Elektronski fakultet u Nišu
11.12.2024. godine

Objektno orijentisano programiranje

I kolokvijum

16.15 – 17.45

Opšte napomene za sve klase:

- Atributi ne mogu biti javni.
- Kreirati sve potrebne javne metode koje će obezbediti da program radi (uključujući konstruktore i destruktor).
- Izbeći na svaki način dupliranje koda u celom projektu.
- Pridržavati se osnovnih principa objektno orijentisanog programiranja.
- Zabranjena upotreba ugrađenog tipa ***string.***

Zadatak B:

Napisati program koji omogućuje rad sa različitim akvarijumima. Akvarijumi se nalaze na policama i mogu biti oblika valjka ili kvadra.

Glavni program treba da omogući sledeće:

- Pravljenje dve police na osnovu zadatog broja akvarijuma. Akvarijum se opisuje tipom vode, da li ima poklopac i treba da omogući računanje koeficijenta, koji predstavlja odnos površine stakla i zapremine.
- Akvarijum može biti oblika valjka ili kvadra. Akvarijum oblika valjak se dodatno opisuje poluprečnikom osnove i visinom, a oblika kvadra dužinom stranica.
- Dodavanje akvarijuma na policu.
- Prikaz svih akvarijuma na polici.
- Potrebno je odrediti ukupnu površinu stakla koja je utrošena za izradu svih akvarijuma na polici.
- Potrebno je odrediti ukupnu zapreminu svih akvarijuma na polici.
- Za specijalne slučajeve potrebno je omogućiti da se akvarijumi sa jedne police prebace na drugu policu na osnovu tipa vode. Prikazati tako dobijenu policu.
- Omogućiti pronalaženje akvarijuma sa najmanjim koeficijentom. Potrebno je vratiti traženi akvarijum i njegov indeks na polici.

PREDMETNI NASTAVNICI I ASISTENTI

main.cpp

```cpp
#include <iostream>
using namespace std;

void main()
{
   const char* tip_akvarijuma[] = { "akvarijum slatke vode", "akvarijum slane vode", "bez vode
- terarijum" };

   // 2 poena
   Akvarijum* p1 = new Valjak(5.0, 10.1, tip_akvarijuma[0], false);
   //p1.print();
   cout << p1 << endl;

   // 2 poena
   Akvarijum* p2 = new Kvadar(200, 40, 55, tip_akvarijuma[1], true);
   //p2.print();
   cout << p2 << endl;

   Polica p(4);
   Polica* pp = new Polica(4);

   // 2 poena
   for (int i = 0; i < p.BrojAkvarijuma() / 2; i++)
   {
      p.Add(new Valjak(100+i*5, i*3, tip_akvarijuma[(2*i)%3], i%2));
      p.Add(new Kvadar(1000+i*20.37, (i+1)*i, i*3, tip_akvarijuma[(2*i+1)%3], (i+1)%2));
   }
   for (int i = 0; i < pp->BrojAkvarijuma() / 2; i++)
   {
      pp->Add(new Valjak(100+i*5, i*3, tip_akvarijuma[(2*i)%3], i%2));
      pp->Add(new Kvadar(1000+i*20.37, (i+1)*i, i*3, tip_akvarijuma[(2*i+1)%3], (i+1)%2));
   }

   // 2 poena
   //p.print();
   cout << p << endl;

   // 2 poena
   delete pp;

   // 2
   cout << p.UkupnaPovrsinaStakla();
   // 2
   cout << p.UkupnaZapremina();

   // 3 poena
   Polica *ppp = p.PrebaciAkvarijume(tip_akvarijuma[2]);
   //ppp->print();
   cout << *ppp << endl;

   // 3 poena
   Akvarijum* pmin = nullptr;
   int ind;
   p.Min(&ind, &pmin);
   cout << ind << " " << *pmin << endl;
}
```