

## Programabilni uređaji i objektno orjentisano programiranje

### Računske vježbe 10

Projektovati klase **krug** i **kvadrat** koje su izvedene iz klase **figura**. Klasa **figura** sadrži težište kao zajedničku katakarakteristiku za sve figure, funkciju koja omogućava pomjeraj težišta za zadatu vrijednost i virtualne funkcije obim, površina i citaj. Izvedene klase treba da imaju specifične funkcije za računanje obima i površine kao i očitavanje odgovarajućih podataka članova.

```
#include <iostream>
#include <math.h>
using namespace std;

class tacka
{
private:
    float x;
    float y;
public:
    tacka(float a=0, float b=0):x(a), y(b){}
    ~tacka(){}
    float vratiX(){return x;}
    float vratiY(){return y;}
    void citajT(){cout<<x<<" "<<y;}
};

const tacka KP;

// virtuelna funkcija je funkcija članica koja je definisana (realizovana) u osnovnoj klasi
// a može biti redefinisana (ponovo realizovana) u izvedenim klasama
class figura
{
private:
    tacka teziste;
public:
    figura(tacka t=KP):teziste(t){}
    virtual ~figura(){}
    void pomjeri(float a, float b)
    {
        cout<<"Funkcija pomjeri() osnovne klase. "<<endl;
        teziste = tacka(teziste.vratiX()+a, teziste.vratiY()+b);
    }
    // čisto virtuelna funkcija - nema realizaciju u osnovnoj klasi (fun()==0)
    // klasa koja posjeduje barem jednu čisto virtuelnu funkciju se naziva apstraktnom klasom
    virtual float obim()=0;
    virtual float površina()=0;
    virtual void citaj(){cout<<" T="; teziste.citajT();}
};

class krug : public figura
{
private:
    float poluprecnik;
public:
    krug(float r=1, tacka k=KP):figura(k), poluprecnik(r){}
    ~krug(){}
    float obim(){return 2*poluprecnik*3.14;}
    float površina(){return pow(poluprecnik,2)*3.14;}
    void citaj();
};

void krug::citaj()
{
    cout<<"Funkcija citaj() izvedene klase. ";
    cout<<"U pitanju je krug: r="<<poluprecnik;
    figura::citaj(); // zabranjeno teziste.citajT() jer je teziste privatan podatak osnovne klase
    cout<<" O="<<obim()<<" P="<<površina()<<endl;
};
```

```

class kvadrat : public figura
{
    private:
        float osnovica;
    public:
        kvadrat(float a=1, tacka t=KP):figura(t), osnovica(a){}
        ~kvadrat(){}
        float obim(){return 4*osnovica;}
        float površina(){return pow(osnovica,2);}
        void citaj();
};

void kvadrat::citaj()
{
    cout<<"Funkcija citaj() izvedene klase. ";
    cout<<"U pitanju je kvadrat a="<<osnovica;
    figura::citaj();
    cout<<" O="<<obim()<<" P="<<površina()<<endl;
};

int main()
{
    figura *pf[4];

    pf[0] = new krug;
    pf[1] = new kvadrat;
    pf[2] = new krug(2,tacka(3,3));
    pf[3] = new kvadrat(2.5,tacka(1.3,2));

    // funkcija citaj() je virtuelna => poziva se na osnovu sadržaja pokazivača pf
    // sadržaj pokazivača pf je objekat klasa krug ili kvadrat
    for(int j=0;j<4;j++)
        pf[j]->citaj();

    // funkcija pomjeri() nije virtuelna => poziva se na osnovu tipa pokazivača pf
    // pokazivača pf je tipa klase figura
    pf[0]->pomjeri(1,0.5);
    pf[1]->pomjeri(0.5,1);

    for(int j=0;j<2;j++)
        pf[j]->citaj();

    // destruktor osnovne klase mora biti virtuelna funkcija!
    // tako se pozivaju oba destruktora, i osnovne i izvedene klase
    for(int j=0;j<4;j++)
    {
        delete pf[j];
        pf[j]=0;
    }
}

```