

9. vežbe

1. zadatak

```
/* Naucnik se bavi stvaranjem svih mogucih lanaca
duzine n sastavljenih od atoma plutonijuma i olova,
tako da ni jedan lanac ne moze da izazove atomsku
eksploziju. Do eksplozije bi doslo ako bi se u lancu
jedan do drugog nasla dva atoma plutonijuma. Koliko
"bezbednih" lanaca duzine n moze da stvori naucnik? */
```

```
class Lanci {
    static int L(int n) {
        if (n == 1)
            return 2;
        else if (n == 2)
            return 3;
        else
            return L(n-1) + L(n-2);
    }

    public static void main(String[] args) {
        System.out.print("Unesite duzinu lanca: ");
        int n = Svetovid.in.readInt();
        System.out.print("L(" + n + ") = " + L(n));
    }
}
```

2. zadatak

`/* Napisati program koji učitava niz celih brojeva i pronalazi
i ispisuje dužinu najdužeg rastućeg podniza u tom nizu.
Primer: za niz 2 4 1 6 2 8 1, podniz 2 4 6 8 ima dužinu 4 */`

```
class NajduziRastuciPodniz {  
  
    static int nadjINRP(int[] niz, int index) {  
        if (index == 0) {  
            return 1;  
        }  
        else {  
            int maxDuzina = 0;  
            for (int i = 0; i < index; i++) {  
                if (niz[i] < niz[index]) {  
                    int duzina = nadjINRP(niz, i);  
                    if (duzina > maxDuzina) {  
                        maxDuzina = duzina;  
                    }  
                }  
            }  
            return maxDuzina + 1;  
        }  
    }  
}
```

2. zadatak

```
public static void main(String[] args) {  
    System.out.print("Unesite duzinu niza: ");  
    int duz = Svetovid.in.readInt();  
    int[] niz = new int[duz];  
    for(int i = 0; i < niz.length; i++){  
        System.out.print("Unesite " + i + ". element niza: ");  
        niz[i] = Svetovid.in.readInt();  
    }  
  
    // Potrebno je izracunati nadjiNRP(niz, i) za svaki indeks i  
    // u nizu, i pronaci najveću vrednost  
    int rez = 0;  
    for (int i = 0; i < niz.length; i++) {  
        int tmp = nadjiNRP(niz, i);  
        if (tmp > rez)  
            rez = tmp;  
    }  
    System.out.println("Duzina najduzeg rastuceg podniza je " + rez);  
}
```

3. zadatak

```
/* Napisati program koji za dati ceo broj n,  $0 \leq n \leq 30$ , ispisuje vrednost  
   elementa rekurentnog niza  $f_n$ . Niz je definisan na sledeci nacin:  
    $f_n = 2f_{n-1} - f_{n-2} + f_{n-3}$ ,  $n \geq 3$   
    $f_2 = 2$ ,  $f_1 = 1$ ,  $f_0 = 0$   
   Element  $f_n$  izracunati:  
   (a) rekurzivno preko definicije,  
   (b) rekurzivno pomocu akumulirajuceg parametra,  
   (c) iterativno.  
   U slucajevima (b) i (c) voditi racuna o efikasnosti resenja.  
*/
```

```
class Recurrent1 {  
    static final int granica = 30;  
  
    // (a) rekurzivno preko definicije  
    static int fRec1(int n) {  
        if (n <= 2)  
            return n;  
        else  
            return 2*fRec1(n-1) - fRec1(n-2) + fRec1(n-3);  
    }  
}
```

3. zadatak

```
// (b) rekurzivno pomocu akumulirajuceg parametra
static int ff(int f2, int f1, int f0, int n) {
    if (n == 0)
        return f0;
    else
        return ff(2*f2-f1+f0, f2, f1, n-1);
}

static int fRec2(int n) {
    return ff(2, 1, 0, n);
}
```

3. zadatak

```
// (c) iterativno, ali prostorno neefikasno
static int fIter1(int n) {
    int f[] = new int[n + 1];
    f[0] = 0;
    f[1] = 1;
    f[2] = 2;
    for(int i = 3; i <= n; i++)
        f[i] = 2*f[i-1] - f[i-2] + f[i-3];
    return f[n];
}
```

3. zadatak

```
// (c) iterativno, efikasno
static int fIter2(int n) {
    int fn;
    int f0 = 0;
    int f1 = 1;
    int f2 = 2;
    if (n == 0)
        fn = f0;
    else if (n == 1)
        fn = f1;
    else if (n == 2)
        fn = f2;
    else {
        fn = 0;
        for (int i = 3; i <= n; i++) {
            fn = 2*f2 - f1 + f0;
            f0 = f1;
            f1 = f2;
            f2 = fn;
        }
    }
    return fn;
}
```


3. zadatak

```
public static void main(String[] args) {  
    System.out.print("Unesite n (0 <= n <= " + granica + "): ");  
    int n = Svetovid.in.readInt();  
    if (0 <= n && n <= granica) {  
        System.out.println("fRec1(n) = " + fRec1(n));  
        System.out.println("fRec2(n) = " + fRec2(n));  
        System.out.println("fIter1(n) = " + fIter1(n));  
        System.out.println("fIter2(n) = " + fIter2(n));  
    }  
    else {  
        System.out.println("n je van dozvoljenih granica");  
    }  
}
```