
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: Februar 2013.
Datum: 08.02.2013.

Kandidat:* _____

Broj indeksa:* _____

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

<i>Zadatak 1</i>	_____ /4	<i>Zadatak 5</i>	_____ /10
<i>Zadatak 2</i>	_____ /8	<i>Zadatak 6</i>	_____ /12
<i>Zadatak 3</i>	_____ /10	<i>Zadatak 7</i>	_____ /8
<i>Zadatak 4</i>	_____ /8		

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

Rok u kome je odbranjen domaći:* _____ (primer: januar 2013)

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] Navesti naziv i dati po jedan konkretan primer za različiti načine predikcije izlaznih rezultata.

2. [8] Prikazati:

- a) klasni dijagram
- b) use/include hijerarhiju za međuklasno testiranje datih java klasa.

```
public class Address implements java.io.Serializable {
    String getAddress1(){...}
    String getAddress2(){...}
    String getState(){...}
    String getPostCode(){...}
    String getCountry(){...}
}

public final class Bookings implements java.io.Serializable {
    abstract class Booking implements Serializable {
        static final int ERROR = -4;
        static final int CARD_DECLINED = -3;
        static final int UNAVAILABLE = -2;
        static final int CANCELLED = -1;
        static final int PENDING = 0;
        static final int SUCCESS = 1;

        int getStatus(){...}
        void setStatus(int status){...}
        String getCreditCardAuthorizationCode(){...}
        void setCreditCardAuthorizationCode(String code){...}
        Person[] getPeople(){...}
        Customer getCustomer(){...}
        CreditCard getCreditCard(){...}
    }

    class final FlightBooking extends Booking {
        Flight[] getFlights(){...}
    }

    class final HotelBooking extends Booking {
        String getHotelName(){...}
        Address getHotelAddress(){...}
        Date getCheckinDate(){...}
        Date getCheckoutDate(){...}
    }

    class final RentalCarBooking extends Booking {
        String SMALL = "SMALL";
        String MEDIUM = "MEDIUM";
        String LARGE = "LARGE";

        String getRentalAgency(){...}
        String getCarClass(){...}
        Date getPickupDate(){...}
        Date getDropoffDate(){...}
    }
}
```

```

    Customer getCustomer(){...}
    CreditCard getCreditCard(){...}
    FlightBooking createFlightBooking(Person[] people, Flight[] flights){...}
    HotelBooking createHotelBooking(Person[] people, String hotelName,
        Address hotelAddress, Date checkinDate, Date checkoutDate){...}
    RentalCarBooking createRentalCarBooking(Person[] people,
        String rentalAgency, String carClass, Date pickupDate,
        Date dropoffDate){...}
    FlightBooking getFlightBooking(){...}
    HotelBooking getHotelBooking(){...}
    RentalCarBooking getRentalCarBooking(){...}
}

public class CreditCard implements java.io.Serializable {
    static final String AMEX = "AMEX" {...}
    static final String MC = "MASTERCARD" {...}
    static final String VISA = "VISA" {...}
    Name getCardHolder(){...}
    String getCardType(){...}
    String getCardNumber(){...}
    Date getExpiryDate(){...}
    int getSecurityDigits(){...}
}

public class Customer extends Person {
    String getAccountNumber(){...}
}

public class Flight implements java.io.Serializable {
    String getFlightNumber(){...}
    String getFromAirportCode(){...}
    String getToAirportCode(){...}
    Date getDepartureTime(){...}
    Date getArrivalTime(){...}
}

public class Name implements java.io.Serializable {
    String getFirstName(){...}
    Character getMiddleInitial(){...}
    String getLastName(){...}
    String getFullName(){...}
}

public class Person implements java.io.Serializable {
    Name getName(){...}
    Address getAddress(){...}
}

public class BookingService {
    void placeBooking(Bookings.Booking booking) throws BookingException {...}
    void cancelBooking(Bookings.Booking booking) throws BookingException {...}
}

```

3. [10] Na predmetu SI3ABC postoje predispitne obaveze (projekat i laboratorijske vežbe) i pismeni deo ispita (dva kolokvijuma i završni ispit). Projekat nosi 20% poena, laboratorijske vežbe 20%, a pismeni ispit 60 % (ili kolokvijumi po 10% i završni ispit 40% ili samo završni ispit 60%). Fakultetski informacijski sistem omogućava nastavniku da na osnovu unetih podataka u veb aplikaciji dobije ukupan broj poena i ocenu, na osnovu algoritma koji vrši to izračunavanje i koji treba testirati preko veb interfejsa.

Metodom graničnih slučajeva treba da testirate veb formu (sa slike levo) u koju se u 5 tekstualnih polja unose poeni. Student ne mora, ali može da izade na oba kolokvijuma. Kao uslov za polaganje ispita, student mora da ima minimalno 20/40 poena na predispitnim obavezama i minimalno 30/60 poena na pismenom delu ispita. Kolokvijumi se rade u vidu testova od 20 pitanja, koji mogu imati maksimalno 10 poena, a minimalno -10 poena (jer svaki pogrešan odgovor donosi negativne poene). Ocena se računa na osnovu ukupnog broja poena (BP maksimalan je 100), koji se dobija kao suma poena na predispitnim obavezama i pismenom delu ispita, kao u tabeli (sa slike desno).

SI3ABC - unos poena za studenta 2006/0396

Predispitne obaveze: Projekat
 Lab.vežbe

Projekat:

Lab:

Pismeni ispit:

- samo završni ispit
- K1 + završni ispit
- K2 + završni ispit
- K1 + K2 + završni ispit

Prvi kolokvijum:

Drugi kolokvijum:

Završni ispit:

Poena:
Ocena:

Ocena	Broj poena
10	BP ≥ 91
9	BP ≥ 81
8	BP ≥ 71
7	BP ≥ 61
6	BP ≥ 51
5	BP < 51

4. [8] Neka je dat Java program koji za određenog studenta na osnovu njegovog/njenog uspeha, materijalnog stanja i finansijske situacije na fakultetu, vraća rezultat molbe za umanjenje školarine za osnovne studije. Pretpostaviti da se funkciji dostavljaju ispravno svi argumenti.

```
public static String oslobodjen_skolarine (boolean kriza, int
broj_stipendija, int god_studija, double prosek, boolean položio_OS,
boolean mat_ugrozen) {
    if(!kriza && broj_stipendija > 0){
        if(prosek > 8.5 && mat_ugrozen)
            return "Oslobodjen 70%";
        if(god_studija >= 3 && prosek > 9 && položio_OS){
            return "Oslobodjen 50%";
        } else if((god_studija >= 2 && prosek > 9.8) ||
(god_studija >= 2 && prosek > 9.5 && položio_OS) ) {
            return "Oslobodjen 30%";
        }
    } else
        return "Niko nije oslobodjen zbog nedostatka novca";
    return "Nema studenata koji zadovoljavaju kriterijume";
}
```

- a) Razviti test primere za dati program metodom pokrivanja svih uslova i odluka (navesti prvo uslove i odluke, a zatim i konkretne test primere).
- b) Razviti test primere za dati program metodom pokrivanja svih složenih uslova.

5. [10] Neka su u programskom jeziku Java date dve metode za uređivanje:

```
public void uredi1() throws Greska{
    int n = niz.length;
    for (int i=1; i<n; i++) {
        double p = niz[i];
        int j = i - 1;
        while (j>=0 && p < niz[j]) {
            niz[j+1] = niz[j--];
            prikazi(); //funkcija za prikazivanje niza
        }
        niz[j+1] = p;
        prikazi();
    }
}

public void uredi2() throws Greska{
    if (q > p) {
        int i = p-1, j = q;
        while (true) {
            do i++; while (niz[i] < niz[q]);
            do j--; while (j>=0 && niz[j] > niz[q]);
            if (i >= j) break;
            double b = niz[i]; niz[i] = niz[j]; niz[j] = b;
            prikazi(); //funkcija za prikazivanje niza
        }
        double b = niz[i]; niz[i] = niz[q]; niz[q] = b;
        prikazi();
        uredi2 (p, i-1); uredi2 (i+1, q);
    }
}
```

Testirati sve petlje u ove dve metode i napisati minimalan broj testova, tako da se uspešno testira i jedna i druga metode. Ukratko obrazložiti svaki test.

Da li iz procesa testiranja može da se zaključi koja je metoda efikasnija? Ukratko obrazložiti.

6. [12] a) Neka su date sledeće klase. Primetite da je A2 podklasa klase A1, koja je podklasa klase A:

```
class A {
    int x;
    A a;
    public f() { g(); x=a.x; }
    public g() { a=new A(); a.x=... }
    public h() { x=0; a.x=0; }
}

class A1 extends A {
    public g() { super.g(); a.x=... }
}

class A2 extends A1 {
    public g() { h() }
}
```

Nacrtati Yo-Yo graf za navedene klase i objasniti šta je prednost Yo-Yo grafa.

b) Za sledeći programski kod prizan klasama A i B, nacrtati graf kontrole toka i ispisati sve DU lance. Ukoliko broj DU lanaca zavisi i od sadržaja uslova if(...) napisati kako zavisi i prikazati implementacije if uslovnih izraza. Za svaku promenljivu koja se koristi u programu, napisati da li svaka određena upotreba predstavlja c-use ili p-use.

```
class A {
    pozivamNesto(...) {
A1:      x = 1;
A2:      if(...)
A3:      ... //ne menja se x
A4:      if(...)
A5:      x = 2;
A6:      nekoB.uradiNesto(x);
        ...
    }
}
```

```
class B {
    uradiNesto(int n) {
B1:      int x = 8;
B2:      if(...) {
B3:          y = n;
B4:      } else {
B5:          y = -n;
B6:      }
B7:      x += 2;
B8:      x = 13;
B9:      y = x;
    }
}
```


7. [8] Neka je data sledeća funkcija:

```
public int funkcija(int x, int y) {  
    int rezultat = 1;  
    int brojac = y;  
    while(brojac > 0){  
        rezultat = rezultat * x;  
        brojac--;  
    }  
    return(rezultat);  
}
```

- a) Koliko će mutanata generisati ova funkcija kada se primeni ABS (absolute value) operator? Napisati sve mutante koji su generisani ABS operatorom, napisati koji su live/killed i izračunati mutacioni rezultat.
- b) Razmotriti sledeći mutant, koji se dobija promenom linije 5:
 rezultat = rezultat * abs(x);
 Napraviti test primer koji razlikuje rezultat od ovog mutanta ili pokazuje da je ekvivalentan ovoj funkciji.
- c) Napisati minimalni skup JUnit testova koji testiraju dati program.