

---

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:* Oktobar 2007.

*Datum:* 26.09.2007.

*Kandidat:* \_\_\_\_\_

*Broj Indeksa:* \_\_\_\_\_ *E-mail:* \_\_\_\_\_

*Ispit traje 3 sata. Dozvoljeno je korišćenje literature.*

***Pismeni ispit:***

*Zadatak 1* \_\_\_\_\_/10

*Zadatak 2* \_\_\_\_\_/10

*Zadatak 3* \_\_\_\_\_/10

*Zadatak 4* \_\_\_\_\_/10

*Zadatak 5* \_\_\_\_\_/20

***Domaći zadaci:***

*Obavezni deo* \_\_\_\_\_/40

*Neobavezni deo* \_\_\_\_\_/10

***Ukupno na ispitu:*** \_\_\_\_\_/60

***Ukupno na domaćem:*** \_\_\_\_\_/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 ponuđene 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 deo koda na jeziku C++. Za svaku liniju označenu brojevima 1-11 navesti da li je ispravna (prevodilac neće prijaviti grešku) ili nije (prevodilac će prijaviti grešku) – u tabelu upisati „Da“ ili „Ne“.

```
int i=0, j=0, a[10];
int* p = a; // 1
int* q = &a; // 2
int* r = i; // 3
int* s = &i; // 4
int& t = j; // 5
int& u = &j; // 6
j = *(p-q); // 7
i = *(p+1); // 8
j = *p+1; // 9
*(p+1) = i; // 10
*p+1 = j; // 11
```

Odgovor:

Linija	Ispravna
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	

## 2. (10 poena)

Program na jeziku C++ se sastoji iz dva fajla, `main.cpp` i `f.cpp`. Sadržaj `main.cpp` je sledeći:

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

void f(char*,int);

const int N=10;
static char s[N];
static char a[N];

void main () {
    int i;
    for (i=0; i<N; i++) a[i]='0'+i;
    for (i=0; i<N; i++) cout<<a[i]; cout<<'\n';
    f(s,N);
    for (i=0; i<N; i++) cout<<a[i]; cout<<'\n';
}
```

Funkcija `f` definisana je u fajlu `f.cpp`. Program se prevodi bez grešaka i upozorenja. Kada se pokrene, ovaj program se regularno završi ostavljajući sledeći ispis:

```
0123456789
!123456789
```

Kakav je vaš komentar o ispravnosti ovog programa? Precizno obrazložiti.

### 3. (10 poena)

Na jeziku C++ realizovati klasu `Directory` koja služi kao jednostavni „rečnik“. Rečnik skladišti listu parova nizova znakova ključ-vrednost i obezbeđuje pretraživanje po ključu. Ova klasa treba da obezbedi sledeće:

- Inicijalizaciju. Rečnik je inicijalno prazan.
- Smeštanje novog para ključ-vrednost u rečnik.
- Vraćanje vrednosti za zadati ključ; ako ključ nije u rečniku, treba vratiti specijalnu vrednost koja se može razlikovati od vrednosti.
- Ispisivanje sadržaja rečnika kao liste parova ključ-vrednost na izlazni tok (`ostream`) dat kao argument.
- Brisanje sadržaja rečnika.

Mogu se koristiti proizvoljne gotove realizacije potrebnih struktura podataka (na primer neka od kolekcija – ne treba je realizovati; treba samo precizno navesti koja se kolekcija koristi i šta se očekuje da rade korišćene metode te kolekcije).

#### 4. (10 poena)

Projektuje se softver za neku elektronsku trgovinu. *Proizvod* ima svoj naziv (niz znakova) i opis (niz znakova). *Cenovnik* definiše cene za određene proizvode koje važe počev od datuma cenovnika. Tokom vremena se može kreirati proizvoljno mnogo cenovnika. Svaki cenovnik definiše cene za određene, ali ne obavezno sve proizvode. Ako neki cenovnik sadrži cenu za neki proizvod, onda on definiše cenu koja važi za taj proizvod počev od datuma cenovnika; u suprotnom, za proizvod važi cena iz prethodnog cenovnika, odnosno podrazumevana cena proizvoda ako uopšte ne postoji cenovnik koji definiše cenu tog proizvoda. Potrebno je predvideti naknadnu ugradnju novih vrsta proizvoda kojima se cene određuju na sasvim drugačiji način.

Predložiti UML strukturni model (model klasa) opisanog sistema. Na jeziku C++ napisati definicije klasa iz ovog modela i realizovati operaciju koja vraća aktuelnu cenu datog proizvoda. Mogu se koristiti proizvoljne gotove realizacije potrebnih struktura podataka (npr. kolekcija) – pretpostaviti da su na raspolaganju (ne treba ih realizovati).

## 5. (20 poena) Konstruktivni zadatak

Napraviti proširenje postojećeg programskog koda TSS, dodavanjem klase koja simulira dvosmernu ulicu sa po dve trake u svakom smeru. Pri ulasku u ulicu, vozilo ulazi u onu traku u kojoj u trenutku ulaska ima manje drugih vozila. Uzeti da je zadržavanje vozila u jednoj od traka istog smera 40% duže od zadržavanja u drugoj traci, a da su zadržavanja vozila u odgovarajućim trakama u različitim smerovima ista.

- a) (10 poena) Klasa **DvosmernaUlica** i potrebne izmene u postojećim klasama.
- b) (5 poena) Dodati u postojeće **visitor** klase metode odgovarajuće klasi **DvosmernaUlica**.
- c) (5 poena) Napisati programski kod koji stvara i ispravno povezuje 2 izvora sa 1 dvosmernom ulicom i potrebnim brojem ponora, te pokreće simulaciju i ispisuje rezultate simulacije (za ulicu obavezno ispisati broj ukupan vozila u svakoj od traka). Na kraju, dealocirati korišćenu dinamičku memoriju.