

ZSI . Egzamin z NMJP-PO . 21.III.2000

Zad 1.

Pewna dobrze prosperująca firma programistyczna realizuje jednocześnie wiele projektów. Nad każdym projektem pracuje zespół składający się z kierownika, analityka, projektanta i programistów (co najmniej jednego). Każdy członek zespołu jest pracownikiem firmy. Jeden pracownik może być jednocześnie zatrudniony w kilku zespołach. Gdy zespół dostaje polecenie stworzenia na podstawie otrzymanej specyfikacji wymagań programu, najpierw zleca analitykowi stworzenie specyfikacji, następnie przekazuje tę specyfikację projektantowi, który tworzy projekt będący kolekcją zadań. Kierownik rozdziela te zadania (w miarę możliwości równo) między programistów, którzy z kolei generują moduły realizujące wskazane zadania.

Należy zaimplementować (w Smalltalku) klasy służące do reprezentowania przedstawionej firmy (metody, atrybuty z opisem, zależności między klasami). Należy zaimplementować następujące metody (i wszystkie niestandardowe metody w nich używane, prócz wymienionych na końcu treści zadania):

- `stwórzProgram: wymagania` (metoda zespołu) jako wynik daje kolekcję wygenerowanych modułów,
- `twórzZespół: IProgr` (metoda firmy) tworzy (daje jako wynik) zespół składający się z kierownika, analityka, projektanta i IProgr programistów (jeśli nie ma tytułu w firmie zgłasza błąd). Wybiera najmniej zajętych pracowników (tzn. tych, którzy są zatrudnieni w najmniejszej liczbie zespołów),
- `wlluJestemZespołach` (metoda pracownika) podaje liczbę zespołów w jakich ten pracownik jest zatrudniony.

Należy założyć, że metody: `dajSpecyfikację: wymagania` (analityk), `dajZadania: specyfikacja` (projektant) – daje `OrderedCollection` zadań, `dajModuł: zadanie` (programista) oraz klasy `Zadanie`, `Specyfikacja`, `Wymagania` są już gotowe.

Zad 2.

Zaimplementuj w C++ klasy służące do reprezentowania prostokątnych rysunków (o bokach równoległych do boków ekranu). Każdy rysunek ma (oprócz wielu innych) metody:

- `void prostokąt(int x1, int y1, int x2, int y2, int kolorTła, int kolorRamki)` rysuje wypełniony prostokąt o bokach równoległych do boków rysunków,
- `void wyczyść()` czyści cały rysunek (na białą).

Każdy rysunek pamięta swój rozmiar (w pionie i poziomie). Rysunki mogą być implementowane jako tablice pikseli lub jako listy obiektów graficznych. Zaimplementuj:

- abstrakcyjną klasę `Rysunek` (reprezentującą wszystkie rysunki),
- klasę `TabRysunek` (reprezentującą rysunek w tablicy pikseli).

Należy zaimplementować metody `prostokąt` i `wyczyść` oraz konstruktor bezargumentowy, destruktor, konstruktor kopiujący i operator przypisania (tam gdzie są potrzebne).