

Demo 3 / 28.9

Tehtävät

V1. Tee [Ville](#)-tehtävät: 2.5, 2.6, 7.1, 7.3, 7.4, 7.5, 7.6

1. **M: 24.4 Negatiiviset binääriluvut:** Oletetaan että tietokoneessa on vain 8 bitin luvut. Eli siihen tilaan on tuloksen mahdollista. Laske allekkain yhteen seuraavat binääriluvut huolellisesti muistinumerot kirjoittaen (myös vasemmanpuoleisesta parista ulos lähtevä muistinumero):

a)	b)	c)	d)
0101 1111	0101 1111	0000 0001	1111 1111
0001 1111	0101 1111	1111 1111	1111 1111

Useimmiten nykytietokoneissa käytetään negatiivisille luvuille niin sanottua kahden komplementtia. Eli positiivinen luku muutetaan negatiiviseksi muuttamalla kaikki bitit päinvastaisiksi ja sitten lisäämällä saatuun lukuun 1. Vastaavasti kun lukua muutetaan "ihmismuotoon", katsotaan sen ensimmäinen bitti ja jos se on 1, niin kyseessä on negatiivinen luku ja se muutetaan positiiviseksi ottamalla siitä kahden komplementti (kaikki bitit päinvastoin ja +1). Tällöin tulostuksessa tulostetaan ensin -merkki ja sitten itse luvun arvo. Esimerkiksi:

$$3 = 0000\ 0011$$

$$\begin{aligned} -3 \text{ tehdään seuraavasti: } & 1) \text{ kaikki päinvastoin } 1111\ 1100 \\ & 2) +1 \qquad \qquad \qquad = 1111\ 1101 = -3 \end{aligned}$$

Vastaavasti jos meillä on binääriluvut 0010 1101 ja 1101 1111 ja ne pitäisi tulkita, niin tulkinta aloitetaan seuraavasti:

0010 1101 luku on positiivinen, eli 45
 1101 1111 luku on negatiivinen, siis ensin 2:n komplementti
 $0010\ 0000 + 1 = 0010\ 0001 = 33$, eli tulos on -33

Tulkitse kohtien a)-d) summattavat luvut ja summat

- e) kun 8-bitin koneessa on vain positiiviset luvut käytössä
- f) kun 8-bitin koneessa on negatiivisille luvuille käytössä 2-komplementti
- g) Mitkä tulokset olivat oikeita e)-kohdan tapauksessa
- h) Mitkä tulokset olivat oikeita f)-kohdan tapauksessa
- i) Anna esimerkki missä kahden negatiivisen luvun summa olisi positiivinen 8-bitin koneessa jossa negatiiviset luvut esitetään 2:n komplementtina.
- j) Anna laskemiesi lukujen ja "virheellisten" tulosten ja muistinumeroiden perusteella sääntö siitä, milloin tuloksen voi todeta olevan oikein ja milloin väärin.

2. **M: 7. Muuttujat:** Miten esittelisit (tyyppi ja nimi) muuttujat seuraavia tilanteita varten. Anna myös esimerkki miten sijoittaisit muuttujalle jonkin arvon. Vastauksen voit palauttaa Java-tiedostonakin.

- a) Työmatkan pituus polkupyörällä
- b) Suoritettujen opintopisteiden määrä
- c) Koko tutkinnon laajuus opintopisteissä
- d) Tehtyjen demojen lukumäärä
- e) Kirjan ensimmäinen kirjain
- f) Sadan metrin juoksun voittajan aika.

3. **M: 7. Muuttujat**: Arvostelee seuraavat muuttujien tyypit ja nimet. Ovatko syntaktisesti oikein? Entä ovatko hyviä valintoja:

```
int mansikanKiloHinta;
double first,second,final;
char omaNimi;
double metriMm;
double tuntejaVuorokaudessa;
real nopeus;
double AuringonEtaisyys;
int kuukaudenKeskiLampo;
int kissoja,2_kanaa,kolmeKoiraa;
int i,j,l,I;
```

4. **M: 9. Aliohjelman paluuarvo**: Katso netistä miten Fahrenheitit muunnetaan Celsius-asteiksi. Kirjoita (funktio)aliohjelma jota voidaan kutsua seuraavasti:

```
double lampotilaC;
lampotilaC = fahrenheitToCelsius(13);
```

Kirjoita vastaava kutsu funktiolle joka muuntaa paunoja (1b) grammoiksi. Kirjoita myös tuo funktio. Muista kirjoittaa aina asianmukaiset kommentit aliohjelmiin!

5. **Aliohjelmat: (Kuormittaminen)** Ota luennolla tehty lumiukko:

<http://users.jyu.fi/~vesal/kurssit/ohj1/luennot/2009s/090915ti/Lumiukot2.java>

ja muuta se sellaiseksi, että lumiukon voi piirtää neljällä eri tavalla (eli on neljä eri lumiukko -aliohjelmaa eri parametrillä, nyt on kaksi aliohjelmaa valmiina). Muista kommentointi! Vinkki: aloita tekemällä aliohjelmalla jota voidaan kutsua tavalla d).

- a) Ilman minkään pallon kokoa
lumiukko(window,124,150);
- b) antamalla alapallon koko
lumiukko(window,20,100,30);
- c) antamalla kahden alimman pallon koko
lumiukko(window,30,300,40,20);
- d) antamalla kaikkien kolmen pallon koko
lumiukko(window,274,190,50,40,30);

Toki voit jatkaa pääteohjauksessa tekemääsi versioon, joka piirtää ukolle silmät ja nenän. Muista "testata" kaikkia neljää eri aliohjelmaa.

6. **M: 7.5 Aritmeettiset lausekkeet**: Kirjoita lausekkeet, joilla lasket (tee Java-ohjelmaksi, main riittää, ei tarvitse aliohjelmaa)

- a) vuorokauden sekuntien lukumäärän
- b) vuoden tuntien lukumäärän
- c) kauanko valolla kuluu aikaa matkaan auringosta maahan (keskietäisyydet riittävät).

d) valovuosi kilometreinä

- B1. **M: [18.2 Käyttäjän syötteen lukeminen](#):** Tee ohjelma, joka kysyy käyttäjältä merkkijonon (käytä Scanner-luokkaa, M: [18.2 Käyttäjän syötteen lukeminen](#)). Sitten tulostetaan käyttäjän syöttämä merkkijono kokonaan isoilla kirjaimilla kirjoitettuna ja kokonaan pienillä kirjaimilla kirjoitettuna (ks. [String-luokan](#) dokumentit M: [12. Merkkijonot](#)). Tulosta jonosta vielä kolme ensimmäistä kirjainta.

GURU-tehtävät

- G1-2 Ota tehtävän 5 lumiukko ja muuta se olioksi. Laita näyttöön 3 lumiukkoa, joista jokainen pyörii omaa tahtiaan. Katso automaattinen pyörittäminen [sample/UkkoSample.java](#) .