

Programowanie komputerowe

Zajęcia 3

Typ logiczny

- Wartości logiczne są reprezentowane przez typ `bool`.
- Typ `bool` posiada tylko dwie wartości: `true` i `false`.
- Zamiast wartości logicznych można używać wartości całkowitych (`int`) lub znakowych (`char`). Wtedy wartość `0` jest interpretowana jako `false`, a każda niezerowa wartość jako `true`.
- Przykład:

```
int n=10;
while (n-->0)
    cout << n;
```

drukuje 9876543210

Instrukcje przypisania

Poza zwykłą instrukcją przypisania, powodującą ustawienie wartości zmiennej na podane wyrażenie, istnieje wiele innych, np.

- += dodaj i przypisz ($a+=b$ jest równoważne $a=a+b$)
- -= odejmij i przypisz ($a-=b$ jest równoważne $a=a-b$)
- *= pomnóż i przypisz ($a*=b$ jest równoważne $a=a*b$)
- i wiele innych tworzonych analogicznie.

Pozwalają one na skrócenie zapisu programu.

Operator warunkowy ?

Składnia:

$w ? a : b$

- w – warunek logiczny
- a, b – wyrażenia, powinny być tego samego typu

Wartością takiego wyrażenia jest a jeśli warunek w jest spełniony i b w przeciwnym przypadku.

Przykład: wartością wyrażenia

$a > b ? a : b$

jest większa z liczb a, b .

Instrukcje `break`, `continue` i `switch`

Instrukcje `break` i `continue` służą do przerywania pętli.

- `break` powoduje przerywanie pętli i przejście do instrukcji po pętli.
- `continue` powoduje przerywanie aktualnego obiegu pętli i bezpośrednio przejście do warunku pętli; jeśli ten jest spełniony, wykonuje się kolejny obieg pętli.
- Instrukcji tych należy w miarę możliwości unikać.

Instrukcja `switch` to instrukcja warunkowa służąca do wyboru spośród wielu wariantów, zależnych od wartości pewnej zmiennej całkowitej. Używa się jej w pewnych konkretnych zastosowaniach, które nie pojawią się w trakcie tego kursu.

Rekursja

- Rekursja to technika programowania polegająca na wywoływaniu funkcji przez siebie samą.
- Każdą funkcję, którą można zapisać za pomocą iteracji można zapisać za pomocą rekursji i na odwrót.

Przykład: Funkcję silnia można zdefiniować na dwa sposoby:

- $n! = 1 * 2 * 3 * \dots * n$ // iteracja
- $n! = n * (n-1)!$ dla $n > 0$ oraz $0! = 1$ // rekursja

Funkcja silnia na dwa sposoby

Iteracja:

```
int silnia(int n) {  
    int w=1;  
    for(int i=1; i<=n; i++)  
        w=w*i;  
    return w;  
}
```

Rekursja:

```
int silnia(int n) {  
    if(n>0)  
        return n*silnia(n-1);  
    return 1;  
}
```

Ćwiczenia

1. Napisać funkcję

```
int nwd(int a, int b)
```

która oblicza największy wspólny dzielnik dwóch liczb.

2. Napisać funkcje `ciag`, `pytaj` i `potega` z poprzednich zajęć używając rekursji.

3. Napisać dwie funkcje

```
void rozklad(int n)
```

które drukują rozkład liczby na czynniki pierwsze. W jednej użyć iteracji, w drugiej rekursji.

Ćwiczenia (2)

4. Napisać funkcję

```
int dwumian(int n, int k)
```

która oblicza dwumian Newtona.

5. Napisać funkcję

```
int pierwiastki(double a, double b, double c)
```

która drukuje pierwiastki wielomianu ax^2+bx+c . Wynikiem powinna być liczba pierwiastków.

6. Napisać funkcję

```
int najmniejszyNieDzielnik(int n)
```

która zwraca najmniejszą liczbę naturalną, która nie jest dzielnikiem n .

Ćwiczenia (3)

7. Napisać funkcję

```
int sumaKwadratow(int n)
```

która zwraca sumę kwadratów liczb 1,2, ... , n.

8. Napisać funkcję

```
void trojkatyProstokatne(int n)
```

która wypisuje na ekranie wszystkie trójki liczb (a,b,c) takie że $0 < a < b < c < n$ oraz $a^2 + b^2 = c^2$.

9. Napisać funkcję

```
bool czySumaKwadratow(int n)
```

która zwraca `true` jeśli `n` da się zapisać jako suma kwadratów różnych liczb naturalnych.