Zadanie 7, Języki programowania, wtorek 26.11.2013

Należy stworzyć dwie klasy: klasę pomocniczą `Pomoc` oraz klasę `KlasaPokazowa`, mające demonstrować różne mechanizmy języka C++.

Klasa `Pomoc` powinna zawierać dwa pola składowe: `string nazwa` oraz `int numer`.
Klasa `KlasaPokazowa` powinna zawierać pola składowe:

- `int a;` // przechowującą liczbę całkowitą
- `Pomoc klasa;` //przechowującą obiekt innej klasy
- `double* c;` //zawierającą tablicę liczb alokowaną dynamicznie (uwaga, tablica!)
- `int fRozmiar;` //ilość elementów tablicy `c`
- `int fIlosc;` //zdefiniowane jako **pole statyczne** zliczające ilość obiektów typu `KlasaPokazowa`

trzy **konstruktory**:

- domyślny– domyślnie ustawia pole `a` na 0, `nazwa` na „napis”, `numer` na 666, rozmiar tablicy `c` na 10, alokuje jej pamięć oraz ustawia wszystkie elementy na wartość 1. Użycie konstruktora zwiększa pole statyczne o 1
- przyjmujący trzy parametry –`a`, rozmiar tablicy `c` oraz nazwę (`nazwa`), reszta jak w konstruktorze domyślnym (można oba konstruktory zapisać jako jeden)
- kopiujący – tworzy kopię danego obiektu (ustawia wszystkie pola nowego obiektu na wartości takie same jak obiektu kopiowanego)

destruktor:

- zgodnie ze zdrowym rozsądkiem

metody:

- `void SetA(int A)` – ustawia pole `a`
- `int GetA()` – zwraca pole `a`
- `void SetCpos(int pos, double val)` – w tablicy `c` na pozycji `pos` ustawia wartość `val`
- `void Wypisz()` - wypisuje na ekran pole `a`, obiekt `b` oraz tablicę `c`
- `int GetIlosc()` - **metoda statyczna**, wypisuje ilość elementów typu `KlasaPokazowa`

w programie należy zaimplementować przeciążanie **operatorów**:

- `operator[]` - zdefiniowany tak, by mógł stać po obu stronach znaku =
- `operator<<` - działa identycznie jak metoda `Wypisz()` (ale jej nie używa)
- `operator=` - przypisuje obiektowi z lewej strony znaku parametry obiektu stojącego z prawej strony znaku (napisać zgodnie ze zdrowym rozsądkiem)
- `operator+` - dodaje wartości `a` do siebie z dodawanych obiektów `KlasaPokazowa`, tablica c wynikowego obiektu powinna mieć 10 elementów ustawionych na wartość 2 (zaimplementować wewnątrz klasy), obiekt b nowej klasy powinien być taki sam jak obiekt b pierwszego z agumentów
- `operator-` - zwraca liczbę która jest różnicą `a` odejmowanych siebie obiektów (zaimplementować na zewnątrz klasy)

Program powinien działać z zamieszczoną poniżej funkcją main:

```
int main()
{
   KlasaPokazowa kp;
   kp.SetA(3);
   kp.SetCpos(2,5);
   kp.Wypisz();
   cout<<"Ilosc: "<<KlasaPokazowa::GetIlosc()<<endl;

   KlasaPokazowa kp2(3,3,"inny");
   cout<<kp[2]<<endl;
   cout<<kp2<<endl;
   cout<<"Ilosc: "<<KlasaPokazowa::GetIlosc()<<endl;
   KlasaPokazowa kp4(kp2);
   cout<<kp4<<endl;
   KlasaPokazowa *kp3;
   kp3 = kp+kp2;
   kp3->Wypisz();
   cout<<(*kp3)[1]<<endl;
   cout<<kp-*kp3<<endl;
   cout<<"Ilosc: "<<KlasaPokazowa::GetIlosc()<<endl;
   delete kp3;
   cout<<"Ilosc: "<<KlasaPokazowa::GetIlosc()<<endl;

  return 0;
}
```

Dodatkowe: (ale zapisać w oddzielnym pliku) zamienić `a` na wskaźnik!