

8. vežbe

1. zadatak

/ Program koji učitava prirodan broj i rekurzivnom metodom ispisuje sve cifre unetog broja u obrnutom redosledu.*

**/*

```
class ObrniRek {
```

```
    static void obrni(int n) {
```

```
        if (n < 10) {
```

```
            System.out.print(n);
```

```
        }
```

```
    else {
```

```
        System.out.print(n % 10);
```

```
        obrni(n / 10);
```

```
    }
```

```
}
```

```
public static void main(String[] args) {
```

```
    System.out.print("Unesite prirodan broj: ");
```

```
    int broj = Svetovid.in.readInt();
```

```
    System.out.print("Obrnuti broj je: ");
```

```
    obrni(broj);
```

```
}
```

```
}
```

2. zadatak

```
/* Program koji rekurzivnom metodom racuna i ispisuje  
   zbir cifara unetog prirodnog broja.  
   */
```

```
class ZbirCifaraRek {
```

```
    static int zbirCifara(int n) {  
        if (n < 10) {  
            return n;  
        }  
        else {  
            return n % 10 + zbirCifara(n / 10);  
        }  
    }
```

```
    public static void main(String[] args) {  
        System.out.print("Unesite prirodan broj: ");  
        int broj = Svetovid.in.readInt();  
        if (broj < 0) broj = -broj;  
        System.out.println("Zbir cifara unetog broja je: " + zbirCifara(broj));  
    }  
}
```

3. zadatak

```
/* Napisati program koji ucitva dva pozitivna cela broja  
   i rekurzivnom metodom izracunava njihov najveći zajednicki  
   delitelj.
```

```
*/
```

```
class NZDRek {  
  
    static int nzd(int prvi, int drugi) {  
        if (prvi == drugi) {  
            return prvi;  
        }  
        else if (prvi > drugi) {  
            return nzd(prvi - drugi, drugi);  
        }  
        else {  
            return nzd(prvi, drugi - prvi);  
        }  
    }  
}
```

3. zadatak

```
public static void main(String[] args) {  
    int prvi, drugi;  
    do {  
        System.out.print("Unesite prvi pozitivan ceo broj: ");  
        prvi = Svetovid.in.readInt();  
    } while (prvi <= 0);  
    do {  
        System.out.print("Unesite drugi pozitivan ceo broj: ");  
        drugi = Svetovid.in.readInt();  
    } while (drugi <= 0);  
    int rez = nzd(prvi, drugi);  
    System.out.println("Najveci zajednicki delitelj je: " + rez);  
}  
}
```

4. zadatak

```
/* Program koji učitava string i rekurzivnom metodom  
   odredjuje da li je uneti string palindrom.  
*/
```

```
class PalRek {  
  
    static boolean palindrom(String s, int poc, int kraj) {  
        if (poc >= kraj) {  
            return true;  
        }  
        else if (s.charAt(poc) == s.charAt(kraj)) {  
            return palindrom(s, poc + 1, kraj - 1);  
        }  
        else {  
            return false;  
        }  
    }  
}
```

4. zadatak

```
public static void main(String[] args) {  
    System.out.print("Unesite string: ");  
    String s = Svetovid.in.readLine();  
    if (palindrom(s, 0, s.length()-1)) {  
        System.out.println("Uneti string jeste palindrom.");  
    }  
    else {  
        System.out.println("Uneti string nije palindrom.");  
    }  
}
```

5. zadatak

/* Program koji učitava rastuci niz celih brojeva duzine duz i jos jedan broj, i rekurzivnom metodom binarnog pretrazivanja pronalazi trazeni broj u nizu i ispisuje njegovu poziciju. Ukoliko se trazeni broj ne nalazi u nizu, ispisati odgovarajucu poruku. */

```
class BinTraRek {
    static int binTrazenje(int[] niz, int levi, int desni, int broj) {
        if (levi > desni) {
            return -1;
        }
        else {
            int srednji = (levi + desni) / 2;
            if (niz[srednji] == broj) {
                return srednji;
            }
            else if (niz[srednji] > broj) {
                return binTrazenje(niz, levi, srednji-1, broj);
            }
            else {
                return binTrazenje(niz, srednji+1, desni, broj);
            }
        }
    }
}
```


5. zadatak

```
public static void main(String[] args) {  
    System.out.print("Unesite broj elemenata u monotono rastucem nizu: ");  
    int duz = Svetovid.in.readInt();  
    int[] niz = new int[duz];  
    for (int i = 0; i < niz.length; i++) {  
        do {  
            System.out.print("Unesite " + i + ". element niza: ");  
            niz[i] = Svetovid.in.readInt();  
        } while (!(i == 0 || niz[i-1] < niz[i]));  
    }  
    System.out.print("Unesite broj koji treba pronaci: ");  
    int broj = Svetovid.in.readInt();  
  
    int poz = binTrazenje(niz, 0, duz-1, broj);  
    if (poz == -1) {  
        System.out.println("Broj se ne nalazi u nizu");  
    }  
    else {  
        System.out.println("Broj je na poziciji " + poz);  
    }  
}
```