
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: Januar 2010.

Datum: 30.01.2010.

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 _____/20

Domaći zadaci _____/30

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 uokviri (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)

Navesti koje simbole će prevodilac zapisati u zaglavlju proizvedenog .obj fajla kao simbole koji se izvoze, nakon prevoda sledećeg fajla:

```
extern int alfa;
extern unsigned beta;
unsigned beta;
void f(int,int);
void g ();
class X;

void h (float) {
    static int h1;
    float h2;
}

void g (void*) {
}

class Y {};
```

Odgovor: _____

2. (10 poena)

Dat je sledeći program:

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

class Device {
public:
    virtual void print () = 0;
    virtual ~Device () { delete myD; }
protected:
    Device (Device* d) : myD(d) {}
    Device* myD;
};

class Flip : public Device {
public:
    Flip (int i);
    virtual void print () { cout<<"Flip-"; if (myD) myD->print(); }
};

class Flop : public Device {
public:
    Flop (int i);
    virtual void print () { cout<<"Flop-"; if (myD) myD->print(); }
};

Flip::Flip (int i) : Device(i?new Flop(--i):0) {}
Flop::Flop (int i) : Device(i?new Flip(--i):0) {}

void main () {
    Flip* f = new Flip(4);
    f->print();
    delete f;
}
```

Šta ispisuje dati program?

Odgovor:

3. (10 poena)

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

- Inicijalizaciju koordinatama, pri čemu je podrazumevana inicijalizacija na $(0,0,0)$.
- 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 koordinatni početak (`operator-`).
- Operaciju pomeranja tačke za $(\Delta x, \Delta y, \Delta z)$.

4. (10 poena)

U nekom okruženju (engl. *framework*) za pravljenje interaktivnih aplikacija postoje apstraktni koncepti *Aplikacije (Application)* i *Dokumenta (Document)*. Kada programer pravi svoju konkretnu aplikaciju, potrebno je da iz apstraktne klase `Application` izvede svoju konkretnu klasu koja implementira konkretnu aplikaciju, koja radi sa dokumentima konkretnog tipa. Klasa `Application` realizuje algoritam otvaranja i čitanja *Dokumenta* u svojoj operaciji `openDocument`. Ovaj algoritam sprovodi sledeće korake:

- Ispituje da li se *Dokument* može otvoriti. Ovaj korak se može, ali ne mora redefinisati u izvedenoj klasi, pri čemu je podrazumevano ponašanje da se *Dokument* može otvoriti. Na ovaj način, izvedena klasa može da definiše specifične kriterijume za otvaranje *Dokumenta* (npr. u zavisnosti od svog trenutnog stanja ili nekog uslova). Ako se *Dokument* ne može otvoriti, operacija otvaranja *Dokumenta* se prekida.
- Kreira *Dokument* (kao objekat). Ovaj korak mora da implementira izvedena klasa, pri čemu ona kreira *Dokument* konkretnog tipa, specifičnog za datu aplikaciju.
- U spisak *Dokumenata* otvorenih u *Aplikaciji* dodaje kreirani *Dokument*. Ovaj korak se ne može promeniti.
- Ostavlja mogućnost izvedenoj klasi da u ovom trenutku izvrši specifične radnje prilikom otvaranja *Dokumenta*.
- Poziva operaciju `open` *Dokumenta* koji je kreiran.

(a) Nacrtati kompletan UML dijagram klasa opisanih u ovoj specifikaciji.

(b) Napisati kompletan C++ kod klase `Application`.

(c) Dati primer i napisati kod jedne jednostavne konkretne klase izvedene iz `Application`.

(d) Koji je projektni obrazac ovde primenjen?

5. (10 poena)

U nekom školskom programu, koji služi za objašnjavanje OO koncepata kroz rad sa zbirkama geometrijskih figura u ravni, podržano je nekoliko tipova figura (krug, kvadrat, trougao). Figure su određene koordinatama težišta i odgovarajućim dimenzijama (krug – poluprečnik, kvadrat – dužina stranice, trougao – dužina sve tri stranice). Program je po vrsti konzolna aplikacija interaktivnog tipa. Program može raditi sa više zbirki istovremeno. Korisnik može birati tekuću zbirku, dodavati figure u tekuću zbirku figura i uklanjati ih iz zbirke, kao i ispisivati podatke o figurama trenutno sadržanim u tekućoj zbirci figura. Korisnik može po svojoj želji za svaku od zbirki figura promeniti način ispisivanja podataka o figurama. Postoje tri načina: po neopadajućem obimu, po nerastućoj površini i po neopadajućoj udaljenosti težišta od koordinatnog početka. Potrebno je napisati samo deo programa koji služi da korisniku omogući izmenu važećeg načina ispisa. Pretpostaviti da je ostatak programa opisanog u postavci napisan i da su klase kojima su figure predstavljene opremljene metodama za određivanje površine, obima i udaljenosti težišta od koordinatnog početka. Takođe, pretpostaviti da su sve klase opremljene ispravno napisanim konstruktorima kopije, operatorima dodele i destruktorima, te da je ostatak programa napisan tako da ispravno koristi dinamičku memoriju.

- (a) Nacrtati kompletan UML dijagram klasa povezanih sa načinom ispisa zbirke figura.
- (b) Napisati kompletan C++ kod metode koja menja način ispisa zbirke figura.
- (c) Napisati kompletan C++ kod metode koja vrši ispis zbirke figura.
- (d) Koji je projektni obrazac ovde primenjen?