

Teme:

- Delo z bitnimi operatorji.

Naloga 1

Javanski program v podatek javanskega tipa `int` zgenerira naključno vrednost v obsegu 8 bitov. V spodnjih osmih bitih prešteje število enic in število ničel iz bitnega zapisa in izpiše 1 v primeru, da je enic več, oziroma 0 v nasprotnem primeru. Skodirajte ga.

Naloga 2

Javanski program v podatek javanskega tipa `int` zgenerira naključno vrednost v obsegu 8 bitov. Teh 8 bitov predstavlja dva polbita (angl. nibble), z 1 označimo polbit z večjo težo, z 0 polbit z manjšo. Program odgovori na vprašanje, vrednost katerega polbita je večja. Spišite ga.

Naloga 3

Opišite vlogo bitnih operatorjev `>>` in `<<`, opišite tudi, kaj spodnji program izpiše.

```
public static void main(String[] args) {  
  
    int a=3;  
  
    int rez;  
  
    rez = a << 1;  
  
    System.out.println(rez+" "+Integer.toString(rez,2)+" "+String.format("%x", rez));  
  
    rez = a << 2;  
  
    System.out.println(rez+" "+Integer.toString(rez,2)+" "+String.format("%x", rez));  
  
    rez = a << 3;  
  
    System.out.println(rez+" "+Integer.toString(rez,2)+" "+Integer.toString(rez,16));  
  
}
```

Naloga 4

Napišite program, ki podano desetiško vrednost izpiše v obliki dvojiškega števila. Dodajte še izpis v osmiški in šestnajstiški obliki: to dvoje dosežete z oblikovanjem načina izpisa (npr.

```
System.out.printf(" %d %o %x ",18,18,18));.
```

Naloga 5

Program naj ugotavlja, koliko naključnih števil je potrebno ustvariti, da z njihovo vsebino lahko zapolnimo vseh spodnjih 16 bitov. Recimo: število 5 pokrije bita 0 in 2, število 10 bita 3 in 1. Ti števili skupaj zapolnita spodnje štiri bite. Število 7 zatem bi bilo nekoristno, ker so njegove enice že vsebovane v številih 5 in 10. Program med izvajanjem izpiše vsa tista števila, ki 'prispevajo' svoje bite oz. so koristna, nekoristnih pa ne. Ob koncu izpiše še, koliko števil je bilo potrebno generirati, koliko od tega je bilo koristnih (ustreznih) in kolikšen je bil odstotek koristnih. Naključna števila izbirajte z intervala:

a. [0-33000]

b. [0-Integer.MAX_VALUE]