

Języki Programowania, Zadanie 7 – Powtórzenie

Należy stworzyć klasę `KlasaTestowa`, mającą demonstrować różne mechanizmy języka C++.

Klasa powinna zawierać pola składowe:

- `double a;` // przechowującą liczbę zmiennoprzecinkową
- `double* b;` //zawierającą 1D tablicę liczb zmiennoprzecinkowych alokowaną dynamicznie o rozmiarze `fRozmiar`
- `int fRozmiar;` //ilość elementów tablicy `b`
- `int fIlosc;` //zdefiniowane jako pole statyczne zliczające ilość obiektów typu `KlasaTestowa`

trzy konstruktory:

- domyślny – ustawia pole `a` na 0, rozmiar tablicy `b` na 100, alokuje jej pamięć oraz ustawia wszystkie 100 elementów na wartość 1.
- z 2 parametrami – ustawia pole `a` na zadaną wartość oraz alokuje rozmiar tablicy `b` na zadaną długość (wpisując w każdy element tablicy losową wartość: (`#include <stdlib.h>`, `rand()`, `#include <time.h>`, `srand(time(NULL))`) – raz na początku programu, ustawienie ziarna))
- kopiujący – tworzy kopię danego obiektu (ustawia wszystkie pola nowego obiektu na wartości takie same jak obiektu kopiowanego)
- użycie dowolnego konstruktora zwiększa pole statyczne o 1

destruktor:

- usuwa tablicę `b` oraz zmniejsza wartość pola statycznego o 1 //2 pkt (po 0.5 za każdy kons./dest.)

Metody: //0.5 pkt (łącznie za wszystkie)

- `void SetA(double A)` – ustawia pole `a`
- `void SetBelement(double val, int pos)` – ustawia element tablicy `b` na zadanej pozycji (`pos`) na zadaną wartość (`val`)
- `void SetAll(double A, double val, int pos)` – ustawia jednocześnie pole `a`, oraz zadany element tablicy `b`
- `void Wypisz()` - wypisuje na ekran wszystkie elementy klasy
- `int GetIlosc()` - funkcja statyczna, wypisuje ilość elementów typu `KlasaTestowa`

w programie powinny znaleźć się globalne funkcje:

- `void WypiszKlasaTestowa1(KlasaTestowa &kp1)` - wypisująca na ekran wszystkie elementy obiektu `kp1` – zaprzyjaźniona z klasą
- `void WypiszKlasaTestowa2(KlasaTestowa *kp1)` – wypisująca na ekran wszystkie elementy obiektu `kp1` – nie będąca zaprzyjaźnioną z klasą //1 pkt (po 0.5 na funkcje)

oraz klasa powinna mieć przeciążone operatory:

- operator `=` - przypisuje obiektowi z lewej strony znaku parametry obiektu stojącego z prawej strony znaku
- operator `[]` - zdefiniowany tak, by mógł stać po obu stronach znaku `=`, użycie z wartością 1 wypisuje pole `a`, z dowolną inną wypisuje ostatni element tablicy `b`
- operator `<<` - działa identycznie jak metoda `Wypisz()` //1.5 pkt (po 0.5 na operator)

W programie głównym należy przetestować działanie wszystkich elementów klasy `KlasaTestowa`.

Dodatkowo (dla chętnych):

- przeciążenie operatora `+` - należy zwrócić liczbę która jest sumą składnika a pierwszej klasy oraz wszystkich elementów tablicy `b` drugiej klasy.
- dodać nowy składnik klasy `char* c` – alokowana dynamicznie tablica na 30 znaków; uzupełnić zadanie w ten sposób, by ustawiane były dla tego składnika domyślne wartości, i by składnik był wypisywany w `Wypisz`