
Elektrotehnički fakultet u Beogradu
Katedra za računarsku tehniku i informatiku

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

Nastavnik: Prof. dr Dragan Milićev

Ispitni rok: Februar 2008.

Datum: 27.02.2008.

Kandidat: _____

Broj Indeksa: _____ *E-mail:* _____

*Ispit ima dva 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 nahođenju. Student dobija drugi deo ispita 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 _____/20

Rad u toku semestra:

Projekat _____/30

Obavezni domaći zadaci _____/10

Neobavezni domaći zadaci _____/10

Ukupno na ispitu: _____/60

Ukupno u toku semestra: _____/50

Ukupno: _____/110

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 ponudene odgovore treba **samo zaokružiti** jedan odgovor. Na ostala pitanja odgovarati **čitko, kratko i precizno**. Zadaci 1-4 su eliminatorni prema Pravilima predmeta.

1. (10 poena)

Dat je sledeći program na jeziku C++:

```
class X {
public:
    X(const char*);
    void f (char*) const;
    void g (const char*);
};

void main () {
    char* s = "Hello";
    const char* cs = s;

    X x(s); // 1
    const X cx(cs); // 2

    x.f(s); // 3
    cx.f(s); // 4

    x.f(cs); // 5
    cx.f(cs); // 6

    x.g(s); // 7
    cx.g(s); // 8

    x.g(cs); // 9
    cx.g(cs); // 10
}
```

Za svaku od linija označenih sa 1-10 navesti da li je ispravna ili nije (već će prevodilac generisati grešku). U tabelu upisati samo „Da“ ili „Ne“.

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

2. (10 poena)

Šta ispisuje sledeći program?

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

class Widget {
public:
    static Widget* create(int);
    virtual void print () = 0;
    virtual ~Widget() {};
protected:
    Widget() {};
};

class Gadget : public Widget {
protected:
    virtual void print () { cout << "Gadget\n"; }
};

class Doodad : public Widget {
protected:
    virtual void print () { cout << "Doodad\n"; }
};

Widget* Widget::create (int i) {
    if (i%2) return new Gadget();
    else return new Doodad();
}

void main () {
    for (int i=0; i<10; i++) {
        Widget* wdg = Widget::create(i);
        wdg->print();
        delete wdg;
    }
}
```

Odgovor:

3. (10 poena)

U potpunosti implementirati klasu `List` koja predstavlja jednostruko ulančanu listu čiji su elementi tipa `Object*`. Ova klasa treba da obezbedi sledeće:

- Podrazumevanu inicijalizaciju (kao prazna lista).
- Propisno uništavanje, uz dealokaciju zauzetog prostora (uključujući i sadržane objekte).
- Konstruktor kopije i operator dodele, uz kopiranje celokupnog sadržaja liste.
- Operaciju stavljanja elementa na početak liste.
- Operaciju uzimanja elementa sa početka liste (izbacuje i vraća prvi element).
- Operacije koje vraćaju prvi i poslednji element liste, bez njihovog izbacivanja iz liste.

4. (10 poena)

Potrebno je realizovati sistem klasa i njihovih relacija, opisanih tekstem u nastavku. Klasa *Korisnik* (`User`) predstavlja apstraktnu generalizaciju svih korisnika nekog programa, koji mogu biti *Pojedinačni Korisnik* (`PersonalUser`), koji predstavlja osobu, ili *Grupa Korisnika* (`UserGroup`), koja predstavlja grupu proizvoljno mnogo *Korisnika*. Jedan *Korisnik* može pripadati najviše jednoj *Grupi Korisnika*, ali ne mora pripadati nijednoj *Grupi Korisnika*.

- (a) Nacrtati dijagrame koji prikazuju UML model opisanog sistema.
- (b) Napisati C++ kod za klasu *Grupe Korisnika* (`UserGroup`) i kod za njenu operaciju `findUser(User*):bool` koja traži zadatog korisnika u sebi ili u svim svojim podređenim *Grupama Korisnika* i vraća `true` ako ga nađe, odnosno `false` ako ga ne nađe. Pretpostaviti da postoji klasa `CollectionOfUsers` koja realizuje zbirku elemenata tipa `User`, sa potrebnim interfejsom.

5. (20 poena) Konstruktivni zadatak

Napraviti proširenje postojećeg programskog koda TSS, dodavanjem klase koja simulira dvosmernu ulicu sa po jednom trakom u svakom smeru. Za određivanje vremena zadržavanja vozila u ulici koristiti generator sa eksponencijalnom raspodelom. Uzeti da je prosečno vreme zadržavanja vozila u jednom smeru 20% duže od vremena zadržavanja u suprotnom smeru. Vozilo treba da pamti svoje vreme zadržavanja u svakoj ulici kroz koju je prošlo.

- a) (10 poena) Klase **DvosmernaUlica** i **Vozilo**. Eventualne izmene u postojećim klasama.
- b) (5 poena) Dodati u postojeće **visitor** klase metode odgovarajuće klasama **DvosmernaUlica** i **Vozilo**. Obezbediti da prilikom ispisa rezultata simulacije za dvosmernu ulicu bude ispisan broj vozila za svaki od smerova, te da za svako vozilo budu ispisana vremena zadržavanja za svaku od ulica na putu datog vozila.
- c) (5 poena) Napisati programski kod koji stvara primerak klase **DvosmernaUlica** i odgovarajući broj izvora i ponora, proverava ispravnost stvorenog modela, te pokreće simulaciju i ispisuje rezultate simulacije. Na kraju, dealocirati korišćenu dinamičku memoriju.