
Elektrotehnički fakultet u Beogradu
Katedra za računarsku tehniku i informatiku

Predmet: Testiranje softvera (SI3TS / 13S113TS)

Nastavnik: Prof. dr Dragan Bojić

Asistent: dipl. ing. Dražen Drašković

Ispitni rok: Februar 2018.

Datum: 09.02.2018.

Kandidat:* _____

Broj indeksa:* _____

*Ispit traje 2.5 sata, a u toku prvog sata nije dozvoljeno napuštanje ispita.
Upotreba literature nije dozvoljena.*

<i>Zadatak 1</i>	_____ /6	<i>Zadatak 4</i>	_____ /12
<i>Zadatak 2</i>	_____ /10	<i>Zadatak 5</i>	_____ /12
<i>Zadatak 3</i>	_____ /12	<i>Zadatak 6</i>	_____ /8

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

Rok u kome je odbranjen domaći:* _____ (primer: januar 2018)

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. [6] Ilustrovati konstrukciju kombinatorne tabele po IPO proceduri za četiri faktora A, B, C i D od kojih A i B imaju po dve vrednosti 0 i 1, a C i D imaju po tri vrednosti 0, 1 i 2. Potrebno je pokriti sve parove vrednosti.

2. [10] Sledeća BNF gramatika opisuje naprednu pretragu na nekom veb sajtu (početno pravilo je search):

```
search = search binop term | "not" search | term.  
binop = "and" | "or".  
term = regexp | "(" search ")".  
regexp = char regexp | char | "{" choices "}" | "*".  
choices = regexp | regexp "," choices.  
char = "x" | "y".
```

a) Prema pravilima testiranja sintakse, odrediti validne test primere.

b) Ukoliko se usvoje mutacioni operatori m1: izostavljanja obaveznog sintaksnog elementa i m2: uvođenja nedozvoljene vrednosti za sintakсни element, prikazati test primere koji se dobijaju sistematičnom primenom ovog operatora na pravila za *regexp* i *choices* (treba naravno uzeti u obzir i ostala pravila ali na njih ne treba primenjivati operatore).

3. [12] Aranžmani turističke agencije mogu biti: (1) samo hotel i (2) paket aranžman (hotel+avion). Ukoliko se turista odluči da ide samostalno i izabere aranžman (1), on plaća 300 evra, a ako izabere aranžman (2) plaća 500 evra. Ukoliko se odluči da ide sa suprugom, aranžman (1) plaća ukupno 500 evra, a aranžman (2) 700 evra. Turista može povesti i decu, ali samo ukoliko idu u pratnji oba roditelja i ukoliko nije odabran aranžman koji uključuje samo hotel (aranžman #1). Iznos paketa sa decom bio bi 800 evra. Nacrtati uzročno-posledični graf aplikacije koja kao izlaz prikazuje ukupan iznos cene aranžmana. Nakon toga, izvesti tabelu odlučivanja primenom tehnike senzitivizacije putanja (eng. *Basic Path Sensitization Technique*) i napisati konačan skup test primera posle primene te tehnike testiranja.

4. [12] 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) 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				

b) Testirati metodu factors() i navesti minimalni broj test primera ako se pokrivaju:

- (1) sve odluke;
- (2) svi višestruki uslovi;

Napomena: Za svaki test navesti putanje koje test pokriva i koja odluka/višestruki uslov je pokriven/a.

Odgovor:

c) Šta označavaju tagovi @Before i @Ignore u sledećem JUnit testu. Objasniti ukratko šta se dešava u sledećem programskom kodu.

```
public class A {
    @Before public void runB() {
        simpleMath = new SimpleMath();
    }
    @Ignore("Neki komentar")

    @Test public void multiplication() {
        assertEquals(15, simpleMath.multiply(3,5));
    }
}
```

Odgovor:

5. [12] Testira se grafički-korisnički interfejs (GUI) sledeće aplikacije za glasanje na takmičenju Evrovizija. Voditeljka bira zemlju iz prve padajuće liste i nakon izbora zemlje automatski se popunjava grad u drugoj padajućoj listi. Važi i obrnuto (ako se odabere grad, automatski se popunjava zemlja). Jednom kada odabere zemlju ili grad, voditeljka ne može više da promeni. Glasanje se sprovodi tako što se izabere pesma i izabere broj poena (videti sliku). Bira se pesma iz liste mogućih pesama i bira se broj poena iz liste poena (radio dugmad 12, 10 ili 8). Interfejs ima dva dugmeta GLASAJ i OBJAVI GLASOVE (ovo dugme postaje aktivno tek nakon što iz jedne zemlje dodele sve poene - 12, 10 i 8). Kada odabere podatke, voditeljka klikne na dugme GLASAJ. Ukoliko nešto nije odabrano, javljaju se poruke sa greškama (primeri ekrani sa porukama su dati u drugom redu). Kada glasa, korisnik dobija prozor u kome treba da potvrdi čekboks da želi da snimi glasove (opciono) i dugme OK čime se vraća na prethodni prozor za glasanje. Kada objavi glasove, voditeljka klikom na dugme izlazi iz aplikacije za glasanje, čime se glasanje te zemlje završava.
- a) Testirati ovaj GUI korišćenjem FSM (*Finite State Machine*) grafa i napisati kompletne sekvence koje ne mogu da se izvrše (FCIS - *Faulty Complete Interaction Sequences*) u ovom interfejsu.
- b) Napisati klase ekvivalencije za sledeće korisničke ekrane i opis specifikacije koji je dat.

The image shows the main voting interface and four error message screens. The main interface is titled "Evrovizija 2019 - Aplikacija za glasanje" and contains three main sections: "Ko glasa?" with dropdowns for "Zemlja:" and "Grad"; "Pesma" with a list of songs including "Toto Cotugno [ITA] - Insieme", "Elena Paparizu [GRE] - My number one", "Marija Šerifović [SRB] - Molitva", "Ruslana [UKR] - Wild Dances", "Azucar Moreno [ESP] - Bandido", and "ABBA [SWE] - Waterloo"; and "Broj glasova" with radio buttons for 12, 10, and 8. There are "GLASAJ" and "OBJAVI GLASOVE" buttons at the bottom.

The four error message screens are:

- Poruka:** "Niste odabrali pesmu za koju glasate" with a "U redu" button.
- Poruka:** "Niste odabrali broj dodeljenih poena" with a "U redu" button.
- GLAS:** "From SERBIA, Belgrade: 12 points for: Elena Paparizu [GRE] - My number one" with a checkbox "Sačuvaj glasove" and an "OK" button.
- Prikaz objavljenih glasova:** "List of votes from SERBIA, Belgrade:" followed by a list: "1. Elena Paparizu [GRE] - My number one = 12", "2. ABBA [SWE] - Waterloo = 10", "3. Ruslana [UKR] - Wild dances = 8" and an "Izlaz iz aplikacije" button.

6. [8] Za dijagram klasa aplikacije prikazan na slici integraciono testirati aplikaciju sledećim pristupima:

- od dna ka vrhu,
- sendvič testiranjem.

U nizu koraka potrebno je objasniti svaki od ova dva pristupa. Skicirati šta tačno sadrži deo sistema koji se testira (koje module, stabove, drajvere) u svakom koraku, do potpune integracije sistema. Ukratko objasniti svaki pristup (ne morate objašnjavati svaki korak).

Dijagram prikazuje kupovinu na Internetu. Svaki kupac ima svoj nalog (*Account*) sa jedinstvenim ID. Nalog ima korpu za kupovinu i narudžbinu (*Order*). Narudžbina može imati više stavki (*LineItem*) koje predstavljaju dostupne proizvode u radnji (*Product*). Klasa (*WebUser*) služi za logovanje korisnika.

