

---

---

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:* Septembar 2007.

*Datum:* 03.09.2007.

*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 1-10 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)

Dat je sledeći program na jeziku C++:

```
#include <string.h>
#include <iostream.h>

class X {
public:
    X (const char*);
    void setChar(int index, char);
    const char* getStr () const { return s; }
private:
    char* s;
};

X::X (const char* str) {
    if (s = new char[strlen(str)+1]) strcpy(s,str);
}

void X::setChar (int i, char c) {
    if (0<=i && i<strlen(s)) s[i]=c;
}

void main () {
    X x1("X1");
    X x2(x1);

    cout<<x1.getStr()<<'\n';
    cout<<x2.getStr()<<'\n';

    x2.setChar(1,'3');

    cout<<x1.getStr()<<'\n';
    cout<<x2.getStr()<<'\n';

}
```

Pored svake linije koda sa ispisom na `cout`, na liniji sa desne strane ispisati šta ispisuje ta linija.

### 3. (10 poena)

Na jeziku C++ realizovati klasu `clock` koja apstrahuje vremenski brojač koji se može pokrenuti da meri interval vremena. Ova klasa treba da obezbedi sledeće:

- Inicijalizaciju zadatim vremenskim intervalom koji treba da meri (interval je predstavljen kao ceo broj vremenskih jedinica).
- Pokretanje merenja vremena.
- Zaustavljanje merenja vremena.
- Kada istekne zadati interval vremena, brojač treba da pozove polimorfnu operaciju `timeout()` (indirektne) instance apstraktne klase `Timeout` na koju ukazuje pokazivač dostavljen kao opcioni argument konstruktora brojača.
- Operaciju koja vraća vreme koje je preostalo do kraja intervala koji se meri.

Protok jedne jedinice vremena signalizira se spolja pozivom operacije `tick()` ove klase.

#### **4. (10 poena)**

Projektuje se softver za arhiviranje video zapisa koje korisnik želi da uključi u svoj „album“. Svaki video zapis praćen je sledećim svojstvima: datum snimanja, mesto snimanja, ime snimatelja, tema (kratak tekstualni opis), detaljan opis (tekst). Da bi se korisniku omogućilo praktičnije manipulisanje snimljenim zapisima, oni se organizuju u hijerarhijsku strukturu „kataloga“ poput fajlova i foldera na standardnom fajl sistemu (kao što folder može da sadrži fajlove i druge foldere, tako katalog može da sadrži video zapise i druge kataloge). Katalog je opisan istim svojstvima kao i pojedinačni video zapis. Potrebno je obezbediti i operacije premeštanja, kopiranja i brisanja jednog zapisa ili kataloga iz njegovog roditeljskog kataloga. Predložiti UML strukturni model (model klasa) opisanog sistema. Ukratko objasniti predloženi model i način implementacije navedenih operacija. Nije potrebno pisati programski kod navedenih klasa i operacija.

## 5. (20 poena) Konstruktivni zadatak

Radi optimalne organizacije javnog skupa, potrebno je napraviti uprošćenu simulaciju potrošnje piva. Pri realizaciji sistema, poči od sledećih pretpostavki:

- Šanker je sve vreme koncerta zaposlen i konstantno toči pivo, koje raspoređuje na  $N$  mesta na šanku. Za točenje svakog piva potrebno je  $T (\pm 10\%)$  sekundi (ravnomerna raspodela). Mesta se opslužuju kružnim redosledom, počev od prvog.
- Na svakom od mesta se sigurno nalazi posetilac javnog skupa, koji konzumira pivo, koje se zadržava u organizmu posetioca  $Z (\pm 30\%)$  minuta, pre nego posetilac ode do sanitarnog čvora i izluči pivo. Ako posetilac u nekom trenutku u organizmu ima više od  $M$  piva, odlazi do sanitarnog čvora i izbacuje sva piva koja ima u organizmu.
- Zadržavanje posetioca u sanitarnom čvoru je isto za sve posetioce. Kapacitet sanitarnog čvora je neograničen.

$N$  i  $T$ , odnosno  $Z$  i  $M$ , ne moraju biti isti za sve šankere, odnosno posetioce. Nije potrebno projektovati klase koje će podržati opisano ponašanje šankera, posetioca i sanitarnog čvora. Pretpostaviti da ove klase već postoje, da su izvedene iz odgovarajućih klasa TSS i da imaju implementirane/redefinisane odgovarajuće metode za opisano ponašanje. Potrebno je napisati programski kod koji stvara i povezuje opisane objekte, te ispisuje rezultate simulacije. Pri ovome, može se koristiti i/ili menjati postojeći programski kod TSS. Naglasiti koje se metode iz koje klase koriste (važi i za one koje postoje u TSS i za one koje opisane klase treba da imaju na raspolaganju).

- a) (7 poena) Napisati programski kod koji stvara i povezuje jednog šankera, broj posetilaca koji se zadaje sa standardnog ulaza i jedan sanitarni čvor. Broj posetilaca nije isti kao broj mesta na šanku. Redosled dolaska posetilaca na šank je slučajan.
- b) (5 poena) Na početku i na kraju simulacije, ispisati svojstva svih simuliranih objekata (npr. za šankera:  $N$ ,  $T$  i ID svih posetilaca koji su trenutno na šanku).
- c) (5 poena) Na kraju simulacije, ispisati broj natočenih piva za svakog šankera, broj konzumiranih piva za svakog posetioca i broj svih poseta sanitarnom čvoru.
- d) (3 poena) Napisati programski kod koji uništava sve stvorene objekte.