
Elektrotehnički fakultet u Beogradu
Katedra za računarsku tehniku i informatiku

Predmet: Objektno orijentisano programiranje
(OE2OOP, OE4OOP, OF2OOP, OS2OOP, OS3OOP,
OT2OOP, OT3OOP)

Nastavnik: Prof. dr Dragan Milićev

Ispitni rok: Jun 2010.

Datum: 26.06.2010.

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 nađenju. 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 u okviru (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)

Data je sledeća funkcija:

```
int strcmp (char* str1, char* str2) {
    while (1) {
        if ((*str1!=*str2) || (!*str1)) return *str1-*str2;
        str1++; str2++;
    }
    return 0;
}
```

Prepraviti ovaj kod tako da data funkcija bude:

- upotrebljiva i za konstantne stringove;
- otporna na nula-pokazivače;
- otporna na nizove znakova koji nisu ograničeni terminalnim znakom '\0'.

2. (10 poena)

Sledeći program se ispravno prevodi, ali ima jednu logičku grešku pri izvršavanju.

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

class Widget {
public:
    static Widget* create(int);
    virtual ~Widget() { cout << "Widget\n"; }
    Widget() {}
};

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

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

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

int main () {
    const size_t WIDGET_COUNT = 5;
    Widget *magicBox[WIDGET_COUNT];
    for (int i=0; i<WIDGET_COUNT; i++)
        magicBox[i] = Widget::create(i);
    return 0;
}
```

- (a) Šta ispisuje dati program?
- (b) Prokomentarišite ovaj program u smislu iznad pomenute greške. Šta treba promeniti u ovom programu i/ili dodati u ovaj program da bi greška bila uklonjena?
- (c) Po uklanjanju greške, šta ispisuje popravljani program?

3. (10 poena)

Na jeziku C++ implementirati potrebne klase koje realizuju jednostruko ulančanu listu. Svaki element liste kao podatak sadrži jedan celi broj. Treba obezbediti sledeće operacije:

- Podrazumevanu inicijalizaciju liste (vrednost svih pokazivača je 0).
- Operaciju `front()`, koja vraća pokazivač na prvi element liste.
- Operaciju `back()`, koja vraća pokazivač na poslednji element liste.
- Operaciju `push_front()`, koja dodaje element na početak liste.
- Operaciju `push_back()`, koja dodaje element na kraj liste.
- Propisnu dealokaciju prilikom uništavanja.
- Konstruktor kopije i operator dodele koji vrše duboko kopiranje cele liste.

4. (10 poena)

Potrebno je realizovati sistem klasa i njihovih relacija, opisanih tekstem u nastavku. Jedinstveno *Preduzeće*, koje se bavi pružanjem *Usluga* u oblasti nege tela, može imati više *Salona*. U svakom od salona radi jedan ili više *Terapeuta*. Jedan terapeut može raditi samo u jednom salonu. Svaki salon vodi evidenciju o zaposlenim terapeutima, kao i o tome koje su sve usluge terapeuti pružili korisnicima. Na osnovu ove evidencije, preduzeće može da pravi razne izveštaje: o prometu svakog salona, o popularnosti određenih usluga i o učinku svakog od terapeuta. Usluge koje saloni pružaju su *Vežbanje*, *Masaža* i *Kozmetički tretman*. Cena i naziv neke usluge su isti u svakom od salona u sastavu preduzeća.

- (a) Nacrtati dijagrame koji prikazuju UML model klasa opisanog sistema, sa navođenjem naziva metoda pomenutih u opisu. Objasniti koji je DP uočljiv u postavci zadatka.
- (b) Napisati kompletan C++ kod za klasu *Usluga*.
- (c) Napisati metodu `float Salon::promet()`, koja vraća ukupan promet ostvaren u datom salonu. Metode koje ova metoda koristi nije potrebno realizovati, ali je za svaku od njih potrebno navesti deklaraciju i opis funkcionalnosti u jednoj ili dve rečenice.

5. (10 poena)

U nekom školskom programu, koji služi za objašnjavanje OO koncepata kroz rad sa zbirka 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). Pored opisanih, postoji i složena figura, koja u sebi može sadržati i proste i složene figure. Površina i obim složene figure su određeni kao zbir odgovarajuće osobine sadržanih figura. Koordinate težišta su određene kao srednja vrednost odgovarajućih koordinata sadržanih figura. 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. Potrebno je napisati samo deo programa koji služi za dodavanje figure u tekuću zbirku i uklanjanje figure iz tekuće zbirke. 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 svih klasa koje predstavljaju opisane figure.
- (b) Napisati kompletan C++ kod za izračunavanje površine složene figure.
- (c) Napisati kompletan C++ kod za uklanjanje date figure iz tekuće zbirke figura.
- (d) Koji je projektni obrazac ovde primenjen?