
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 (decembar 2016.)
Datum: 06.12.2016.

Kandidat:* _____

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

*Kolokvijum traje 120 minuta, a u prvih sat vremena nije dozvoljeno napuštanje kolokvijuma.
Upotreba literature nije dozvoljena.*

Zadatak 1 _____/5
Zadatak 2 _____/5
Zadatak 3 _____/6
Zadatak 4 _____/4

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. [5] U programskom jeziku C++ promenljiva se deklarise (kada se uđe u blok {List* l;...} gde je njena deklaracija), dodeljuje joj se vrednost (npr. l = new List();), koristi se (npr. x = l->getValue();), briše joj se vrednost (delete l;) i uklanja se kada se izade iz deklaracionog opsega.

a) Prikazati dijagram stanja C++ promenljive, ako se pretpostavi da imamo ukupno tri stanja: “nedeklarisana”, “deklarisana a nema vrednost” i “ima vrednost” i četiri događaja: d – deklarisanje, v – dodela vrednosti, u – upotreba i r – uklanjanje deklaracije. Nelegalne tranzicije (koje odgovaraju programerskim greškama) ne prikazivati.

b) Prikazati tabelarno dijagram stanja iz tačke a) sa SVIM tranzicijama.

c) Napraviti jedan ili više test primera programa na C++ koji pokrivaju nelegalne tranzicije. Za svaku tranziciju uvesti posebno ime promenljive iz koga se vidi na koju tranziciju se promenljiva odnosi.

2. [5] Za sledeći fragment programa prikazati minimalne skupove testova koji pokrivaju:
- a) sve odluke
 - b) sve elementarne uslove
 - c) sve minimalne višestruke uslove
 - d) sve putanje

Navesti za svaki test iz svakog skupa koju tačno (odluku, elementarni uslov, itd.) pokriva:

```
IF ((A > 1) AND (B == 0)) THEN  
  X = X / A  
IF ((A == 2) OR (X > 1)) THEN  
  X = X + 1
```

Odgovori:

3. [6] Realizovan je program u Javi koji sa standardnog ulaza od korisnika prihvata jedan ceo broj n i poziva metodu `factors(int n)` sa tim unetim argumentom. Pored void metode `factors`, postoji pomoćna boolean metoda `pr(int n)` i obe implementacije su date u sledećim listinzima:

```
1 public static boolean pr(int n) {
2     if (n == 1) {
3         return false;
4     }
5     if (n == 2) {
6         return true;
7     }
8     int i = 2, s = (int) Math.sqrt(n);
9
10    while ((i <= s) && (n % i != 0)) {
11        i++;
12    }
13    if (n % i != 0) {
14        return true;
15    } else {
16        return false;
17    }
18 }
19
20 public static void factors(int n) {
21     if (pr(n)) {
22         System.out.println("(" + n + ")");
23     } else {
24         int i = 2;
25         int check = 0;
26         while (n > 1) {
27             if (pr(i)) {
28                 while (n % i == 0) {
29                     n /= i;
30                     check++;
31                 }
32                 if (check == 1) {
33                     System.out.print("(" + i + ")");
34                     check = 0;
35                     if (n > 1) {
36                         System.out.print(" *");
37                     }
38                 } else if (check != 0) {
39                     System.out.print("(" + i + " * " + check + ")");
40                     check = 0;
41                     if (n > 1) {
42                         System.out.print(" *");
43                     }
44                 }
45             }
46             i++;
47         }
48         System.out.println();
49     }
50 }
```

a) Testirati WHILE petlje u metodama *factors* i *pr*, prema pravilima o testiranju petlji i odrediti minimalni skup test primera potreban da se petlje istestiraju prema pravilima. Za svaki test primer napisati šta je ulazni podatak i šta je očekivani izlazni podatak (za veće cele brojeve na ulazu, nije potrebno pisati izlaz).

Odgovor:

b) Odrediti sve definicije, c- i p- upotrebe za navedene promenljive iz metode *factors* i DU lance samo za promenljivu *check* (napomena: smatrati da se promenljiva n prvi put inicijalizuje u liniji #20):

Promenljiva	DEF	C- USE	P- USE	DU lanci
n				/
i				/
check				

4. [4] Odrediti sve LCSAJ sekvence za datu metodu *factors* iz zadatka 3. Svaku sekvencu predstaviti kao uređenu trojku:
[POČETAK_SEKVENCE, KRAJ_SEKVENCE, LINIJA_NA_KOJU_SKAČEMO].

Odgovor:

Da li minimalni skup testova iz tačke a) pokriva sve LCSAJ iz tačke c) ili je moguće redukovati broj test primera? Da li su pokrivanjem svih sekvenci LCSAJ pokriveni i sve odluke?

Odgovor: