
Elektrotehnički fakultet u Beogradu
Katedra za računarsku tehniku i informatiku

Predmet: Testiranje Softvera (SI3TS)
Nastavnik: vanredni prof. dr Dragan Bojić
Asistent: dipl. ing. Dražen Drašković
Ispitni rok: Drugi kolokvijum (novembar 2015.)
Datum: 24.11.2015.

Kandidat:* _____

Broj indeksa:* _____ *E-mail*:* _____

*Kolokvijum traje 100 minuta, a u prvih sat vremena nije dozvoljeno napuštanje kolokvijuma.
Upotreba literature nije dozvoljena.*

Zadatak 1 _____/4
Zadatak 2 _____/6
Zadatak 3 _____/4
Zadatak 4 _____/6

Ukupno na kolokvijumu: _____/20

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

* popunjava student.

1. [4] Napomena za tačke a), b, c): ako je odgovor potvrđan, dati kratko obrazloženje, ako je odgovor odričan, navesti kontra-primer (dati što jednostavniji primer programskog koda, seriju testova koji zadovoljavaju jedan kriterijum, a ne zadovoljavaju drugi).

- a) Da li pokrivanje LCSAJ garantuje pokrivanje odluka?
- b) Da li pokrivanje LCSAJ garantuje pokrivanje elementarnih uslova?
- c) Da li minimalno pokrivanje višestrukih uslova garantuje pokrivanje odluka?
- d) Objasniti razliku između pokrivanja višestrukih uslova i minimalnog pokrivanja višestrukih uslova na primeru uslovnog izraza:

$((x < 5) \parallel ((y / z) == 2)) \&\& m.isSelected()$.

Navesti konkretno šta različiti testovi treba da pokriju u jednom slučaju, a šta u drugom (nije potrebno smišljati konkretne ulazne vrednosti).

Odgovor:

2. [6] Potrebno je realizovati jedinično testiranje java funkcije m() klase M za jedan test primer. Napisati:

- a) Odgovarajući drajver.
- b) Odgovarajuće dublere (stub-ove).

Napomena: U odgovoru dati čist Java kod, bez korišćenja JUnit alata i mock biblioteka.

```
class M {
    public static void main(String [] argv){
        M obj = new M();
        if (argv.length > 0)
            obj.m(argv[0], argv.length);
    }
    public void m(String arg, int i) {
        int q = 1;
        A o = null;
        Impossible nothing = new Impossible();
        if (i == 0)
            q = 4;
        q++;
        switch (arg.length()) {
            case 0: q /= 2; break;
            case 1: o = new A(); q = 25; break;
            case 2: o = new A(); q = q * 100;
                // no break
            default: o = new B(); break;
        }
        if (arg.length() > 0) {
            o.m();
        } else {
            System.out.println(q);
        }
        nothing.happened();
    }
}
```

Odgovor:

3. [4] a) Odrediti sve c-upotrebe, p-upotrebe i sve DU lance za datu Javinu metodu, po svim promenljivama. Zatim odrediti minimalan skup testova koji pokrivaju sve DU lance.

```
1. public int getDecimalFromBinary(int binary){
2.     int decimal = 0;
3.     int power = 0;
4.     while(true){
5.         if(binary == 0){
6.             break;
7.         } else {
8.             int tmp = binary%10;
9.             decimal += tmp*Math.pow(2, power);
10.            binary = binary/10;
11.            power++;
12.        }
13.    }
14.    return decimal;
15. }
```

Odgovor:

- b) Odrediti sve DU lance za datu Javinu metodu, po svim promenljivama. Zatim odrediti minimalan skup testova koji pokrivaju sve DU lance. Da li se ovim testovima otkriva neka greška u programu?

```
1. public static String decimal2hex(int d) {
2.     String digits = "0123456789ABCDEF";
3.     String hex = "";
4.     while (d > 0) {
5.         int digit = d % 16;
6.         hex = digits.charAt(digit) + hex;
7.         d = d / 16;
8.     }
9.     return hex;
10. }
```

Odgovor:

4. [6] Dva softverska inženjera napisali su programe za sortiranje celobrojnih nizova. Prvi inženjer uradio je implementaciju svog algoritma u programskom jeziku JavaScript, a drugi inženjer uradio je implementaciju svog algoritma u programskom jeziku Java. Vi ste treći inženjer u kompaniji koji treba da testira oba programa, tako što pozove napisane funkcije sa traženim argumentima.

PRVI INŽENJER:

```
var countSort = function(arr, min, max) {
  var i, z = 0, count = [];
  for (i = min; i <= max; i++) {
    count[i] = 0;
  }
  for (i=0; i < arr.length; i++) {
    count[arr[i]]++;
  }

  for (i = min; i <= max; i++) {
    while (count[i]-- > 0) {
      arr[z++] = i;
    }
  }
}
```

DRUGI INŽENJER:

```
public static int[] sort(int[] old) {
  for (int shift = Integer.SIZE - 1; shift > -1; shift--){
    int[] tmp = new int[old.length];
    int j = 0;

    for (int i = 0; i < old.length; i++) {
      boolean move = old[i] << shift >= 0;

      if (shift == 0 ? !move : move) {
        tmp[j] = old[i];
        j++;
      } else {
        old[i - j] = old[i];
      }
    }

    for (int i = j; i < tmp.length; i++) {
      tmp[i] = old[i - j];
    }
    old = tmp;
  }
  return old;
}
```

a) Napisati test primere koji uspešno testiraju petlje u oba programa. Za svaki test potrebno je definisati dužinu niza celih brojeva i sve elemente niza (a u prvom programu i granice min i max).

b) Ukoliko je prvi programer pogrešio i u while petlji prvog programa zamenio:

- postdekrement (`count[i]--`) sa predekrementom (`--count[i]`), a zatim sa postinkrementom (`count[i]++`)
- postinkrement (`arr[z++]`) sa preinkrementom (`arr[++z]`), a zatim sa postdekrementom (`arr[z--]`)

dokazati da svaka od ove četiri izmene u programu za definisane test primere (iz tačke a)) ne utiče na izvršavanje programa tako da radi ispravno (ne sortira niz) ili dokazati suprotno - da postoji takav test primer za koji neka od ovih izmena ne utiče na konačni izlaz programa.

Odgovor: