
Elektrotehnički fakultet u Beogradu
Katedra za računarsku tehniku i informatiku

Predmet: Testiranje softvera (SI3TS)
Nastavnik: Prof. dr Dragan Bojić
Asistent: dipl. ing. Dražen Drašković
Ispitni rok: Januar 2016.
Datum: 22.01.2016.

Kandidat:* _____

Broj indeksa:* _____

*Ispit traje 3 sata, a u toku prvog sata nije dozvoljeno napuštanje ispita.
Upotreba literature nije dozvoljena.*

<i>Zadatak 1</i>	_____ /4	<i>Zadatak 5</i>	_____ /10
<i>Zadatak 2</i>	_____ /12	<i>Zadatak 6</i>	_____ /7
<i>Zadatak 3</i>	_____ /9	<i>Zadatak 7</i>	_____ /6
<i>Zadatak 4</i>	_____ /12		

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

Rok u kome je odbranjen domaći:* _____ (primer: januar 2016)

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. [4] Nabrojati i sa po jednom rečenicom objasniti aktivnosti planiranja testa u okviru procesa testiranja. Ko je odgovoran za planiranje testa?

2. [12] a) Nacrtati konačni automat za klasu *CoffeeMachine*. Uslove u eventualnim uslovljenim prelazima označiti sa [uslov].
b) Napraviti tabelu pokrivanja sa 1 izmenom za automat iz tačke a).
c) Razviti test primere (sa konkretnim podacima) za tabelu iz tačke b).
d) Napraviti tabelu pokrivanja nelegalnih prelaza za automat iz tačke a).

```
class CoffeeMachine{
    Inventory i=new Inventory(0,0);
    int price=10;
    int amtPaid=0;

    makeCoffee() {
        boolean canMakeCoffee=true;
        if(amtPaid<price)
            canMakeCoffee=false;
        if(canMakeCoffee)
            amtPaid-=price;
    }

    addCoins(int amt){
        amtPaid+=amt;
    }

    addInventory(int coffee, int sugar){
        boolean canAdd=true;
        if(coffee<0||sugar<0)
            canAdd=false;
        else{
            i.setSugar(i.getSugar()+sugar);
            i.setCoffee(i.getCoffee()+coffee);
        }
        return canAdd;
    }
}
```


3. [9]

a) Za klasu *CoffeeMachine* datu u zadatku 2 odrediti *intraclasse dataflow* graf.

b) Navesti sve DU lance u grafu dobijenom u tački a).

c) Razviti test primere koji pokrivaju sve DU lance iz tačke b). Za svaki test primer navesti koji DU lanci su pokriveni.

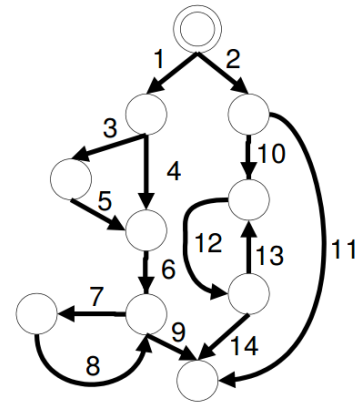
4. [12]

a) Odrediti broj ciklomatske kompleksnosti datog grafa.

b) Navesti McCabe-ov bejzline algoritam.

c) Odrediti putanje za dati graf po algoritmu iz tačke b).

d) Nacrtati matricu incidencije za putanje dobijene pod c) i na osnovu nje odrediti putanju koja predstavlja zbir putanja dobijenih u tački c). Navesti koja je to putanja.



5. [10] Potrebno je testirati veb aplikaciju “Prodavnica cipela”, sa formom za registraciju prikazanom na slici. Programski kod ove veb aplikacije nije poznat, pa se veb aplikacija testira na osnovu specifikacije zadatka. Kada se klikne na stavku menija dobija se jedna od stranica: men.xhtml, woman.xhtml i kids.xhtml. Na slici je prikazana stranica signup.xhtml.

The image shows a registration form for Timberland. At the top, there is a dark navigation bar with the Timberland logo on the left and the menu items 'MEN', 'WOMAN', 'KIDS', and 'SIGN UP' in white text. Below this is a light-colored registration form. The form contains the following fields and elements:

- Username:** A text input field with a placeholder 'Username'.
- Password:** A text input field with a placeholder 'Password'.
- Email Address:** A text input field with a placeholder 'Email Address'.
- Mobile Number:** A text input field with a placeholder 'Mobile Number'.
- Date of Birth:** A text input field with a placeholder 'DD/MM/YYYY'.
- Gender:** Radio buttons for 'Male' and 'Female'.
- Select Your Card:** A dropdown menu.
- Card Number:** A text input field with a placeholder 'YYYY-YY-YY-YY'.
- Your Promo Code:** A text input field.
- Terms and Conditions:** A checkbox labeled 'I've read the terms and conditions'.
- Submit:** An orange button labeled 'OK'.

At the bottom of the form area, there is a dark bar with the text 'Copyright © 2015'.

Potrebno je testirati da polje username ima najmanje 6 znakova, a najviše 16 znakova, koji mogu biti velika ili mala slova engleskog alfabeta ili brojevi (0-9), ali tako da username ne sme počinjati brojem. Zna se da polje Password ima najmanje 8 znakova, koji mogu biti velika ili mala slova engleskog alfabeta i brojevi. Potrebno je obezbediti i pravilan format e-mail adrese: nesto@domen.poddomen.X, gde je X org.rs ili co.rs ili org.me ili co.me, gde je poddomen opcioni, a domen obavezan). Kod domena i poddomena je dovoljno da postoji najmanje 1 karakter, koji predstavlja malo slovo ili broj. Broj telefona počinje sa 060, 063, 064, 065, 066, 069 i nastavlja se sa 6 ili 7 cifara. U listi za odabir kartice *Select Your Card* postoje sledeće opcije VISA, Maestro, MasterCard, American Express, DINA Card. Card number mora da ima tačno 16 cifara. Your promo code se izračunava kada se klikne na to polje, i formira se prema sledećem obliku: username + DD + MM + YY, gde je DD oznaka dana rođenja, MM oznaka meseca, a YY oznaka poslednje dve cifre godine. Na primer, ako se unese username softversko i datum rođenja 15.05.1995. promo kod koji treba da se dobije je: softversko150595.

Klikom na dugme OK ukoliko su sve provere ispunjene i ukoliko je čekirano I've read... korisniku se prikazaju poruka “Welcome!”, u suprotnom prikazuje se poruka o grešci/greškama (“Check your username!”, “Check your password!”...).

a) Metodom podele na klase ekvivalencije, kombinovano sa metodom graničnih slučajeva, testirati datu veb aplikaciju, pri čemu treba testirati i ulazne podatke i izlazne podatke.

b) Metodom kombinatornog testiranja *All-pairs*, napisati skup test primera koji će testirati sledeću formu sa neke druge veb stranice iste veb aplikacije:

Unos artikla:

Velicina:

- 35
- 36
- 37
- 38
- 39
- 40
- 41
- 42
- 43
- 44
- 45
- 46

Kolicina:

za suvo vreme
 za kisno vreme
 otporne na sneg

c) Za tačku pod b) nacrtati EFG (*event flow graph*), ako je neophodno da se unese količina (pozitivna celobrojna vrednost) i čekira bar jedan čekboks. Iz padajuće liste podrazumevano je odabrana prva veličina.

6. [7] Data je sledeća klasa `OsobaInfo` koja čuva podatke o nekoj osobi:

```
package adresar;
public class OsobaInfo{
    private String name,address,email;
    private int id, phone;

    public OsobaInfo() {
        name = "";
        address = "";
        email = "";
        id = 0;
        phone = 0;
    }

    public OsobaInfo(int id, String name, String address,
        int phone, String email) {
        this.id = id;
        this.name = name;
        this.address = address;
        this.phone = phone;
        this.email = email;
    }

    public void setId(int i) { id = i; }
    public void setName(String n) { name=n; }
    public void setAddress(String a) { address=a; }
    public void setPhone(int ph) { phone=ph; }
    public void setEmail(String e) { email=e; }
    public int getId( ) { return id; }
    public String getName() { return name; }
    public String getAddress() { return address; }
    public int getPhone() { return phone; }
    public String getEmail() { return email; }
}
}
```

Potrebno je napraviti niz sa osobama i doraditi Javinu metodu `pronadjiOsobu` u okviru klase `OsobaDAO`, koja vraća niz koji zadovoljava tekstualni podatak koji se šalje kao argument metode, a zatim napisati JUnit test primere u okviru posebne klase, tako da test primeri testiraju postojanje jedne, više ili nijedne osobe u napravljenom nizu.

```
package adresar;
public class OsobaDAO{
    private ArrayList osobeList;
    public OsobaDAO(){
        osobeList = new ArrayList();
    }
    public ArrayList pronadjiOsobu(String name){
        //implementirati
    }
}
}
```


7. [6] Formirati mutant programe za program *KonvertujDecimalniBroj*, ako se primenjuju sledeći mutacioni operatori:

- a) ROR (relacioni operator zamene) koji zamenjuje relacioni operator sa $=$, $>$, $<$, \geq , \leq , \neq , $!=$.
- b) AOR (aritmetički operator zamene) koji se primenjuje na izrazu: **result = result + residue**
* **multiplier**, odnosno aritmetičke operacije se menjaju sa $+$, $*$, $-$, $/$ i $\%$.

```
public class KonvertujDecimalniBroj {
    public void convert(int decimal, int base) {
        int result = 0;
        int multiplier = 1;
        while (decimal > 0) {
            int residue = decimal % base;
            decimal = decimal / base;
            result = result + residue * multiplier;
            multiplier = multiplier * 10;
        }
        System.out.println("Binarna predstava: " + result);
    }

    public static void main(String args[]) {
        KonvertujDecimalniBroj conv = new KonvertujDecimalniBroj();
        conv.convert(127,2); //izvršavanje konverzije
    }
}
```

Za svaki od ova dva operatora napraviti sledeću tablicu sa test primerima:

Program	Test 1	Test 2	...	Test m	Skup LIVE	Skup KILL
P1						
Mutanti						
M ₁						
M ₂						
...						
M _n						

Zatim izračunati mutacione rezultate koji se dobijaju primenom operatora ROR, odnosno operatora AOR.