

Języki Programowania, Wtorek 29.10.2013

Zadanie 3

Ideą programu jest stworzenie “bazy danych” w C++ dla sklepu z komputerami. Sklep ma miejsce na 70 komputerów (można zadeklarować tablicę statyczną na [70] komputerów), a każdy komputer ma: nazwę, wiek, prędkość procesora, czy laptop oraz cenę. Nowe komputery można dodawać, oraz wypisywać wszystko, co sklep ma na dostępne.

1) Stwórz klasę Komputer zawierającą pięć składników prywatnych: nazwa (std::string), wiek (int), prędkość procesora (double), czy laptop (bool) oraz cenę (double). Klasa powinna być podzielona na dwa oddzielne pliki: komputer.h i komputer.cpp. W trzecim pliku powinna znajdować się funkcja główna main.

Klasa powinna mieć dwie funkcje składowe (tzw. metody):

- SetNazwa(string nazwa) - ustawiająca nazwę danego komputera
- SetLaptop(bool czyLaptop) - zapisująca informację, czy komputer jest laptopem (true) czy nie (false)
- wypisz() - wypisująca na ekran informacje o danym komputerze (po spacjach wszystkie pola prywatne)

2) Należy również zaimplementować konstruktor domyślny, główny, kopiujący oraz destruktor. Dstruktor należy opatrzyć komentarzem - wypisywaniem na domyślne wyjście „Zostal uzyty destruktor”.

3) 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 instrukcję.

4) W programie głównym, należy:

- stworzyć obiekty napisanej klasy używając konstruktora domyślnego, głównego i kopiującego, wypisać je na ekran (1,5 p.)
- stworzyć wskaźniki do obiektów (operator new) napisanej klasy używając konstruktora domyślnego, głównego i kopiującego, wypisać je na ekran (0,5 p.)
- użyć operatora delete do usunięcia wskaźnika na obiekty klasy Komputer (0.5 p.)

5) Należy przygotować „bazę danych” z możliwością dodawania, usuwania oraz wypisywania komputerów używając w tym celu tablicy statycznej na 70 komputerów. W głównym programie (funkcji main) wymagana jest instrukcja typu switch - case, pytająca w pętli o podanie liczby od 1-3 lub 0.

- Wpisanie 1 - wypisanie wszystkich zapisanych komputerów na ekran. (0.5p) (cout<<zmienna)
- Wpisanie 2 - dodanie nowego komputera do “bazy danych”. (0.5 p) (cin>>zmienna)
- Wpisanie 3 - usuwanie komputera o konkretnym indeksie z „bazy danych” (należy pamiętać o przesuwaniu kolejnych komputerów w tablicy). (1p)
- Wpisanie 4 - zmiana ceny komputera o konkretnym indeksie z „bazy danych” (0.5p)
- Wpisanie 0 - powinno przerwać pętlę oraz spowodować opuszczenie programu.
- Wpisanie liczby innej niż dozwolone - program zakomunikuje, iż została wprowadzona błędna wartość.

Przydatne nagłówki (biblioteki):

```
#include <iostream>

#include <string>

#include <cstring>
```

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

Makefile

CXX = g++

CXXFLAGS = -Wall

LFLAGS =

OBJS = komputer.o main.o

all: prog

prog: \$(OBJS)
\$(CXX) \$(CXXFLAGS) \$^ -o \$@

clean:
rm -f *.o prog

.PHONY: all clean

Zadania dodatkowe:

- Dynamiczna „baza danych”: użycie klasy vector lub queue.