Zadanie 7, Języki Programowania, 19.11.2011

Operator przypisania = (za wykładem)

Dwuargumentowy operator przypisania = służy do przypisanie jednemu obiektowy klasy treści drugiego obiektu tej samej klasy.

```
klasa & klasa::operator=(klasa &);
```

Jeśli operator= nie zostanie zdefiniowany, to zostanie on automatycznie wygenerowany (**podobnie jak konstruktor kopiujący**!), przepisane zostaną pola "składnik po składniku" (podobnie jak w przypadku automatycznie generowanego konstruktora kopiującego). W rezultacie takiego przypisania będą dwa obiekty o bliźniaczej treści.

Kłopoty natomiast mogą pojawić się wtedy, kiedy **składnikami klasy są wskaźniki** lub jeśli klasa używa **operatora new do rezerwacji miejsca w zapasie pamięci**.

Operator przypisania składa się z części:

(1) sprawdzenie, czy nie jest kopiowane siebie samego

(2) części "destruktorowej" (np. zwalnianie pamięci)

(3) części "konstruktorowej", przypominającej konstruktor kopiujący

Przykład użycia: **(jeśli operator znajduje się wewnątrz klasy):**

```
  Uczen & operator=(Uczen & u){;
    if (&u == this) return *this; //sprawdzenie, czy nie kopiuje samego siebie
    delete X; //zwalniamy pamięć dla rzeczy, które mogły mieć poprzednio zaalokowaną pamięć
    //alokujemy nową pamięć i przepisujemy wartości tak samo jak w konstruktorze kopiującym
    return *this;
 }
```

Treść zadania

1. Należy stworzyć klasę Uczen (w oddzielnych plikach .h i .cpp, do tego main w oddzielnym pliku) zawierającą trzy pola składowe:

- `fImie (std::string)`
- `fNazwisko (std::string)`
- `fOceny (int*)`
- `fLiczbaOcen (int)`

Pole 'fOceny' zawiera tablicę ocen danego ucznia. Domyślnie są to 4 oceny, same 1. Ponadto klasa zawiera zestaw metod: konstruktor domyślny, konstruktor z 4 parametrami oraz konstruktor kopiujący. Do tego należy stworzyć metody „set" dla pól `fImie`. Należy również przeciążyć operator przyrównania ("="). Oraz dodać metody:

```
double Srednia() //zwracajace srednią ocen ucznia
void Wypisz() //wypisującą wszystkie dostępne informacje o danym uczniu
void DodajOcene(double Ocena) //pozwalającą dodać ocenę Ocena do tablicy fOceny
```
(ponowna alokacja pamięci!)

2. Tworzenie programu

Uwaga: w przypadku obiektów typu „`string`" nie należy używać funkcji typowych dla pracy z ciągami znaków z C (`strcpy`, `strcmp`). Funkcje biblioteki `<cstring>` używamy dla obiektów typu char*.

Program powinien być kompilowany przy pomocy komendy „`make`" (należy stworzyć odpowiedni Makefile!)

1. Należy stworzyć trzy puste pliki: uczen.cpp, uczen.h oraz main.cpp. Do tego należy stworzyć Makefile (kolejny plik tekstowy, o nazwie „**Makefile**") z treścią:

```
all:
	g++ uczen.cpp main.cpp -o program
```

w pliku program.cpp należy dodać funkcję `int main()` zwracającą 0.

Należy skompilować tak przygotowany program używając w linii poleceń komendy: `make`.

2. Od teraz **zawsze** należy owijać definicje klas w plikach nagłówkowych w strukturę „ifndef”:

```
#ifndef _NAZWA_TOKENU
#define _NAZWA_ TOKENU
      class klasa{...}; - definicja naszej klasy
#endif
```

Działa to w ten sposób: jeśli token _NAZWA_TOKENU nie był jeszcze definiowany: wejdź do środka (zdefiniuj token i zapisz go sobie w pamięci, po czym wykonaj wszystkie instrukcje znajdujące się w środku). Jeśli już go deklarowałeś – przejdź od razu do #endif. W ten sposób już nigdy nie będziemy się przejmować kolejnością ładowanych bibliotek – jeśli zostały one wcześniej załadowane, kompilator w ogóle pominie taką instrukcje.

3. Należy w funkcji main po kolei:

- Stworzyć Makefile i owinąć nagłówek funkcji w #ifndef (0.5 p)
- Stworzyć pojedynczy obiekt Uczen u1. Jest to Jan Kowalski, o ocenach 3, 4, 5. (1 p)
- Należy wypisać owego ucznia na ekran przy użyciu metody Wypisz. (0.5 p)
- Należy stworzyć kolejny obiekt Uczen (u2), używając konstruktora kopiującego. (0.5 p)
- Należy zmienić imię tak skopiowanemu uczniowi na “Marek”, używając metody set.
- Należy wypisać Marka na ekran.
- Należy stworzyć kolejnego ucznia u3, używając konstruktora domyślnego. (0.5 p)
- Należy przypisać uczniowi u3 wartości składników ucznia u2 używając operatora =. (0.5 p)
- Należy wypisać trzeciego ucznia na ekran. (0.5 p)
- Dla trzeciego ucznia należy wypisać średnią. (0.5 p)
- Dla trzeciego ucznia należy dodać ocenę “5” do tablicy ocen, oraz ponownie wypisać średnią (0.5 p)

Dodatkowo (po zaliczeniu całego zadania):

- Zmienić klasę tak, by pole imię było typu char* zamiast std::string. (0.5 p)