
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: Drugi kolokvijum (decembar 2014.)
Datum: 02.12.2014.

Kandidat:* _____

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

*Kolokvijum traje 2 sata, prvih sat vremena nije dozvoljeno napuštanje kolokvijuma.
Upotreba literature nije dozvoljena.*

Zadatak 1 _____/5
Zadatak 2 _____/4
Zadatak 3 _____/4
Zadatak 4 _____/7

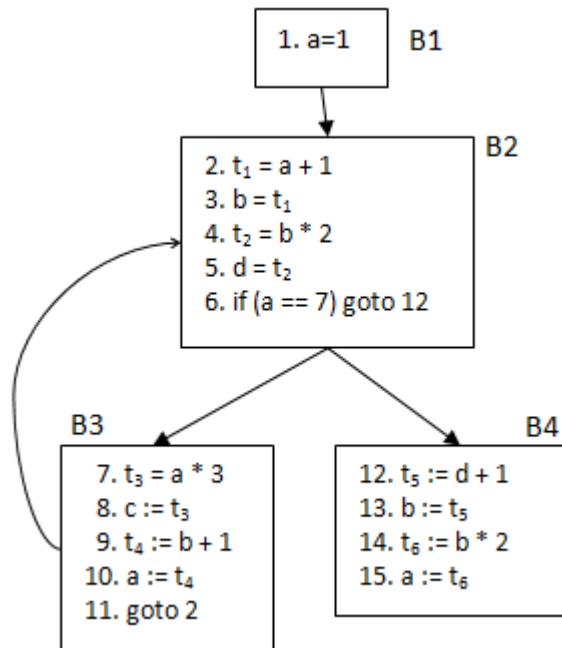
Ukupno na kolokvijumu: _____/20

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 i precizno**.

* popunjava student.

1. [5]

- a) Odrediti in, out, gen i kill skupove za osnovne blokove sa donjeg grafa.
- b) Odrediti DU lance za donji graf na osnovu informacija dobijenih pod a).



Odgovor:

2. [4] Na slici je prikazana veb stranica koja korisniku na osnovu unetih podataka prikazuje vrstu jakne koju treba da obuče na osnovu vremenskih podataka, koje korisnik unosi. Metodom testiranja svih parova (*all-pairs*) napisati test primere i ukratko objasniti prednosti i nedostatke tog metoda testiranja.

Padavine: DA NE

Temperatura: (*C)

Vetrovitost: (m/s)

Potvrdi

Odgovor:

3. [4] Neka je data funkcija u prog. jeziku Java sa 5 argumenata.

```

public boolean greska(int a){
    if(a==0)
        return false;
    else
        return true;
}

public int funkcija(int a, int b, int c, int d, float e){
    if(a==0)
        return 0;
    int x=0;
    if((a==b)||((c==d)&& greska(a)))
        x=1;
    e=1/x;
    return (int)e;
}

```

Napisati minimalan skup test primera za pokrivanje sledećih vrsta testiranja belom kutijom:

Vrsta testiranja	Test primeri
odluka (Branch/Decision Coverage)	
uslova (Condition Coverage)	
višestrukih uslova (Multiple Condition Coverage)	

4. [7] Dati su delovi implementacija klasa za sortiranje: apstraktne klase *AbstractSort* i klase koja je iz nje izvedena, *QuickSort*:

```

abstract class AbstractSort{
    int v[];
    public AbstractSort(int nDim){
        v = new int[nDim];
        for (int i = 0; i<nDim; i++)
            v[i] = 0;
    }
    public void setVector(int nVector[]){
        for (int i=0; i<nVector.length; i++){
            v[i] = nVector[i];
        }
    }
    public void swap(int i, int j){
        int tmp;
        tmp = v[i];
        v[i] = v[j];
        v[j] = tmp;
    }
    abstract public void sort();
    ...
}

```

```

class QuickSort extends AbstractSort
{
    public QuickSort(int nDim)
    {
        super(nDim);
    }

    private void quick(int low, int high) {
        int i = low, j = high;
        if(high<0) return;
        int pivot = v[low + (high-low)/2];
        while (i < j) {
            while (v[i] < pivot) {
                i++;
            }
            while (v[j] > pivot) {
                j--;
            }

            if (i <= j) {
                swap(i, j);
                i++;
                j--;
            }
        }
        if (low < j)
            quick(low, j);
        if (i < high)
            quick(i, high);
    }

    public void sort()
    {
        quick(0, v.length-1);
    }
}

```

- a) [3] Napisati test primere koji uspešno testiraju petlje u metodi *quick(int,int)*. Za svaki test potrebno je definisati dužinu niza celih brojeva i sve elemente niza.
- b) [3] Primeniti mutacioni operator ROR koji zamenjuje relacioni operator u petljama WHILE drugim relacionim operatorima i napisati mutacioni skor za test primere definisane pod a). Koristiti samo mutante prvog reda.
- c) [1] Koliki je mutacioni skor ukoliko u metodi *quick(int,int)* primenimo operator kojim izraz $x--$ mutira u $x++$ i $--x$?