
Elektrotehnički fakultet u Beogradu
Katedra za računarsku tehniku i informatiku

Predmet: Testiranje Softvera (SI3TS)
Nastavnik: doc. dr Dragan Bojić
Asistent: dipl. ing. Dražen Drašković
Ispitni rok: Januar 2011.
Datum: 15.01.2011.

Kandidat:* _____

Broj Indeksa:* _____

*Ispit traje 3 sata, prvih sat vremena nije dozvoljeno napuštanje ispita.
Upotreba literature nije dozvoljena.*

<i>Zadatak 1</i>	_____ /5	<i>Zadatak 5</i>	_____ /11
<i>Zadatak 2</i>	_____ /5	<i>Zadatak 6</i>	_____ /10
<i>Zadatak 3</i>	_____ /12	<i>Zadatak 7</i>	_____ /10
<i>Zadatak 4</i>	_____ /7		

Ukupno na ispitu: _____ /60 *Ukupno na domaćem*:* _____ /40

Rok u kome je odbranjen domaći:* _____ (primer: januar 2011)

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. Kod pitanja koja imaju ponuđene odgovore treba **samo zaokružiti** jedan odgovor. Na ostala pitanja odgovarati **čitko, kratko i precizno.** * popunjava student.

1. [5]
 - a) Kriterijumi za određivanje kritičnih modula u integracionom testiranju.
 - b) Kako treba postupati sa kritičnim modulima?

2. [5]
 - a) Šta je cilj mutacionog testiranja?
 - b) Kada se za mutanta kaže da je mrtav?

3. [12] Na skimpozijumu na Kopaoniku učestvuju:
- studenti, koji imaju rad na konferenciji, položili su sve ispite u semestru i znaju da skijaju, učestvuju besplatno;
 - studentkinje, koje imaju visok prosek, položile su sve ispite i hoće da nauče da skijaju, takođe učestvuju besplatno;
 - studentkinje i studenti, koji nemaju visok prosek, a položili su sve ispite, plaćaju polovinu cene kotizacije;
 - studentkinje i studenti koji znaju da skijaju, a nisu položili sve ispite, plaćaju punu cenu kotizacije.

Oni koji imaju visok prosek, položili su sve ispite u semestru.

Nacrtati uzročno-posledični graf, prikazati sva ograničenja među uzrocima i kreirati test primere za opisani sistem. Izvesti tabelu odlučivanja koristeći tehniku senzitivizacije putanja (Basic Path Sensitization Technique).

4. [7] Jednostavni program za vremensku prognozu prihvata tri faktora:
- trenutnu temperaturu, koja se zadaje u jednom od tri opsega: manje od 0, od 0 do 30 i preko 30;
 - oblačnost, sa jednom od tri vrednosti: oblačno, umereno oblačno i vedro;
 - vetrovitost: duva košava ili nema vetra.

IPO strategijom potrebno je izgenerisati pokrivajući niz sa mešovitim nivoima, za tri navedena faktora.

5. [11] Dat je SelectionSort algoritam koji sortira niz od n elemenata, indeksiranih od 0 do n-1:

```
int iPos;
int iMin;
for (iPos = 0; iPos < n; iPos++){
    iMin = iPos;
    for (i = iPos+1; i < n; i++){
        if (a[i] < a[iMin]){
            iMin = i;
        }
    }
    swap(a, iPos, iMin);
}
```

Testirati sve petlje u ovoj proceduri i detaljno obrazložiti.

6. [10] Razvijena su četiri modula, označena sa modul1 do modul4.
- Predložiti pristup za integraciono testiranje i obrazložiti. Naznačiti kako bi izgledali dražveri/stabovi u svakom koraku postupka integracije.
 - Kako bi izgledala mešovita integracija (*Sandwich testing*)?
- Napomena: Podrazumevati da za svaki interfejs postoji odgovarajuća implementacija.

```
MODUL 1
public class Sortiranje {

    public static void sort(KolekcijaElemenata elementi,
                           Uredjenje uredjenje,
                           Zamena zamena) {
        for (int i = 0; i < elementi.brojElemenata() - 1; i++)
            for (int j = i + 1; j < elementi.brojElemenata(); j++) {
                Elem e1 = elementi.dohvatiElem(i);
                Elem e2 = elementi.dohvatiElem(j);
                if (uredjenje.ispred(e1, e2)) {
                    zamena.zameni(e1, e2);
                }
            }
    }
}
```

```
MODUL 2
public interface Zamena {
    public void zameni(Elem e1, Elem e2);
}
```

```
MODUL 3
public interface KolekcijaElemenata {
    public Elem dohvatiElem(int index);
    public int brojElemenata();
}

public interface Elem {
}
```

```
MODUL 4
public interface Uredjenje {
    public boolean ispred(Elem e1, Elem e2);
}
```

7. [10] Za sledeći program, strategijom svih DU lanaca, odrediti test primere.
Napomena: izostaviti iz razmatranja promenljivu broj_domacih.

```
package exams;

public class Ocena {
    private static int EXTRA = 5;

    public static void main(String args[]) {
1       int broj_domacih = args.length - 1;
2       int domaci = 0, ukupno = 0;

3       while (domaci < broj_domacih) {
4           int poeni = Integer.parseInt(args[domaci]);
5           if (poeni < 0)
6               System.out.println("Nekorektan unos");
7           else {
8               ukupno += poeni;
9               domaci++;
10          }
11     }

12     int aktivnost = Integer.parseInt(args[broj_domacih]);
13     if (aktivnost == 1)
14         ukupno += EXTRA;

15     System.out.println("Ukupno poena " + ukupno);

16     if (ukupno > 90)
17         System.out.println("Ocena 10");
18 }
19}
```

Na osnovu odrađenih testova, potrebno je implementirati skup JUnit testova (najmanje 3).
Definisati potrebne izmene nad programom kako bi testiranje bilo moguće.