zalogowany jako: Jan Kowalski(1212) (Wyloguj)

Należności do zapłacenia

Data	Opis	Kwota
2005-02-10	OPL NR 18447 ZA STUDIOWANIE	2340.00 (+odsetki 89.53)

Oblicz odsetki dla daty: 2005.06.04 [pokaż](#)

• [wydrukuj blankiety przelewów](#)

Dane do przelewu na Twoje indywidualne konto:

Nazwa odbiorcy:	Jakub Uczelnia
	00-000 Warszawa, Uczelniana 1
Nazwa banku:	Bank Uczelniany S.A., 1 oddział w Warszawie
Numer rachunku:	BL 1030 1944 0000 8000 6000 0000 2120

- Po uruchomieniu modułu, widzimy nasze należności do zapłacenia
- Można wydrukować blankiet (jest wymagany do tego Acrobat Reader)
- Można też przeglądać wcześniejsze wpłaty (klikając *Należności zapłacone*)

Moduł *Płatności* pozwala studentom na przeglądanie informacji o stanie ich zobowiązań finansowych wobec uczelni.

Studenci mogą sprawdzić tytuł, termin zapłaty i wysokość należności oraz indywidualny numer konta bankowego, na który należy dokonać wpłaty. Można też wyliczać odsetki za zwłokę na dowolny dzień oraz przeglądać historię wpłat. Dodatkowo istnieje możliwość uzyskania wypełnionego blankietu wpłat w formacie PDF.

Każdy student ma indywidualny numer konta bankowego, na które wpłaca kwoty związane z zobowiązaniami finansowymi wobec Uniwersytetu Warszawskiego. Numer ten ma ściśle określony format jednoznacznie identyfikujący posiadacza.

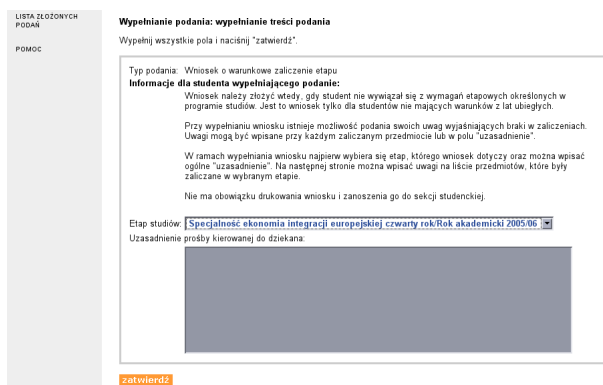
Przykładami zobowiązań finansowych wobec UW mogą być np. opłaty za czesne, powtarzanie roku, zaliczenia warunkowe itp.

Moduł Akademiki



- Można przeglądać informacje o akademikach, rozdzielaniach miejsc
- Można też wypełnić i wydrukować podanie o prawo do zamieszkania w Domu Studenckim — należy kliknąć *Wypełnij formularz*

Moduł Podania



- Aby złożyć podanie, należy wybrać program studiów
- Następnie przy interesującym nas podaniu należy kliknąć odnośnik *Wypełnij i złóż podanie*
- Można pisać treść podania

Moduł *Podania* umożliwia studentom składanie podań do dziekana.

Pomoc systemu USOSweb



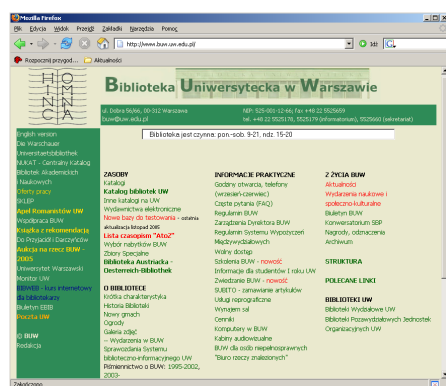
- Aby wyświetlić pomoc, należy kliknąć *Pomoc* w prawym górnym rogu
- Następnie warto kliknąć *E-Nauka* oraz *E-Podręczniki* i uruchomić podręcznik klikając na któryś z odnośników np. *Rejestracja*

E-Nauka to część USOSweb zawierająca materiały multimedialne ilustrujące działanie różnych modułów systemu. *E-Podręczniki* prezentują funkcjonalność modułów, a *E-Ćwiczenia* umożliwiają nabycie większej biegłości w ich używaniu. Ta forma pomocy systemowi USOSweb jest warta polecenia szczególnie początkującym użytkownikom systemu. Z informacji zawartych w ww. materiałach korzystano przy pisaniu niniejszego rozdziału.¹

Oprócz *E-Podręczników* USOSweb zawiera pomoc kontekstową. Aby uzyskać dostęp do pomocy kontekstowej należy kliknąć na pomarańczowy ? pojawiający się w różnych miejscach w systemie. Pomoc kontekstowa dotyczy elementów obsługi modułu z którego korzystamy w danej chwili.

16.3 Zasoby BUW dostępne drogą elektroniczną

16.3.1 Strona główna BUW



¹ Dzięki uprzejmości Autorki.

- Zasoby BUW dostępne drogą elektroniczną znajdują się na stronie BUW
- Strona główna BUW dostępna jest pod adresem <http://www.buw.uw.edu.pl/>
- Serwis internetowy BUW-u jest dość rozbudowany

16.3.2 Katalog komputerowy

Najbardziej podstawowym zasobem BUW dostępnym przez przeglądarkę internetową jest katalog on-line.

- Aby uzyskać dostęp do katalogu, należy kliknąć odnośnik *Katalog bibliotek UW*
- Można teraz wyszukiwać w katalogu

Katalog pozwala na przeszukiwanie zbiorów według wielu parametrów: autora, tytułu, haseł przedmiotowych, sygnatury/symbolu klasyfikacji oraz słów kluczowych. Według informacji podawanych przez BUW, zawiera on opisy wszystkich książek znajdujących się w wolnym dostępie oraz części książek zlokalizowanych w magazynie, księgozbiorach podręcznych pracowni i gabinetów zbiorów specjalnych BUW. W katalogu znajdują się również opisy aktualnie ukazujących się czasopism. BUW informuje, iż katalog elektroniczny jest sukcesywnie uzupełniany, również opisami wcześniejszych publikacji, w tym starszych czasopism i umożliwia też dostęp do opisów druków najwcześniejszych — z XV wieku (całość zasobu) oraz częściowo z XVI wieku (hasła na litery A-B). Wprowadzane są także opisy wydawnictw znajdujących się w niektórych bibliotekach wydziałowych Uniwersytetu Warszawskiego.

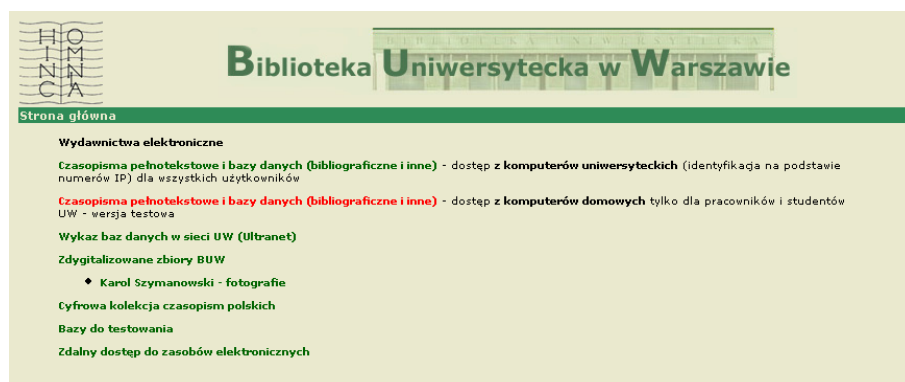
Zamawianie książek



- Po wyszukaniu w katalogu system umożliwia zamówienie książki
- Zamówienie jest możliwe, jeśli posiadamy aktualną kartę BUW
- Można też rezerwować książki jeśli egzemplarz jest aktualnie wypożyczony

16.3.3 Wydawnictwa elektroniczne

Dostęp do wydawnictw elektronicznych



- Aby uzyskać dostęp do wydawnictw elektronicznych, należy kliknąć odnośnik *Wydawnictwa elektroniczne*
- Największą częścią wydawnictw elektronicznych są czasopisma pełnotekstowe i bibliograficzne bazy danych
- Należy kliknąć odnośnik *Czasopisma pełnotekstowe i bazy danych (bibliograficzne i inne)*

Zbiór dostępnych elektronicznie czasopism i ich zakres tematyczny jest dość imponujący. Czasopisma są dostępne najczęściej w formacie PDF.

Dostęp do czasopism elektronicznych jest dozwolony jedynie z komputerów pracujących w sieci Uniwersytetu Warszawskiego lub po zalogowaniu. Aby uzyskać dostęp np. z domu należy użyć odnośnika *Czasopisma pełnotekstowe i bazy danych (bibliograficzne i inne) - dostęp z komputerów domowych tylko dla pracowników i studentów UW - wersja testowa* Można też korzystać poprzedniego odnośnika, ale wtedy trzeba użyć serwera proxy (jeśli jest dostępny na wydziale). Konfiguracja serwera pośredniczącego została opisana w rozdziale dotyczącym przeglądarek internetowych.

Rozdział 17

Egzamin

17.1 Zasady przeprowadzania egzaminu

1. Egzamin jest przeprowadzany w laboratorium komputerowym, pod bezpośrednim nadzorem egzaminatora. Każdy student pracuje na **własnym koncie** przy osobnym komputerze, wyposażonym w przynajmniej dwa sprawne systemy operacyjne: Linux i Windows. Pod każdym z systemów dostępne jest sprawnie działające oprogramowanie omawiane w niniejszym podręczniku (**DO ZROBIENIA** → (**lista**)). Student może wykonywać kolejne zadania egzaminacyjne w wybranym przez siebie systemie operacyjnym (niekoniecznie tym samym dla wszystkich zadań).

Ze względu na nieuniknione drobne (ale uciążliwe) różnice pomiędzy wersjami oprogramowania dostępnymi w labie, a np. zainstalowanymi na komputerze domowym studenta, wskazane jest, by na jakiś czas przed egzaminem każdy student miał możliwość dokładnego zapoznania się, przez co najmniej godzinę, z konfiguracją środowiska, którego będzie używać w trakcie egzaminu, używając w tym celu specjalnego gościnnego konta. Dodatkowo, w wyznaczonym czasie, administrator pracowni będzie odpowiadać na pytania dotyczące szczegółów użytkowania pracowni, które nie są opisane w standardowych dokumentach dostępnych w pracowni.

2. Komputery, na których będzie przeprowadzany egzamin, muszą być podłączone do Internetu.
3. Egzamin trwa 2 godziny i ma formę *praktycznego* sprawdzianu umiejętności poprzez wykonanie zestawu zadań. .
4. Studenci mogą korzystać z dowolnych pomocy naukowych w postaci notatek, książek (także niniejszego podręcznika) oraz materiałów o trwałej formie, zamieszczonych w internecie przed rozpoczęciem egzaminu. Jakikolwiek inne pomoce są zabronione.
5. W pracowni komputerowej powinny być łatwo dostępne (np. na zadanej stronie WWW) standardowe informacje o sposobie jej wykorzy-

stania. Podczas egzaminu nie będzie możliwości wydrukowania jakiegokolwiek dokumentu, ani zadawania egzaminatorowi lub opiekunowi pracowni pytań odnośnie sposobu wykorzystania sprzętu i oprogramowania.

6. Zdający egzamin jest odpowiedzialny za wykonywanie kopii zapasowych tworzonych plików w sposób umożliwiający kontynuację pracy po ewentualnej chwilowej awarii lub niestabilności systemu. Decyzję o sposobie postępowania w sytuacji poważniejszej awarii sprzętu podejmuje prowadzący egzamin w zależności od konkretnych okoliczności.
7. Ocenie podlegać będą *wyłącznie* pliki o nazwach takich, jakich zażądano w treści zadań.
8. Na ocenę składać się będą nie tylko ostateczne wyniki, ale także sposób ich uzyskania, przykładowo, jeśli ktoś zupełnie prawidłowo, ale *ręcznie* sporządzi spis treści, nie dostanie nawet połowy punktów co osoba, która ten sam spis wygenerowała automatycznie stosując dostępne narzędzia.

17.2 Cele egzaminu: lista umiejętności

Praca z dużym tekstem. Search and replace. Współpraca pomiędzy różnymi aplikacjami. Korzystanie z USOSweba w stopniu podstawowym (znalezienie wykładowcy, wysłanie do niego u-maila, odszukanie wykładu, ...). Posługiwanie się BUW-em i wyszukiwarkami. Kopiowanie (nie-plagiatowanie) dokumentów z internetu. Skomponowanie raportu od zera do PDF. Wprowadzenie poprawek do tekstu w formacie ODT, HTML lub RTF. Konwersja między nimi. Konwersja między kodowaniami tekstów po polsku. Archiwizacja dużych zbiorów plików. Tworzenie katalogów. Edycja grafik na potrzeby publikacji. Automatyczne spisy, itp. Automatyzacja formatowania przez style. Zamieszczenie tabeli opracowanej w Excelu w dokumencie tekstowym. Funkcje Excela, import, eksport, poprawianie danych, efektywne wykresy i włączanie ich do dokumentu. Automatyczna numeracja i odsyłacze w dokumencie tekstowym: rozdziały, strony, bibliografia, ilustracje, tabele. Podpisy. Wyszukiwanie: w pliku, w katalogu, w internecie.

17.3 Przykładowe zestawy egzaminów

17.3.1 Baza każdego egzaminu

W poniższym dokumencie znajdują się treści zadań.

1. Utwórz katalog o nazwie `Egzamin-login`, gdzie `login` jest taki, jak nazwa Twojego konta, np. użytkownik o nazwie `przykry` utworzy katalog `Egzamin-przykry`, a użytkownik o nazwie `pi160` utworzy katalog `Egzamin-pi160`.

Skonfiguruj wybrany przez siebie program pocztowy do obsługi konta o nazwie `login@students.mimuw.edu.pl`, gdzie `login` jest taki, jak nazwa Twojego konta, a hasło dostępowe jest takie samo jak hasło dla Twojego konta.

Na Twoim koncie znajduje się już list od egzaminatora, o temacie „*Egzamin PI*”, w którym znajdziesz ewentualne brakujące do wykonania zadań pliki.

Inne pliki potrzebne do wykonania zadań znajdziesz na stronie <http://www.mimuw.edu.pl/pi/~pi160-a08f56c/zadanie.html>.

Pliki, jakie są wymagane w każdym z zadań, zamieszczaj wprost w katalogu `Egzamin-login`. Po zakończeniu pracy, spakuj zawartość całego katalogu do jednego pliku i odeślij go na adres e-mail egzaminatora (tzn. nadawcy listu zawierającego zadania egzaminacyjne).

2. ...itd...

POWODZENIA! Pamiętaj, ocenie będą podlegać *wyłącznie* rozwiązania zawarte w mailu, który wyślesz do egzaminatora!

17.3.2 Przykładowy egzamin

W poniższym dokumencie znajdują się treści zadań.

Zadanie 17.3.1. Utwórz katalog o nazwie `Egzamin-login`, gdzie `login` jest taki, jak nazwa Twojego konta, np. użytkownik o nazwie `przykry` utworzy katalog `Egzamin-przykry`, a użytkownik o nazwie `pi160` utworzy katalog `Egzamin-pi160`.

Skonfiguruj wybrany przez siebie program pocztowy do obsługi konta o nazwie `login@students.mimuw.edu.pl`, gdzie `login` jest taki, jak nazwa Twojego konta, a hasło dostępowe jest takie samo jak hasło dla Twojego konta.

Na Twoim koncie znajduje się już list od egzaminatora, o temacie „*Egzamin PI*”, w którym znajdziesz ewentualne brakujące do wykonania zadań pliki.

Inne pliki potrzebne do wykonania zadań znajdziesz na stronie <http://www.mimuw.edu.pl/pi/~pi160-a08f56c/zadanie.html>.

Pliki, jakie są wymagane w każdym z zadań, zamieszczaj wprost w katalogu `Egzamin-login`. Po zakończeniu pracy, spakuj zawartość całego katalogu do jednego pliku i odeślij go na adres e-mail egzaminatora (tzn. nadawcy listu zawierającego zadania egzaminacyjne).

Rozwiązanie: Prawidłowa konfiguracja klienta poczty jest prosta, pod warunkiem, że przeprowadzimy ją z należytą ostrożnością (literówka w nazwie serwera lub w hasle może być źródłem niepotrzebnej irytacji i straty cennego czasu). Należy zastosować się do instrukcji korzystania z kont studenckich, która powinna znajdować się w LABie, albo być dostępna na stronie WWW. Najwygodniej użyć Thunderbirda albo Mozilli, bo schemat wykorzystania nie zależy od wybranego systemu operacyjnego.

Jeśli poczta nie działa, zazwyczaj lepiej zacząć całą procedurę od nowa, niż szukać błędów w ustawieniach.

Kopię listu od egzaminatora warto zawczasu gdzieś zachować na swoim koncie — na wszelki wypadek!

Jeśli jeszcze nie umiesz utworzyć samodzielnie katalogu o zadanej nazwie, to z pewnością nie powinieneś czytać niniejszego tekstu, ani tym bardziej przystępować do egzaminu z PI. Zatem tę kwestię pomijamy.

Pakowanie całego katalogu do jednego pliku nie powinno sprawić trudności. W przeglądarce plików (Linux) lub eksploratorze (Windows) klikamy prawym klawiszem myszki na ikonkę pliku i wybieramy **Utwórz archiwum** (Linux) lub analogiczne polecenie dla Windows.

Taki plik należy oczywiście wysłać jako *załącznik* do poczty. Odsyłanie listu na adres nadawcy musisz opanować sam, jeśli jeszcze nie wiesz, jak to się robi.

□ **Zadanie 17.3.2.** Jest dany plik `Egzamin1/katalogi.zip`. W nim — ogromna liczba podkatalogów i plików. Należy znaleźć plik o nazwie `zadanie.txt`, albo `zadania.txt`, a może `zadanie120.txt`, albo coś w tym rodzaju. Po odnalezieniu pliku, należy wkleić jego treść do pliku Writera o nazwie `znaleziony`. □

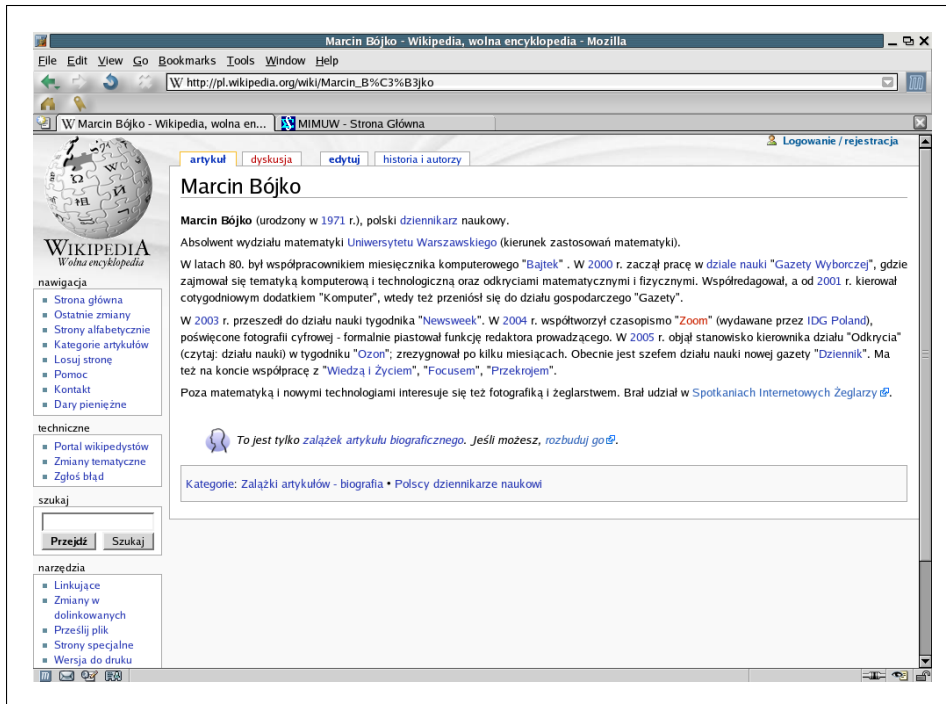
Rozwiązanie: Najprościej rozpakować całe archiwum do osobnego katalogu, a następnie przeprowadzić wyszukiwanie w podkatalogu `test`, do którego całe archiwum się rozpakowuje. Oczywiście, wszystko wskazuje na to, że szukamy pliku o nazwie w rodzaju „`zadan*.txt`” i tak właśnie specyfikujemy nazwę pliku w wyszukiwarce plików.

Lokalizujemy ten plik ostatecznie w katalogu `test/test12/test101/test1020` pod nazwą `zadanie007.txt`. Reszta jest banalna.

□ **Zadanie 17.3.3.** Dowiedz się, kim jest Marcin Bójko i gdzie do tej pory pracował. Zamieść krótką notatkę o nim w pliku tekstowym. Jeśli jest to możliwe, notatkę uatrakcyjnij zdjęciem. □

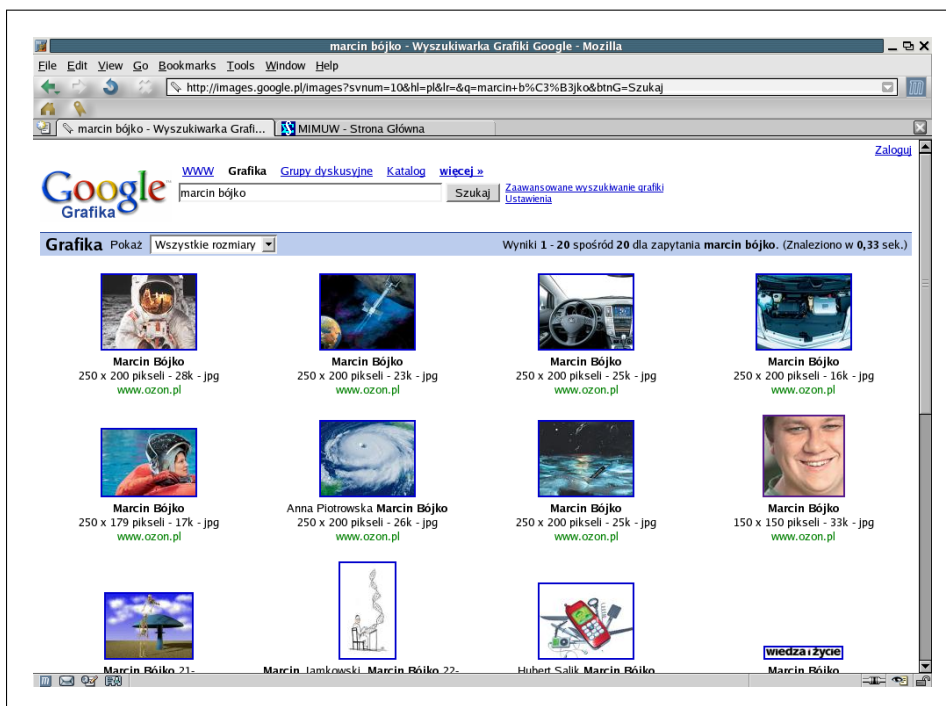
Rozwiązanie: Trzeba poszperać w sieci. Do wyszukiwarki Google wpisujemy Marcin Bójko

(nie: „Bojko”) i szybko trafiamy na stronę w Wikipedii:



Strona w Wikipedii o Marcinie Bójko

Aby znaleźć zdjęcie redaktora Bójko, spróbujemy skorzystać z opcji wyszukiwania grafiki w Google:



Szukamy plików graficznych: czy Bójko to ten na pierwszym, a może na dziesiątym zdjęciu? A może na żadnym z nich?

— i jak można przypuszczać, jedno ze zdjęć może być zdjęciem interesującej nas osoby. Jednak nie jest to z drugiej strony pewne — zdjęcie nie jest podpisane.

Wczytując się w tekst na stronach Wikipedii dowiadujemy się, że Bójko pracuje obecnie w gazecie „Dziennik”. Może więc tam więcej się o nim dowiemy? Rzeczywiście, są dalsze informacje, a nawet jednoznaczne zdjęcie delikwenta.



Dodatkowe informacje o M. Bójko

Teraz formułujemy tekst notatki, w którym

- Sygnalizujemy dokładnie źródła informacji
- Fragmenty cytowane otaczamy cudzysłowami
- Skopiowane z internetu zdjęcie podpisujemy z podaniem źródła pochodzenia

Rozwiązanie naruszające prawa autorskie będzie bardzo nisko punktowane!

Zadanie 17.3.4. Zbadaj, czy na UW jest już książka *Numerical mathematics*, której autorami są Alfio Quarteroni i inni. Podaj, na jakim wydziale i jaka jest jej sygnatura. Sprawdź także, jaka jest jej orientacyjna cena w Euro. Wynik swoich poszukiwań zamieść w notatce o nazwie `sygnatura.txt`.

Rozwiązanie: Korzystamy z katalogu BUW. Sprawdzamy ile jest takich książek i gdzie się znajdują. Jedna z takich książek jest na Wydziale MIM. Druga — w IFT. Efekt zapisujemy do zwykłego pliku tekstowego, korzystając ze standardowego edytora ASCII, np. w Linuxie możemy skorzystać z `nedit'a`.

Co do ceny, można to wygooglować, albo można zauważyć na „fiszce” w BUWie, że wydała ją wydawnictwo Springer. Lokalizujemy w Google to wydawnictwo,

wyszukujemy w nim tę książkę w jego wewnętrznej wyszukiwarce i odczytujemy cenę. Oczywiście w notatce napiszemy również, gdzie taką cenę znaleźliśmy. W przeciwnym razie informacja jest mało wartościowa.

Zadanie 17.3.5. Dane są trzy pliki tekstowe `ryby-win.txt`, zakodowany w standardzie Windows-1250, `ryby-iso.txt`, zakodowany w standardzie ISO-8859-2 oraz `ryby-utf.txt`, zakodowany w UTF-8. Sporządź jeden dokument w formacie PDF, `ryby.pdf`, w którym zamieścisz wszystkie trzy.

Rozwiązanie: Musimy skonwertować te trzy teksty do jednego kodowania, a potem scalić i przerobić na PDF. W Linuxie jest to bardzo łatwe dzięki poleceniu `iconv`:

```
iconv -f WINDOWS-1250 -t ISO-8859-2 ryby-win.txt > ryby.txt
cat ryby-iso.txt >> ryby.txt
iconv -f UTF-8 -t ISO-8859-2 ryby-utf.txt >> ryby.txt
```

Potem konwertujemy tekst do formatu PDF, najprościej z wykorzystaniem OpenOffice, choć oczywiście można też skorzystać z odpowiednio skonfigurowanego polecenia konwersji na Postscript, `a2ps`, po którym użyjemy konwersji z Postscriptu na PDF poleceniem `ps2pdf`.

W dowolnym systemie operacyjnym całe zadanie możemy wykonać także w OpenOffice, wczytując każdy z plików do OpenOffice; we Writerze, wybieramy `Plik` → `Otwórz` → `Tekst zakodowany`. W dialogu wybieramy odpowiednie kodowanie dla każdego z plików. W ten sposób mamy otwarte trzy okienka z trzema dokumentami.

Teraz tworzymy nowy dokument i po kolei kopiujemy i wklejamy do niego zawartość wczytanych poprzednio tekstów. Gotowy plik ewentualnie jednolicie formatujemy, a potem zapisujemy do formatu PDF.

Zadanie 17.3.6. Poniżej znajduje się zestaw bardzo wielu danych, który zawarty jest w pliku PDF.

```
PESEL Pyt-A Pyt-B Pyt-C Quiz
6706120402 TAK NIE TAK 18
6706120501 TAK TAK NIE 4
6706120924 TAK TAK TAK 16
6706120225 NIE TAK TAK 17
6706120660 TAK NIE NIE 1
6706120679 NIE NIE TAK 1
6706120828 TAK TAK TAK 8
6706120379 NIE NIE NIE 3
6706120556 TAK TAK TAK 15
6706120746 TAK NIE NIE 7
6706120501 TAK NIE TAK 19
6706120654 TAK TAK NIE 16
6706120934 NIE NIE NIE 19
6706120070 NIE NIE NIE 8
6706120784 NIE NIE TAK 11
6706120434 TAK TAK NIE 18
6706120534 TAK TAK NIE 17
6706120611 TAK TAK TAK 2
6706120327 TAK TAK NIE 19
6706120554 NIE NIE TAK 3
6706120922 TAK TAK TAK 13
6706120213 TAK NIE NIE 2
```

6706120997 NIE TAK TAK 18
6706120947 TAK TAK NIE 19
6706120909 TAK NIE NIE 18
6706120685 TAK TAK TAK 16
6706120822 TAK NIE NIE 13
6706120950 NIE NIE NIE 14
6706120230 TAK NIE TAK 0
6706120623 NIE TAK NIE 13
6706120559 NIE TAK NIE 6
6706120270 NIE NIE NIE 17
6706120724 TAK NIE NIE 17
6706120361 TAK NIE TAK 3
6706120537 NIE NIE NIE 15
6706120856 NIE TAK NIE 18
6706120486 TAK NIE TAK 5
6706120860 TAK TAK TAK 3
6706120170 NIE NIE TAK 7
6706120481 NIE NIE NIE 1
6706120430 TAK NIE TAK 1
6706120570 NIE TAK NIE 0
6706120434 NIE NIE TAK 10
6706120204 TAK TAK NIE 1
6706120631 NIE NIE TAK 16
6706120276 TAK TAK NIE 9
6706120869 TAK NIE NIE 9
6706120006 NIE TAK TAK 14
6706120363 NIE NIE TAK 8
6706120574 NIE NIE TAK 8
6706120417 NIE NIE NIE 7
6706120914 TAK NIE NIE 10
6706120270 NIE TAK TAK 8
6706120662 NIE NIE NIE 10
6706120238 NIE TAK TAK 3
6706120391 TAK NIE TAK 10
6706120856 NIE TAK TAK 9
6706120097 NIE TAK NIE 3
6706120140 NIE NIE TAK 8
6706120340 NIE TAK NIE 5
6706120586 TAK NIE NIE 16
6706120388 TAK TAK NIE 2
6706120772 NIE NIE TAK 5
6706120200 TAK NIE NIE 9
6706120783 TAK TAK NIE 9
6706120174 NIE NIE TAK 3
6706120099 TAK TAK TAK 6
6706120689 NIE NIE TAK 12
6706120158 NIE TAK TAK 18
6706120768 TAK NIE TAK 5
6706120007 TAK TAK NIE 17
6706120659 TAK NIE NIE 1
6706120549 TAK TAK TAK 7
6706120186 TAK TAK TAK 18
6706120959 TAK NIE NIE 12
6706120544 TAK TAK NIE 8
6706120771 TAK NIE TAK 14
6706120743 TAK TAK NIE 1
6706120275 NIE NIE NIE 18
6706120647 NIE TAK NIE 2
6706120595 NIE NIE NIE 10
6706120838 TAK NIE TAK 5

6706120593 NIE TAK TAK 13
 6706120744 TAK NIE TAK 18
 6706120786 NIE TAK NIE 16
 6706120333 NIE TAK TAK 16
 6706120291 TAK NIE TAK 8
 6706120203 NIE NIE TAK 5
 6706120942 TAK TAK TAK 0
 6706120688 TAK TAK NIE 17
 6706120095 TAK NIE TAK 11
 6706120247 NIE NIE NIE 16
 6706120851 NIE NIE NIE 19
 6706120222 TAK NIE TAK 16
 6706120340 NIE NIE TAK 5
 6706120395 TAK TAK NIE 9
 6706120907 TAK NIE TAK 19
 6706120392 NIE NIE NIE 12
 6706120474 TAK NIE TAK 1
 6706120134 TAK NIE TAK 18
 6706120494 NIE TAK NIE 18
 6706120481 TAK NIE TAK 19
 6706120792 TAK NIE TAK 19
 6706120821 NIE TAK TAK 5
 6706120187 NIE TAK TAK 17
 6706120823 NIE NIE NIE 11
 6706120905 TAK NIE NIE 14
 6706120637 NIE NIE NIE 11
 6706120046 TAK NIE TAK 3
 6706120624 TAK TAK TAK 13
 6706120291 TAK NIE TAK 12
 6706120478 TAK NIE NIE 14
 6706120531 NIE NIE NIE 3
 6706120197 TAK TAK NIE 10
 6706120122 TAK TAK TAK 0
 6706120604 NIE TAK TAK 9
 6706120535 NIE NIE TAK 2
 6706120752 NIE TAK TAK 13
 6706120760 NIE NIE TAK 17

Zlicz wszystkie twierdzące odpowiedzi na każde z pytań. Policz, ile osób odpowiedziało "TAK" na wszystkie pytania. Wyznacz maksymalną i minimalną liczbę punktów uzyskanych w quizie. Dodatkowo, oblicz też średnią liczbę punktów w quizie. Sformatuj pięknie tabelkę. Wyróżnij i dobrze opisz wyznaczone przez siebie liczby. Zrób wykres, najlepiej obrazujący podział odpowiedzi "TAK" i "NIE" w każdym pytaniu. W tym celu musisz wybrać najbardziej odpowiedni typ wykresu. Nie zapomnij dokładnie opisać wykresu (tytuł, etykiety,..) Wyznacz każdemu ocenę z quizu, jeśli punktacja jest następująca: 0-10 pkt: ndst, 11-20 pkt: dst, 21-30pkt: db, a powyżej 30: bdb.

Wynik zapisz do pliku o nazwie quiz z odpowiednim rozszerzeniem.

□

Rozwiązanie:

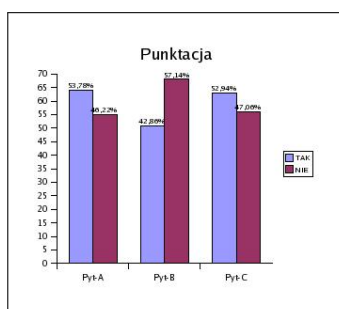
Jest to złożone zadanie z arkusza kalkulacyjnego. Najpierw jednak wczytujemy do niego dane: skopiuj je z pliku PDF, korzystając z narzędzia do zaznaczania tekstu, do pliku tekstowego, a następnie wczytaj te dane do arkusza kalkulacyjnego. (Pa-

miętaj by jako typ pliku podać **Tekst CSV**. Podczas importu może będzie trzeba scalić separator.)

Już w arkuszu, będziesz potrzebować dodatkowej kolumny, gdzie zliczysz dla danej osoby, ile udzieliła twierdzących odpowiedzi. Wykorzystasz funkcję **LICZ.JEŻELI**.

Użyj funkcji **MAX** i **MIN**, by wyliczyć skrajne wyniki w Quizie. Ze **ŚREDNIĄ** i formatowaniem nie powinieneś mieć kłopotów.

Najlepiej tutaj skorzystać z wykresu słupkowego, gdzie w kolejnych seriach podasz dwa słupki: na **TAK** i na **NIE**:



Wykres, jaki wydaje się całkiem dobry. Jeszcze lepszy wykres miałby powiększone czcionki wyrażające procenty. Nie musisz tego umieć, chociaż możesz.

Aby wystawić oceny, oczywiście musisz skorzystać z funkcji **WYSZUKAJ.PIONOWO**. Punktację musisz zamieścić w osobnej tabelce.

Oczywiście, wszystko zapisujemy do pliku **ODS**.

Zadanie 17.3.7. Będziemy działać na poniższym tekście, którego wersja elektroniczna znajduje się w pliku **ryby.txt**.

Ryby

Każdemu z nas słowo ryba kojarzy się chyba w jakiś określony sposób. Jedni widzą ją w postaci smukłego rekina, inni jako grupę jaskrawych rybek przemykających wśród rafy koralowej w wodach tropikalnych, jeszcze inni błędnie kojarzą z tym słowem wieloryby i delfiny, które przecież są wodnymi ssakami. Ryby to w rzeczywistości ogromna i wielce zróżnicowana grupa fascynujących stworzeń, tak jak nikt inny przystosowanych do życia w wodzie. Ryby można podzielić na trzy główne grupy, z których każda (wskazuje na to zewnętrzna i wewnętrzna budowa ciała) powstała całkowicie odrębną drogą ewolucji. Pomiędzy tymi grupami występują różnice tak duże jak np. pomiędzy gadami i ssakami. Niemniej można jednak dokonać pewnych uogólnień. Bardzo duża większość ryb żyje tylko w wodzie, oddycha za pomocą skrzel, ciało ma pokryte łuskami i pływa dzięki płetwom. Wszystkie ryby należą do kręgowców, czyli mają rozbudowany wokół kręgosłupa szkielet wewnętrzny, a nie zewnętrzną osłonę w postaci np. muszli czy pancerzyka. Kręgowce podzielić można na pięć dużych grup: ryby, płazy, gady, ptaki i ssaki. Każdy z nas zdaje sobie sprawę z różnorodności czterech ostatnich grup. A przecież ryby obejmują prawie tyle samo gatunków, co pozostałe cztery

grupy razem wzięte - około 21000!

Budowa wewnętrzna ryb

Typowa ryba ma równie wiele narządów co gady ptaki czy nawet ssaki. Szkielet u ryb kostnych (np. okoń) ochrania narządy wewnętrzne, które zazwyczaj umieszczone są w dolnej części ciała. Mózg odbiera informacje o świecie zewnętrznym, przekazywane za pomocą bodźców od narządów zmysłowych: oczu i linii bocznej, a także kieruje grupą mięśni odpowiedzialnych za pływanie oraz manewrowanie ciałem w wodzie. Zamiast płuc, które posiadają zwierzęta oddychające powietrzem atmosferycznym, ryby posiadają skrzela, które umożliwiają oddychanie tlenem rozpuszczonym w wodzie - woda obmywa delikatne, aczkolwiek bardzo silnie ukrwione płatki skrzelowe. Tlen jest wiązany przez hemoglobinę zawartą we krwi i rozprowadzany po całym ciele przez serce. Właśnie serce ryby rozprowadza krew poprzez sieć naczyń. Ważnym elementem budowy ciała ryby jest pęcherz pławny - uwypuklenie jelita. Służy on do kontrolowania pławności - pomaga rybie zanurzać się i wynurzać na określoną głębokość.

Łuski

Ciało znakomitej większości ryb pokryte jest warstwą przezroczystych, często tylko delikatnie połyskujących płytek zwanych łuskami. U każdego gatunku ryb różnią się one wielkością i kształtem, ale przeciętne łuski ryb kostnych są małe, lekko zaokrąglone, umiarkowanie elastycznie i jednowarstwowe. Ponadto istnieją jeszcze trzy inne podstawowe typy łusek. U rekinów i płaszczyk - plzkoidalne, często nazywane też zębami skórnymi. U ryb trzonopłetwych łuski posiadające aż cztery warstwy - kosmoidalne, a u łuskostów (np. niszczuka) ganoidalne o kształcie romboidalnym. Część ryb nie posiada łusek, a jedynie bardzo mocną skórę. Oślizłość ryb wcale nie jest spowodowana łuskami, tylko specjalnym śluzem produkowanym przez śródskórne gruczoły - ułatwia on rybom pływanie, ponieważ znacznie zmniejsza tarcie.

Płetwy

Płetwy umożliwiają rybom ruch, a i nieruchome zawiśnięcie w toni dokładnie w wybranym miejscu. Większość ryb wykorzystuje płetwy, a przede wszystkim zazwyczaj największą płetwę ogonową w charakterze źródła napędu. Z kształtu płetw można bezbłędnie wywnioskować, jaki tryb życia wiecie ryba. Długie i wąskie płetwy piersiowe i wysmukła, głęboko wcięta ogonowa jak u tuńczyka, wskazuje na umiejętność szybkiego pływania. Natomiast szerokie płetwy piersiowe i jednolity ogon praktycznie pozbawiony wcięcia spotkać można wśród żyjących przy dnie lub wśród raf koralowych ryb, które nie mogą poszczycić się umiejętnością dobrego pływania.

Źródło: <http://www.ryby-pl.com>

Na początku dokumentu wpisz swoje imię i nazwisko oraz dzisiejszą datę.

Wszystkie wyrazy "ryba" zamień na "mątwą".

Sformatuj elegancko tekst.

Uzupełnij tekst spisem treści automatycznie tworzonym na końcu dokumentu.

Zapisz wszystko do pliku w formacie PDF pod nazwą `danie.pdf`. □

Rozwiązanie:

Najwygodniej skorzystać z systemu L^AT_EX, choć w OpenOffice również wykonamy zadanie bez trudu. Zamiana ryb na mątwy była dyskutowana gdzie indziej, zob. zadanie 17.3.3. Swoje dane i datę wpisujemy jako pozycje `\author{}` i `\date{\today}`. Formatowanie sprowadza się do zaznaczenia tytułu (Mątwy) i śródtytułów w tekście. Spis treści generujemy standardowo (`\tableofcontents`), możemy jeszcze wykorzystać pakiet `hyperref` i komendę `\url{...}` do sprytnego zaznaczenia adresu strony WWW.

POWODZENIA! Pamiętaj, ocenie będą podlegać *wyłącznie* rozwiązania zawarte w mailu, który wyślesz do egzaminatora!

17.3.3 Inne pomysły na zadania egzaminacyjne

□ **Zadanie 17.3.8.** Studenci dostają *szczegółową* drukowaną instrukcję, jak publikować strony WWW na zadanym koncie na serwerze (łącznie z nazwą użytkownika i hasłem). Następnie mają uaktualnić informacje na zamieszczonej tam wcześniej stronie. □

□ **Zadanie 17.3.9.** Przygotować zadane zdjęcie znajdujące się pod adresem http://imgsrc.hubblesite.org/hu/db/2005/01/images/a/formats/full_tif.TIFF do publikacji na stronie WWW. □

Rozwiązanie: Ściągamy obrazek na dysk. Konwertujemy go GIMPem (Obraz → Skaluj obraz) tak, by zmniejszyć jego rozmiary do kilkuset (mniej więcej 200) pikseli. Następnie zapisujemy (Plik → Zapisz jako) pod nazwą `obrazek.jpg` — rozszerzenie nazwy wpisujemy z klawiatury! Jako poziom kompresji dajemy około 0.8.

□ **Zadanie 17.3.10.** Zrobić stronę WWW na podstawie danego tekstu ze zdjęciem i elegancko sformatować przy użyciu stylu używanego na stronach www.mimuw.edu.pl.

Tekst jest zadany w postaci pliku Writera, `notatka.odt`.

Wynikiem zadania ma być spakowany do pliku `strona.zip` zestaw plików potrzebnych do wyświetlenia tej strony na komputerze połączonym z Internetem. Uwaga: pliki niepotrzebne będą traktowane jako błąd w rozwiązaniu. □

Rozwiązanie: Mamy do wyboru trzy drogi:

- eksportować plik do formatu HTML wprost z Writera, a następnie edytować jego kod źródłowy w HTML
- zapisać tekst Writera jako „tekst zakodowany” do pliku ASCII, a następnie

dopisać do niego znaczniki HTML

- przepisać wszystko od zera w ulubionym edytorze stron WWW

Okazuje się, że optymalne jest pierwsze rozwiązanie, ewentualnie w połączeniu z drugim. Po zapisaniu tekstu do formatu HTML, obrazek zamieszczony w tekście łąduje do osobnego pliku JPG (lub np. PNG, w zależności od tego, w jakim formacie był oryginał) o nazwie w rodzaju `strona_html_m5066d3d6.jpg`. Teraz tylko musimy *pouzuwać* (prawie) wszystkie znaczniki HTML z pliku HTML, a następnie na nowo wpisać własne. Musimy też uzupełnić część nagłówkową pliku HTML. Struktura dokumentu jest prosta, więc nie powinniśmy mieć z tym większych kłopotów. Jeśli jednak przeraża nas natłok znaczników wstawianych przez Writera do generowanego przezeń pliku HTML, możemy teraz zapisać oryginalny plik Writera do pliku tekstowego i „obrobić” go od zera, korzystając z gotowego obrazka.

Do pliku ZIP pakujemy więc dwa pliki: `strona.html` i `strona_html_m5066d3d6.jpg`. Oczywiście pliku CSS nie musimy pakować, bo jest on dostępny przez internet (wystarczy, że w pliku HTML podamy prawidłowy adres dla pliku CSS)!

Zadanie 17.3.11. Zrobić stronę WWW na podstawie danego tekstu ze zdjęciem i elegancko sformatować przy użyciu stylu używanego na stronach www.mimuw.edu.pl.

Tekst jest zadany w postaci zestawu: plik tekstowy ASCII `notatka.txt` i zdjęcie `athlon.tif` w formacie TIFF, do zamieszczenia na samym końcu dokumentu. Wynikiem zadania ma być spakowany do pliku `strona.zip` zestaw plików potrzebnych do wyświetlenia tej strony na komputerze połączonym z Internetem. Uwaga: pliki niepotrzebne będą traktowane jako błąd w rozwiązaniu.

Rozwiązanie: Zadanie podobne do innego w tym stylu, tym razem musimy edytować plik tekstowy (którego nazwę zmieniamy na samym początku na np. `strona.html`), wstawiając doń właściwe znaczniki HTML (w tym znaczniki tytułów, akapitów plus nagłówek `<head>` z informacją o kodowaniu polskich znaków!). Zdjęcie w formacie TIFF nie nadaje się do publikacji na WWW, dlatego trzeba je skonwertować do formatu np. JPG, `zdjecie.jpg`, pamiętając o ewentualnym zmniejszeniu rozdzielczości w celu zminimalizowania wagi obrazka. Do pliku ZIP pakujemy więc dwa pliki: `strona.html` i `zdjecie.jpg`.

Zadanie 17.3.12. Utworzyć katalog Egzamin, w nim podkatalogi Zadanie1, ..., Zadanie5, w których będą rozwiązania kolejnych zadań. Na koniec egzaminu należy spakować zawartość katalogu Egzamin i wysłać pocztą elektroniczną na zadany adres e-mail.

Zadanie 17.3.13. Jaki adres zdecydowałbyś się podać, by załogować do hipotetycznego systemu obsługującego Twój elektroniczny indeks (i dlaczego):

1. `http://www.uw.edu.pl/indeks`
2. `https://www.uw.edu.pl/indeks`
3. `html://www.uw.edu.pl/indeks`

4. [htm://www.uw.edu.pl/indeks](http://www.uw.edu.pl/indeks)

□

Rozwiązanie: To oczywiście niemożliwe, że tak proste zadanie pojawi się na prawdziwym egzaminie. Ale na wszelki wypadek potwierdźmy: oczywiście, należy wpisać adres

<https://www.uw.edu.pl/indeks>

— i żaden inny. Jeśli system akceptuje połączenie szyfrowane <https://...> to mamy pewność, że nikt nie przechwyci np. naszego hasła i nie narobi nam później kłopotów na uczelni.

A co zrobić, jeśli system do obsługi elektronicznego indeksu *nie obsługuje* takiego bezpiecznego połączenia¹, czy w takim razie opcjonalnie wybrać <http://...>? Odpowiedź jest jedna: NIE. Nie używać w ogóle takiego systemu.

□ **Zadanie 17.3.14.** W zadanym dokumencie `ryby.txt`, zamień wszystkie wystąpienia słowa 'ryba' na 'mątwą', pamiętając o odmianie tych wyrazów.

□

Rozwiązanie: To oczywiście bardzo łatwe, bo przecież mątwą-ryba, mątwie-rybie, mątw-ryb, itd. A więc wystarczy w całym tekście zamienić „ryb” na „mątw”! Ale to zadanie ma pułapki:

- Co się stanie np. z wyrazem „wielorybów”? Zamieni się na „wielomątwów”! Aby się tego ustrzec, musimy po takiej globalnej zamianie sprawdzić pisownię dokumentu: i tak nie zawadzi, a wychycimy oprócz „wielomątwy” także „mątwek”, które chyba nabezpieczonej będzie zastąpić „małych mątw”.
- A jeśli było zdanie zaczynające się dużą literą: „Ryby to takie zwierzęta...”? Dostaniemy „mątwy to takie zwierzęta...”. Na to może pomóc stosowanie wyrażeń regularnych.
- Adres internetowy „www.ryby.com.pl” zmieni się na „www.mątwy.com.pl”, a to na pewno nie było naszym zamiarem.

Jeśli więc nie chcemy niespodzianek, to przy takich wielkich zamianach raczej należy krok po kroku zastanowić się przy każdej pojedynczej zamianie czy o to właśnie nam chodziło.

□ **Zadanie 17.3.15.** Zamieść na stronie WWW o nazwie `lotnictwo.html` plik graficzny `samolot` w formacie PNG, znajdujący się w katalogu `Egzamin1`

□

Rozwiązanie: Zadanie złośliwe dla użytkowników Windows, gdyż nasz plik nazywa się dokładnie `samolot.PNG`, co łatwo przeoczyć. Otóż w Windows rozszerzenia plików nie dość, że są ukrywane przed użytkownikiem, to dodatkowo przeglądarka traktuje plik `samolot.PNG` (dopóki znajduje się lokalnie na komputerze) tak samo, jakby to był plik z rozszerzeniem `samolot.png`. Niestety, tak *nie działają* adresy WWW, gdzie wielkość liter ma znaczenie. Aby więc uniknąć takiej niespodzianki, należałoby w Windows dokładnie sprawdzić, jakie jest rozszerzenie pliku, jaki nam dano.

Poza tym, oczywiście liczy się to, jak napisano stronę WWW.

¹Zagrożenie jest realne. Np. do czerwca 2006, kiedy powstawało to zadanie, system obsługi kont bibliotecznych w BUWie nie dawał takich zabezpieczeń.

□ **Zadanie 17.3.16.** Zadany jest duży, niesformatowany tekst w formacie ODT. Należy odszukać w nim wszystkie rozdziały i podrozdziały (są oznaczone specyficznymi: rozdziały słowem ROZDZIAŁ, a podrozdziały znacznikiem *) i wprowadzić je do spisu treści. Następnie należy wstawić do tekstu ilustracje znajdujące się w plikach `ilustr0xx.png` na stronie WWW niniejszego egzaminu w miejscach zaznaczonych w tekście [ILUSTR XX]. Należy zrobić spis tych ilustracji. Wreszcie, w pliku BIBLIO.html jest spis pozycji bibliograficznych, które należy uwzględnić w tekście. Obecnie miejsca odwołań do literatury są oznaczone jako [CITE XYZ], gdzie XYZ jest kodem pozycji bibliograficznej. □

Rozwiązanie: Klasyczne zadanie na nadanie struktury plikowi ODT i sporządzenie automatycznych indeksów. To trzeba umieć robić rutynowo!!!

□ **Zadanie 17.3.17.** Zadany jest duży plik tekstowy ASCII `ksiazka.txt` zapisany w kodowaniu ISO-8859-2. Należy odszukać w nim wszystkie rozdziały i podrozdziały (są oznaczone specyficznymi: nazwy rozdziałów są zaznaczone `\sekcja{...}`, a podrozdziały `\podsekcja{...}`) i wprowadzić je do spisu treści. Następnie należy wstawić do tekstu ilustracje znajdujące się w pewnych plikach na stronie WWW niniejszego egzaminu w miejscach zaznaczonych w tekście `\rysunek{pi16/...}`. Należy zrobić spis tych ilustracji. Wreszcie, w pliku `ksiazka.bbl` jest spis pozycji bibliograficznych, które należy uwzględnić w tekście. Obecnie miejsca odwołań do literatury są oznaczone jako `\cite{XYZ}`, gdzie XYZ jest kodem pozycji bibliograficznej.

Wynikiem ma być plik w formacie PDF.

□

Rozwiązanie: Klasyczne zadanie na nadanie struktury plikowi tekstowemu i sporządzenie automatycznych indeksów. Krótka rzut oka na dane pokazuje, że mamy do czynienia z dokumentem o formie prawie jak dokument \LaTeX 'a: wystarczy `\sekcja` zamienić na `\section`, itp. Wielką ulgą jest to, że bibliografia jest już w formacie \LaTeX 'a, podobnie jak odwołania do niej. Tak więc szybciej dopisujemy niezbędną preambułę do dokumentu w formacie \LaTeX i reszta jest oczywista. To trzeba umieć robić rutynowo!!!

Musimy jednak uważać, bo w naszym tekście występują znaki „%”, których znaczenie w \LaTeX 'u jest odmienne (oznaczają linie komentarza). Musimy więc znak procenta zapisać jako `\%`.

Oczywiście, jeśli ktoś chce, może także pomęczyć się z OpenOffice.

□ **Zadanie 17.3.18.** W plikach `lista.txt`, `lista*.jpg`, `lista.html` oraz `lista*.xls` znajdują się informacje o populacji łośa w Nibylandii. Na podstawie tych plików utwórz jeden dokument w formacie PDF, o nazwie `polana.pdf` — w którym zamieścisz notatkę na temat łośi w Nibylandii, korzystającą ze *wszystkich* dostępnych informacji. □

Rozwiązanie: Aby wygenerować dokument w formacie PDF, skorzystamy z OpenOffice, ze względu na występujący plik XLS (arkusz kalkulacyjny w formacie Excela). Reszta jest standardem.

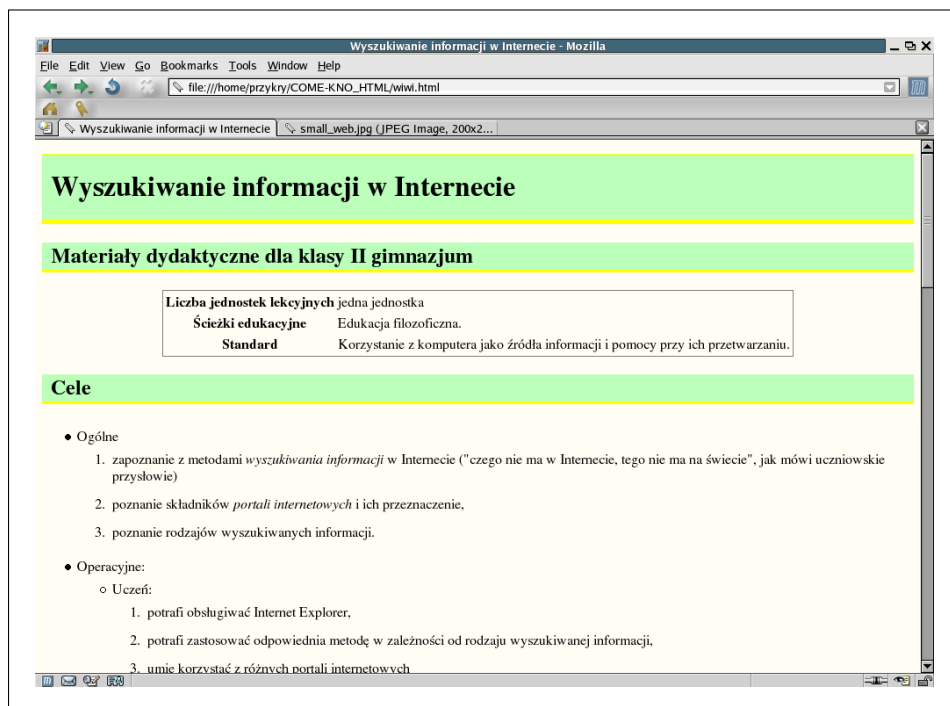
□ **Zadanie 17.3.19.** W zadanym tekście, który znajduje się w pliku² `wiwi.txt`, należy we właściwy sposób skorzystać ze znaczników strukturalnych HTML i utworzyć zeń poprawną stronę WWW. Formatowanie strony WWW należy przeprowadzić wyłącznie przy użyciu CSS; nie ma obowiązku, by było ono dokładnie takie, jak we wzorze rozwiązania, zamieszczonym w pliku `wiwi.pdf` — jest to tylko sugestia. *Strukturę* całego tekstu najlepiej sprawdzić, podglądając jego wersję w formacie PDF. Plik `wiwi.txt` zawiera w sobie dodatkowe wskazówki, precyzujące, co i gdzie trzeba wstawić. Są one oznaczone

>>> WSKAZOWKA:

Oczywiście w pliku HTML te wskazówki trzeba będzie usunąć. Rozwiązaniem jest komplet: plik `wiwi.html` zawierający stronę WWW oraz plik `wiwi.css`, zawierający reguły prezentacji tego tekstu. □

Rozwiązanie:

Zaczynamy od zachowania pliku `wiwi.txt` pod nazwą `wiwi.html`.



Wygląd tego pliku

W pomocniczym pliku PDF widzimy, że dokument ma prostą strukturę: tytuły, podtytuły oraz liczne wyliczenia. Odrobinę bardziej skomplikowane elementy to: tabela oraz obrazek: na pewno w tekście będą wskazówki, skąd go pobrać.

Formatowanie zostawiamy sobie na sam koniec, na razie bierzemy się do nadania tekstowi struktury. Pomagając sobie plikiem PDF, oznaczamy tytuły tekstu

²Autorka oryginalnej strony WWW z której pochodzi tekst: mgr inż. Angela Łakomy, zob. http://angela_janik.w.interia.pl/Wyszukiwanie_informacji_w_Internecie.htm.

znacznikami <h1>, a podtytuły — znacznikami <h2>, na przykład

```
<h1>Wyszukiwanie informacji w Internecie</h1>
```

```
<h2>Materiały dydaktyczne dla klasy II gimnazjum</h2>
```

Potem konsekwentnie nadajemy dalszą strukturę tekstowi. Uwaga: tego nie da się wygodnie zrobić korzystając z Kompozytora Stron Mozilli! Dlatego prowadzimy edycję tekstu w edytorze ASCII albo w Notatniku. Warto zwrócić uwagę na dwa istotne miejsca, które łatwo przeoczyć:

- cytata „czego nie ma w Internecie, tego nie ma na świecie”: oczywiście, usuwamy cudzysłowy i zastępujemy je znacznikami <q>
- wezwanie „*Koniecznie* tylko *praca samodzielna*!!!”: jest to tekst z wyraźnie zaznaczoną emfazą. Dlatego należy zastosować znaczniki

Następnie dodajemy do tekstu nagłówek <head>, nie zapominając zamieścić tam informacji o kodowaniu polskich liter. Oczywiście nie wiemy, jakie ono jest! Dlatego musimy zrobić mały test i sprawdzić (w dobrej przeglądarce, np. Firefoxie) czy przyjęte kodowanie tekstu jest adekwatne.

Jeśli chodzi o zdjęcie, nie używamy kopii tego obrazka, tylko wklejamy go wprost z oryginalnej lokalizacji:

```
<div style="text-align:center;" >
  <div style="text-align:center;" >
    <br>
    <span style="font-size:smaller;" >
      Źródło: <cite>Teleskop Hubble'a,
        <a
          href="http://www.hubblesite.org">www.hubblesite.org</a></cite><br>
    </span>
  </div>
  Co na to kosmici?
</div>
```

Zauważmy, że *wbrew wzorcowi* dodaliśmy trochę treści: mianowicie informacje o źródle obrazka — to bardzo ważne!

W końcu, tworzymy plik CSS według własnego gustu.

Gdy wszystko jest gotowe, sprawdzamy poprawność HTML i CSS przy użyciu stosownych Validatorów, poprawiamy niedoróbki — i gotowe.

No i jeszcze jedna uwaga: ktoś sprytny(?), a niezbyt uczciwy, kto wejdzie po prostu na stronę http://angela_janik.w.interia.pl/Wyszukiwanie_informacji_w_Internecie.htm i ją do siebie skopiuje, nic na tym nie zyska: strona ta — choć ładnie wygląda — jest *wyjątkowo źle* opracowana w HTML i w konsekwencji byłaby bardzo nisko oceniona!

□ **Zadanie 17.3.20.** Z zadanego pliku tekstowego `ryby.txt`, zrobić sformatowane dokumenty w formatach:

- HTML,

- OpenOffice,
- L^AT_EX,
- PDF

□

Rozwiązanie: Zaczynamy od OpenOffice, wczytując nasz plik jako tekst zakodowany.

Tekst na którym mamy pracować, jest zakodowany w formacie ISO-8859-2, co musimy odkryć metodą prób i błędów, patrząc, jak wyglądają polskie litery, gdy wczytamy go do OO przy jednym z kilku popularnych kodowań.

Gdy to już wiemy, to przygotowujemy jego formatowanie (jak widać, wystarczy tu wyłącznie formatować nagłówki) i zapisujemy do pliku ODT, a także możemy wyeksportować go do PDF.

Następnie, bierzemy się za L^AT_EX, gdzie musimy: podać znów właściwe kodowanie tekstu, a także zaznaczyć w nim nagłówki. Wynik pracy zapisujemy z rozszerzeniem `.tex`. Sprawdzamy, czy nie pomyliliśmy się, „texując” plik T_EX do formatu DVI albo PDF.

W końcu, zapisujemy oryginalny plik `ryby.txt` jako `ryby.html` i w edytorze tekstu wprowadzamy właściwe znaczniki strukturalizujące.

□ **Zadanie 17.3.21.** Dowiedz się, jakie są średnice wszystkich dziewięciu planet Układu Słonecznego. Następnie zamieść tę informację w tabelce w której dodatkowo wyliczysz ich objętości (jeśli nie pamiętasz wzoru, z pewnością znajdziesz go w Internecie) i którą posortujesz według objętości: od największej do najmniejszej. Ładnie sformatowaną tabelkę, wraz z wykresem ilustrującym procentowy udział objętości planet w Układzie Słonecznym, zamieść w krótkiej notatce na ten temat, którą zapiszesz w formacie PDF w pliku `planety.pdf`. □

Rozwiązanie: Łatwe.

□ **Zadanie 17.3.22.** Przygotuj do zamieszczenia na stronie WWW, skomplikowany tekst zadany w postaci pliku OpenOffice. Plik wynikowy powinien nosić nazwę `wynik` i mieć właściwe rozszerzenie. □

Rozwiązanie: Rozwiązanie najprostsze i najlepsze: eksportować go do formatu PDF!

□ **Zadanie 17.3.23.** Ze strony <http://www.mimuw.edu.pl/~przykry/Zajecia-Pedagogika> ściągnij plik `biblio.sxc` i zapisz go na dysku. Znajdź największą i najmniejszą liczbę wypożyczeń w jednym miesiącu. Oblicz, ile w sumie książek wypożyczono z biblioteki. Tę liczbę dopisz na końcu dokumentu tekstowego `danie`, jaki wcześniej stworzyłeś. Zrób wykres obrazujący strukturę wypożyczeń.

□

Rozwiązanie: Łatwe. Praca w arkuszu kalkulacyjnym.

□ **Zadanie 17.3.24.** Wyszukaj stronę internetową Wydziału Psychologii na UW. Adres tej strony zapisz w pliku `danie.sxw`. Na stronie internetowej `index.html` zamieść swoje imię i nazwisko, kilka słów o sobie oraz dodatkowo, w sekcji „Linki” — link do Wydziału Psychologii UW. Uzupełnij `index.html` o przepis na zagotowanie wody w postaci listy trzech czynności: woda do garnka, zapałka do gazu, garnek na gaz. Pamiętaj, by użyć właściwych znaczników HTML! Ściągnij z adresu <http://www.mimuw.edu.pl/~przykry/WWW/Zajecia-Pedagogika> plik o nazwie `dym.tif`. Sprawdź, jaki jest rozmiar obrazka `dym.tif` (w bajtach i w pixelach) i wpisz ten rozmiar (w kilobajtach! ...i w pixelach) do pliku `danie`. Przekształć obrazek tak, by miał objętość około 100KB przy wciąż dobrej jakości; efekt zapisz pod nazwą `dymek.jpg`. Ten mały plik wklej do tekstu `danie`. Podpisz go według uznania. □

Rozwiązanie: Łatwe.