
Elektrotehnički fakultet u Beogradu
Katedra za računarsku tehniku i informatiku

Predmet: Objektno orijentisano programiranje
(OE2OOP, OE4OOP, OF2OO1, OS2OO1, OS3OOP,
OT2OOP, OT3OOP)

Nastavnik: Prof. dr Dragan Milićev

Ispitni rok: Oktobar 2009.

Datum: 16.09.2009.

Kandidat: _____

Broj Indeksa: _____ *E-mail:* _____

*Ispit ima 2 dela ukupnog trajanja 3 sata. Na prvom delu **nije** dozvoljeno korišćenje literature. Na drugom delu **jest**e dozvoljeno korišćenje literature. Trajanje prvog dela student određuje prema ličnom nahodjenju. Student može početi da koristi literaturu kad preda prvi deo.*

UPISATI SVOJE PODATKE I NA PRVOJ STRANI DRUGOG DELA ISPITA!

Pismeni ispit:

Zadatak 1 _____/10

Zadatak 2 _____/10

Zadatak 3 _____/10

Zadatak 4 _____/10

Zadatak 5 _____/10

Rad u toku semestra:

Projekat _____/25

Domaći zadaci _____/25

Ukupno na ispitu: _____/50

Ukupno u toku semestra: _____/50

Ukupno: _____/100

Ocena: _____ (_____)

Napomena: Ukoliko u zadatku nešto nije dovoljno precizno definisano, student treba da uvede razumnu pretpostavku, da je u okviru (da bi se lakše prepoznala prilikom ocenjivanja) i da nastavi da izgrađuje preostali deo svog odgovora na temeljima uvedene pretpostavke. Ocenjivanje unutar potpitanja je po sistemu "sve ili ništa", odnosno nema parcijalnih poena. Kod pitanja koja imaju ponuđene odgovore treba **samo zaokružiti** jedan odgovor. Na ostala pitanja odgovarati **čitko, kratko i precizno**.

1. (10 poena)

Dat je deo koda na jeziku C++. Za svaku liniju označenu brojevima 1-10 navesti da li je ispravna (prevodilac neće prijaviti grešku) ili nije (prevodilac će prijaviti grešku) – u tabelu upisati „Da“ ili „Ne“.

```
class X {  
protected:  
    X (double=0.0);           // 1  
private:  
    X (int);                  // 2  
};
```

Klasa Y definisana je na sledeći način,

```
class Y : public X {  
public:  
    *  
};
```

s tim da na mestu linije označene sa * stoji jedna od sledećih linija:

```
Y () {}                       // 3  
Y () : X(1) {}                // 4  
Y (int) {}                    // 5  
Y (int i) : X(i) {}           // 6  
Y (int i) : X((double)i) {}   // 7  
Y (double) {}                 // 8  
Y (double d) : X(d) {}        // 9  
Y (double d) : X((int)d) {}   // 10
```

Odgovor:

Linija	Ispravna
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	

2. (10 poena)

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

class B {
public:
    B () : myI(0) {}
    B (int i) : myI(i) {}

    virtual void print () { cout<<"myI<<" "; }

protected:
    int myI;
};

class D : public B {
public:
    D (int i) : B(i) {}
    virtual void print () { cout<<"X "; }
};
```

Date su dve verzije funkcije main():

(a)

```
void main () {
    B array[5];
    for (int i = 0; i<5; i++) {
        array[i] = D(i);
        array[i].print();
    }
}
```

(b)

```
void main () {
    B* array[5];
    for (int i = 0; i<5; i++) {
        array[i] = new D(i);
        array[i]->print();
    }
}
```

Šta ispisuje dati program?

Odgovor:

(a) _____

(b) _____

3. (10 poena)

Na jeziku C++ realizovati klasu `Point` koja apstrahuje tačku u Dekartovom koordinatom sistemu. Tačka se predstavlja koordinatama (x,y) tipa `double`. Ova klasa treba da obezbedi sledeće:

- Inicijalizaciju koordinatama, pri čemu je podrazumevana inicijalizacija na $(0,0)$.
- Operacije poređenja na jednakost (`operator==`) i nejednakost (`operator!=`) dve tačke.
- Operaciju izračunavanja rastojanja (`operator-`) dve tačke.
- Operaciju koja vraća tačku simetričnu datoj tački u odnosu na y -osu (`operator-`).
- Operaciju koja vraća tačku simetričnu datoj tački u odnosu na x -osu (`operator!`).
- Operaciju pomeranja tačke za $(\Delta x, \Delta y)$.

4. (10 poena)

Potrebno je realizovati sledeći sistem klasa i njihovih relacija. *Građanin (Citizen)* može imati jedan ili više svojih *Ličnih Dokumentata (Personal Document)*, koji mogu biti *Pasoš (Passport)* ili *Lična Karta (ID Card)*. Svaki Lični Dokument ima svoj datum izdavanja (*date of issue*) i datum isteka važnosti (*date of expiry*).

(a) Nacrtati dijagram klasa koji prikazuje UML model opisanog sistema.

(b) Napisati kompletan C++ kod klase *Građanin (Citizen)* i svih njenih operacija, uključujući operaciju `getValidPassport()` koja vraća Pasoš datog građanina koji je još uvek važeći, ako takvog ima. Pretpostaviti da je na raspolaganju apstraktni tip podataka *Datum (Date)* sa svim potrebnim operacijama hronološkog poređenja, kao i funkcija `const Date GetToday()` koja vraća tekući datum.

5. (10 poena)

Napraviti proširenje postojećeg programskog koda TSS, tako da je moguće na kraju simulacije utvrditi koje su ulice bile najprometnije (kroz koje je prošlo najviše vozila). Napisati kod kojim treba proširiti postojeći sistem TSS da bi po završetku simulacije bilo moguće ispisati ime n najprometnijih ulica, gde je n parametar koji će biti zadat po završetku simulacije. Ne treba pisati programski kod koji obavlja opisani ispis.