
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 2009.

Datum: 26.01.2009.

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 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)

Date su sledeće deklaracije na jeziku C++:

```
class Base {
public:
    Base (int);
protected:
    int bi;
    void f ();
private:
    int bj;
};
```

```
class Derived : public Base {
public:
    Derived (int i) : Base (i) {          // 1
        bi = 0;                          // 2
        bj = 0;                          // 3
    }
    Derived () {}                        // 4
    static void g () { Base::f(); }     // 5
};
```

Za svaku od linija označenih sa 1-5 navesti da li je ispravna ili nije (već će prevodilac generisati grešku) – u drugu kolonu tabele upisati samo „Da“ ili „Ne“. Ukoliko je u liniji greška, navesti šta je uzrok te greške.

Linija	Ispravna?	Uzrok greške
1		
2		
3		
4		
5		

2. (10 poena)

Prokomentarišite ovaj deo programa u smislu problema pri izvršavanju.

```
class X {};  
  
X* getX () { return new X; }  
  
void f () {  
    X x = *getX();  
    delete &x;  
}
```

Odgovor:

3. (10 poena)

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

- Podrazumevanu inicijalizaciju (kao prazna lista).
- Operator dodele i konstruktor kopije.
- Propisno uništavanje, uz dealokaciju zauzetog prostora.
- Operaciju stavljanja elementa na kraj liste.
- Operaciju uzimanja elementa sa početka liste.

4. (10 poena)

Definisati klase čiji se objekti mogu povezivati u strukturu oblika stabla, uz primenu projektnog obrasca *Composite*. Listovi treba da budu objekti klase `Leaf` koji u sebi sadrže podatak član tipa `int`. Potom implementirati dva postupka obilaska takvih stabala primenom projektnog obrasca *Visitor*. Jedan postupak treba da sabira sve vrednosti podataka članova objekata tipa `Leaf`, a drugi da ih sve redom ispisuje na standardni izlaz.

- (a) Nacrtati dijagrame koji prikazuju UML model klasa opisanog sistema.
- (b) Napisati kompletan C++ kod za sve klase iz sistema.

5. (10 poena)

Potrebno je realizovati uprošćenu simulaciju plaćanja na kasi u menjačnici. Klijenti ulaze u menjačnicu, obavljaju tačno jednu transakciju i zatim odlaze iz menjačnice. Pretpostaviti da zadržavanje u menjačnici traje između 30 i 180 sekundi (ravnomerna raspodela). Po završenoj simulaciji, potrebno je ispisati sledeće rezultate: zbirni promet (za svaku valutu i otkupljen i prodat iznos) i broj klijenata koji su prošli kroz menjačnicu. Pretpostaviti da menjačnica radi samo sa RSD, EUR i USD, te da je konverzija moguća samo između RSD i EUR, odnosno RSD i USD, u bilo kom smeru, po deviznom kursu odgovarajućem za dati smer. Uzeti da je devizni kurs u toku simulacije isti za sve transakcije iste vrste. Parametre svake transakcije (strana valuta, smer konverzije, iznos koji je klijent ostavio u menjačnici) određivati pseudoslučajno za svaku transakciju. Pri rešavanju zadatka je dozvoljeno koristiti postojeći programski kod TSS uz eventualne potrebne izmene i dopune.

- a) (7 poena) Napisati klase potrebne za opisanu simulaciju. U slučaju upotrebe TSS, navesti eventualne dopune/izmene u postojećim TSS klasama.
- b) (3 poena) Napisati programski kod koji stvara i ispravno povezuje odgovarajući broj primeraka potrebnih klasa, te pokreće simulaciju i ispisuje tražene rezultate simulacije. Na kraju simulacije, osloboditi korišćenu dinamičku memoriju.