

## R0027 – Ohjelmoinnin perusteet Kehittämisharjoitukset - D

### OSA-1: Muuttujien vertailua, Ehtolauseet, Toistolauseet (do-while), String-luokan metodeita

#### TEHTÄVÄT

##### 1. Lukujen vertailua (Ehtolauseiden Kertausta)

Tee ohjelma, joka kysyy käyttäjältä ikää. Kirjoita ohjelmaan ehtolauseet (if-else if), jotka reagoivat annettuun ikään seuraavalla tavalla:

- Jos ikä on pienempi kuin 0 tai suurempi kuin 150, tulostetaan "Virheellinen ikä!"
- Jos ikä on alle 6 tulostetaan "Esikoululainen"
- Jos ikä on 6-14 tulostetaan "Olet lapsi"
- Jos ikä on tasan 14 tulostetaan "Haastava ikä"
- Jos ikä on 16 - 18 tulostetaan "Lähes aikuinen"
- Jos ikä on suurempi kuin 18, mutta pienempi kuin 30 tulostetaan, "Olet täysi-ikäinen, muttet vielä keski-ikä kriisissä".
- Jos ikä on suurempi tai yhtä suuri kuin 30, mutta pienempi kuin 45 tulostetaan: "Olet keski-ikässä"
- Jos ikä on suurempi kuin 45 mutta alle 65 tulostetaan "Vielä ehtii ennen eläkettä!"
- Muuten ohjelma tulostaa "Olet eläkeläinen".

##### 2. Sähköpostin tarkistus

Tee ohjelma, joka kysyy käyttäjältä sähköpostiosoitetta ja tarkastaa onko se kelvollinen. Syöte on kelvollinen, mikäli se täyttää kaikki ao. vaatimukset:

- sisältää yhden @ -merkin
- sisältää vähintään yhden pisteen (.)
- on pidempi kuin 5 merkkiä
- keksitkö muita kelpoisuusehtoja?

*Vihjeitä: Merkkijonon sisältämiä merkkejä voit tutkia indexOf-metodia käyttäen, kts. ohje Java APIsta. Merkkijonon pituutta voit tutkia length() -metodilla.*

### 3. Valikon tulostaminen

Tee switch – case –rakennetta käyttäen seuraavanlainen valikko  
Tulosta jokaisen valinnan kohdalla ruudulle jotain ko. kohtaan liittyvää

```
*****
* 1 - Käynnistä auto
* 2 - Aja autoa
* 3 - Sammuta auto
* 4 - Lopeta peli
*****
```

### 4. Käyttäjätunnus ja salasana (Merkkijonojen vertailua, toistorakenteita)

Tee ohjelma, joka aluksi kysyy käyttäjältä oikeaa käyttäjätunnusta ja salasanaa. Sijoita luetut tiedot talteen muuttujiin nimeltä String tunnus ja String salasana.

Tämän jälkeen ohjelma simuloi tietokoneen sisäänkirjautumisvaihetta ja pyytää käyttäjää syöttämään tunnuksen ja salasanan. Jos käyttäjä syöttää oikeat arvot, tulostetaan ”Oikein meni, ” sekä sen perään syötetty tunnus ja salasana. Jos arvot ovat väärät, ohjelma tulostaa virheilmoituksen.  
Ohjelma ei lopu väärään vastaukseen vaan tunnusta ja salasanaa kysytään kunnes ne syötetään oikein.  
Käytä tehtävässä do – while tai while - toistorakennetta.

Huom!

Toisin kuin lukuja, merkkijonoja ei Javassa voi vertailla käyttäen yhtäsuuruusoperaattoria. Niiden vertaamiseen on omat metodinsa *equals* ja *equalsIgnoreCase*.

Esim.

```
String teksti = "Mika";
String toinenTeksti = "Minna";

if ( teksti.equals(toinenTeksti) ) {
    System.out.println("Samat tekstit!");
}
```

### 5. Levykauppa

Tee ohjelma, joka ostaa levyjä niin kauan, kunnes lompakko on tyhjä. Kysy käyttäjältä aluksi kuinka paljon lompakossa on rahaa. Yksi levy maksaa 20 € Tulosta jokaisella toistokerralla jäljellä olevan rahan määrä.

### 6. PIN-koodi

Tee ohjelma, joka kysyy käyttäjältä PIN-koodia. Kyselyä toistetaan kunnes oikea luku (3120) osuu kohdalle. Käytä tehtävässä do – while tai while - toistorakennetta.

## 7. Toistorakenteita: Valikon tulostaminen

Tee switch – case –rakennetta käyttäen seuraavanlainen valikko (voit käyttää myös edellisen kerran tehtävää pohjana). Tulosta jokaisen valinnan kohdalla ruudulle jotain ko. kohtaan liittyvää.

Liitä valikkoon do – while –toistorakenne, joka tulostaa valikon sinnikkäästi ruudulle aina käyttäjän valitseman operaation jälkeen. Toistorakenteesta poistutaan vasta kun käyttäjä valitsee kohtan 4 – ”Lopeta peli”.

```
*****
* 1 - Käynnistä auto
* 2 - Aja autoa
* 3 - Sammuta auto
* 4 - Lopeta peli
*****
```

## OSA-2

### Alitrmisen ajattelun syventämistä, Toistolauseet (for), String-luokan metodeita

Hyödynnä Eclipsen Debug-näkymää toistolauseiden ja muuttujien seuraamisessa.

## TEHTÄVÄT

### 8. Kertotaulu

Tee ohjelma, joka tulostaa käyttäjän syöttämän luvun kertotaulun väliltä 1..20. Käytä for –lausetta. Muotoile tulostus ”kauniiksi”.

### 9. Kurssitaulukko

Tee ohjelma, joka tulostaa markka/euro -taulukon näyttöön. Ohjelma tulostaa ensin otsikkorivin ”Euroa Markkaa” ja sen jälkeen taulukon yhden euron välein väliltä 1 .. 10. Kurssina voit käyttää seuraavaa: yksi euro on n. 5.94573 markkaa. Käytä for- toistolauseetta.

Kun saat ohjelman toimimaan, voit kehittää sitä pyöristämällä desimaalit esim. 2 tai 3:n numeron tarkkuudelle käyttäen System.out.printf –metodia. Muotoile tulostusta myös esim. tasaamalla luvut oikealle, 10 merkin mittaiseen taulukkoon (kts printf-ohje).

```
System.out.printf(“%.2f”, muuttujanNimi);
```

Esimerkkitulostus:

```
*****
* Euroa      =      Markkaa  *
*****

1    =      5,95
2    =      11,89
3    =      17,84
4    =      23,78
```

5	=	29,73
6	=	35,67
7	=	41,62
8	=	47,57
9	=	53,51
10	=	59,46

#### 10. Merkkien laskemista

Tee ohjelma, joka kysyy käyttäjältä syötteen sekä sen, mitä merkkiä etsitään. Tämän jälkeen ohjelma käy läpi merkkijonon ja laskee montako kertaa etsittävä esiintyi siinä. Käytä tehtävässä jotain toistorakennetta.

*Vinkki: String-luokan charAt – metodeista on apua. Kts. Java API.*

#### 11. Karkausvuosien tulostaminen

Tee algoritmin, joka tulostaa ruudulle kaikki karkausvuodet vuosien 1753-2050 välillä. (Kts. edellisistä tehtävistä karkausvuosien tulostamisen logiikka).

Toteuta tämän jälkeen algoritmina hahmottelemasi ohjelma. Käytä ohjelmassa for –toistolauseetta. Voit käyttää apuna edellisen kerran Karkausvuosi –tehtävää.

Karkausvuosia ovat siis kaikki 400:lla jaolliset vuodet sekä lisäksi 4:llä jaollisista vuosista ne, jotka eivät ole 100:lla jaollisia. Voit kerrata esimerkkiratkaisusta karkausvuosien laskemiseen käytetyn algoritmin.

Esimerkkitulostus:

```
Vuosi 1756 oli karkausvuosi.
Vuosi 1760 oli karkausvuosi.
Vuosi 1764 oli karkausvuosi.
Vuosi 1768 oli karkausvuosi.
Vuosi 1772 oli karkausvuosi.
...
Vuosi 2040 oli karkausvuosi.
Vuosi 2044 oli karkausvuosi.
Vuosi 2048 oli karkausvuosi.
```

**THE FOLLOWING EXERCISES ARE MORE DEMANDING. TO BE ABLE TO CODE THE RIGHT SOLUTION YOU WILL NEED FIRST TO FIGURE OUT AN ALGORITHM THAT RESOLVES THE TASK. BEFORE STARTING TO WRITE THE JAVA CODE, YOU SHOULD FIRST THINK AND WRITE ON A PAPER HOW THE ALGORITHM WORKS (USING PSEUDO-CODE OR FLOW-CHARTS). IF YOU ARE ABLE TO RESOLVE THESE EXERCISES, THIS MEANS THAT BESIDES KNOWING HOW TO APPLY PROGRAMMING BASIC STRUCTURES/ELEMENTS YOU HAVE ALSO DEVELOPED YOUR PROBLEM SOLVING SKILLS, ANALYTICAL SKILLS, AND ALGORITHM DEVELOPMENT SKILLS.**

## 12. Väärinpäin

Tee ohjelma, joka kysyy käyttäjältä syötteen ja tulostaa sen tämän jälkeen väärinpäin ruudulle. Käytä tehtävässä jotain toistorakennetta sekä `charAt()` –metodia.

## 13. Palindromeja

Tee sovellus, joka kysyy käyttäjältä syötteen ja kertoo onko kyseessä palindromi. Käytä tässä jotain toistorakennetta sekä em. `String`-luokan metodeja. Tunnettuja palindromeja ovat mm. ”saippuakauppias” ja ”innostunut sonni”.

## 14. Kirjainten arvausta

Tee ohjelma, joka pyytää käyttäjää arvaamaan kirjainta ja etsii sitä määritellystä avainsanasta. Tämän jälkeen avainsana tulostetaan vain oikein arvatun kirjaimen osalta. Alla esimerkkiajo:

```
Avainsana: Banaani
Arvaa kirjain: a
Löytyy kohdasta: 1
*a*****
```

*Vinkki: `String`-luokan `indexOf` ja `charAt` – metodeista on apua. Kts. `Java API`.*

## 15. Sensurointia

Tee ohjelma, joka sensuroi syötteestä joukon arveluttavia sanoja, ja korvaa ne ao. taulukon mukaisesti. Tulosta myös ruudulle montako avainsanaa löydettiin.

```
”kissa” → k*****
”koira” → ###
”auto” → poistetaan kokonaan
”mopo” → välilyönti
```

Voit lisätä joukkoon muitakin haluamiasi sanoja.

*Vinkki: `String`-luokan `replace` – alkuisista metodeista on apua. Kts. `Java API`.*

## 16. Sensurointia, osa 2

Tee ohjelmasta kehitetty versio, joka kysyy ensin syötteen, sitten mitä etsitään ja kolmanneksi millä etsitty korvataan. Tulosta myös montako avainsanaa korvattiin.

## 17. Tekstin parsimista

Tee ohjelma, joka käy läpi String –muuttujaan tallennettua HTML-koodia (alla) ja poimii heading-tägien välissä olevat tekstit ja tulostaa ne ruudulle \*-merkeillä ympäröityinä. Heading tägejä ovat siis <H1></H1>.

String HTML = "<h1>Tässä on otsikko</h1><p>Lorem ipsum dolorum...</p><h2>Toinen otsikko</h2>";

Kun saat ohjelman toimimaan, laajenna sitä siten, että H-tägin tilalla voi olla mikä tahansa avainsana.

*Vinkki: String-luokan metodeista on apua. Esim. yllä olevissa tehtävissä käytetyistä. kts. myös substring yms. Kts. Java API.*

## 18. Sisäkkäiset silmukat (for-for)

Tee sovellus "Kolmio", joka osaa tulostaa kolmion tekstigrafiikkana. Ohjelma kysyy käyttäjältä kuinka korkea kuvio tulostetaan. Tämän jälkeen ohjelma tulostaa ruudulle annetun levyisen ja korkuisen kolmion.

Esimerkkiajo:  
Anna kolmion koko: 5

```
0
0 0
0 0 0
0 0 0 0
0 0 0 0 0
```

## 19. Hirsipuu

Tee ohjelma joka toimii perinteisen hirsipuu-pelin tavoin. Eli kyselee kirjaimia ja paljastaa oikein arvattuja, kunnes koko sana on näkyvissä. Voit myös toteuttaa laskurin, joka päättää pelin mikäli vääriä arvauksia on kertynyt liikaa.

```
Arvaa kirjain: b
b*****
Arvaa kirjain: a
ba*aa**
Arvaa kirjain: i
ba*aa*i
Arvaa kirjain: n
banaani
Arvasit koko sanan! Arvauksia: 4.
```