

Objektno orijentisano programiranje

I kolokvijum

14.15 – 15.45

Opšte napomene za sve klase:

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

Zadatak A:

Napisati program koji omogućuje rad sa različitim grafičkim oblicima koji mogu biti krugovi i pravougaonici. Figure su organizovane u kanvasima.

Glavni program treba da omogućiti sledeće:

- Pravljenje dva kanvasa na osnovu zadatog broja figura. Figura se opisuje nazivom boje i treba da omogućiti računanje koeficijenta figure, koji predstavlja odnos površine i obima.
- Figura može biti krug ili pravougaonik. Krug se dodatno opisuje poluprečnikom, a pravougaonik dužinom stranica.
- Dodavanje figure na kanvas.
- Prikaz svih figura na kanvasu.
- Potrebno je odrediti ukupnu površinu svih figura na kanvasu.
- Potrebno je odrediti ukupan obim svih figura na kanvasu.
- Za specijalne slučajeve potrebno je omogućiti da se figure jednog kanvasa iste boje prebace na drugi kanvas. Prikazati tako dobijeni kanvas.
- Omogućiti pronalaženje figure sa najvećim koeficijentom. Potrebno je vratiti traženu figuru i njen indeks na kanvasu.

PREDMETNI NASTAVNICI I ASISTENTI

main.cpp

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

void main()
{
    const char* boje[] = { "zelena", "plava", "crvena", "zuta", "roze" };

    // 2 poena
    Figura* p1 = new Krug(5.0, boje[0]);
    //p1.print();
    cout << p1 << endl;

    // 2 poena
    Figura* p2 = new Pravougaonik(399.99, 1, boje[1]);
    //p2.print();
    cout << p2 << endl;

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

    // 2 poena
    for (int i = 0; i < p.BrojFigura() / 2; i++)
    {
        p.Add(new Krug(100 + i * 5, boje[(2*i)%5]));
        p.Add(new Pravougaonik(1000 + i * 20.37, (i + 1) * i, boje[(2 * i+1) % 5]));
    }
    for (int i = 0; i < pp->BrojFigura() / 2; i++)
    {
        pp->Add(new Krug(100 + i * 5, boje[(2 * i) % 5]));
        pp->Add(new Pravougaonik(1000 + i * 20.37, (i + 1) * i, boje[(2 * i + 1) % 5]));
    }

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

    // 2 poena
    delete pp;

    // 2
    cout << p.UkupnaPovrsina();
    // 2
    cout << p.UkupanObim();

    // 3 poena
    Kanvas *ppp = p.PrebacitiPoBoji(boje[2]);
    //ppp->print();
    cout << *ppp << endl;

    // 3 poena
    Figura* pmax = nullptr;
    int ind;
    p.Max(&pmax, &ind);
    cout << ind << " " << *pmax << endl;
}
```