

Zadatak 1.

Prvi deo klasa:

definicija klase se pise u fajl Tacka.h

```
#pragma once
#include <iostream>
using namespace std;
class Tacka
{
private:
    double x, y; //koordinata tacke
public:
    Tacka();      //konstruktor
    ~Tacka();     //destruktor
    //metode klase
    void Pozicija(); //metoda za ispisivanje pozicije
    void Translacija(double dx, double dy); //metoda za translaciju
};
```

definicija metoda klase se pise u fajl Tacka.cpp

```
#include "Tacka.h"
Tacka::Tacka()//konstruktor
{
    // definicija konstruktora Tacka
    x = 0; y = 0;
    cout << "Tacka je inicijalizovana!" << endl;
}
Tacka::~~Tacka()// destruktor
{
    cout << "Tacka je uklonjena!" << endl;
}
void Tacka::Pozicija()//metoda za ispisivanje pozicije
{
    cout << "x=" << x << " y=" << y << endl;
}
void Tacka::Translacija(double dx, double dy) //metoda za translaciju
{
    x += dx;
    y += dy;
}
```

Drugi deo glavni program:

Glavni program, cas1.cpp

```
//header klase Tacka
#include "Tacka.h"

int main()
{
```

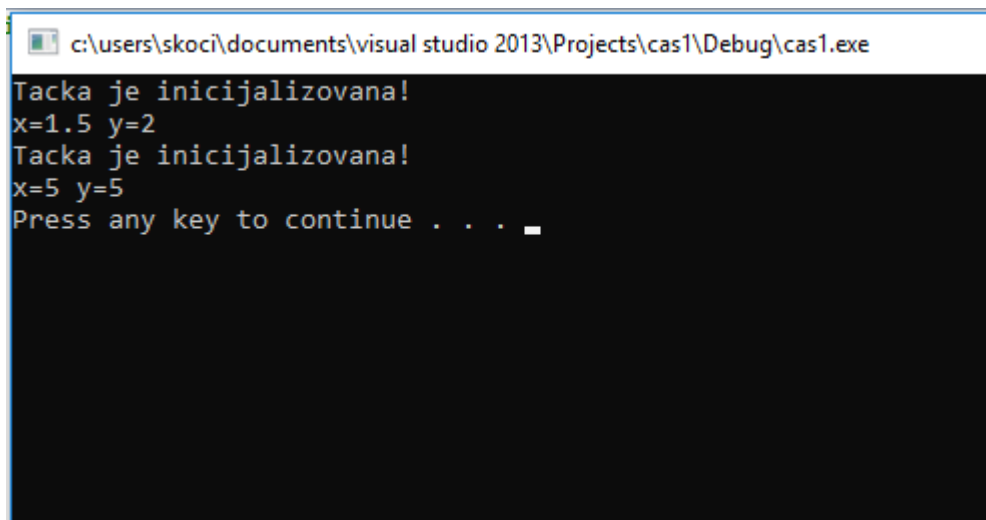
```

Tacka t1; // inicijalizacija tacke
t1.Translacija(1.5, 2.0); // pomeranje tacke
t1.Pozicija(); // ispis pozicije tacke
Tacka t2; // inicijalizacija tacke
t2.Translacija(5, 5); // pomeranje tacke
t2.Pozicija(); // ispis pozicije tacke

//po zavrsetku programa konzola se iskljucuje
//dodajte system("pause"); da vidite rezultat
system("pause");
return 0;
}

```

Izvršenje programa:



```

c:\users\skoci\documents\visual studio 2013\Projects\cas1\Debug\cas1.exe
Tacka je inicijalizovana!
x=1.5 y=2
Tacka je inicijalizovana!
x=5 y=5
Press any key to continue . . . _

```

Pitanje1: zašto se nije izvršio poziv destruktora klase?

Zadatak 2. Drugi zadatak je vezan za takozvanu vidljivost članova klase. Bice uradjen na dva načina, vi treba da uradite oba!

Prvi način:

Klasa: fajl: Tacka.h

definicija metode:

```
double Rastojanje1(Tacka t); //metoda za odredjivanje rastojanja
```

Klasa: fajl: Tacka.cpp

```

//metoda za odredjivanje rastojanja
double Tacka::Rastojanje1(Tacka t)
{
    return sqrt((x - t.x)*(x - t.x) + (y - t.y)*(y - t.y));
}

```

```
}
```

Primeti da su koordinate jedne tacke x,y; a druge t.x i t.y; pogledaj ispod kako se poziva u glavnom programu metoda rastojanje.

Glavni program: fajl: cas1.cpp

```
double d1 = t1.Rastojanje1(t2);  
cout << "Rastojanje tacaka t1 i t2 je " << d1 << endl;
```

Drugi način: upotreba prijateljskih funkcija

Klasa: fajl: Tacka.h

definicija metode:

```
friend double Rastojanje2(Tacka t1, Tacka t2); //metoda za odredjivanje rastojanja
```

Klasa: fajl: Tacka.cpp

```
//metoda za odredjivanje rastojanja  
double Rastojanje2(Tacka t1, Tacka t2)  
{  
    return sqrt((t1.x - t2.x)*(t1.x - t2.x) + (t1.y - t2.y)*(t1.y - t2.y));  
}
```

Glavni program: fajl: cas1.cpp

```
double d2 = Rastojanje2(t2,t1);  
cout << "Rastojanje tacaka t1 i t2 je " << d2 << endl;
```

Očekivano izvršavanje programa na oba načina daje isti rezultat.

```
Tacka je inicijalizovana!  
x=1.5 y=2  
Tacka je inicijalizovana!  
x=5 y=5  
Tacka je uklonjena!  
Tacka je uklonjena!  
Tacka je uklonjena!  
Prvi nacin: Rastojanje tacaka t1 i t2 je 4.60977  
Drugi nacin: Rastojanje tacaka t1 i t2 je 4.60977  
Press any key to continue . . .
```

Pitanje2: zašto se izvršio poziv destruktora klasa 3 puta?