

Oppimispäiväkirja  
Teemu Keskinen

SISÄLLYS

[1. Tehtävät 2](#_Toc189488742)

[1.1 Tehtävä 1, viikko 3 2](#_Toc189488743)

[1.2 Tehtävä 2, viikko 4 3](#_Toc189488744)

[1.3 Tehtävä 3, viikko 5 4](#_Toc189488745)

[1.4 Tehtävä 4, viikko 6 5](#_Toc189488746)

.

# Tehtävät

## Tehtävä 1, viikko 3

Tehtävässä luodaan muuttuja numberToGuess käyttäen javan math-kirjaston random operaatiota. Operaatiolle määritetään minimi- ja maksimiarvot muuttujilla min ja max, jolloin voidaan helposti muokata arvauspelin haastavuutta. Käyttäjän syötettä kysytään scanner-kirjaston avulla ja tallennetaan käyttäjän antama vastaus muuttujaan. Tätä muuttujaa sen jälkeen verrataan random operaattorilla generoituun lukuun ja tämän perusteella annetaan vihje käyttäjälle, onko generoitu luku pienempi vai isompi kuin käyttäjän arvaama luku. Mikäli käyttäjä arvaa luvun kerrotaan hänelle tästä. Mikäli käyttäjä ei arvaa lukua oikein yritysmäärän sisällä niin tulostetaan oikea luku käyttäjälle näkyviin.

Kuva, joka sisältää kohteen teksti, kuvakaappaus, Fontti

Kuvaus luotu automaattisesti

<https://github.com/TeemuKeskinen/ohjelmoinnin-syventavat/blob/main/tehtavat/week%203/NumberGuessingGame.java>

## Tehtävä 2, viikko 4

Tehtävässä luodaan luokka Birthday, jossa ympäristömuuttujasta puretaan pihalle käyttäjän syntymäpäivä käyttämällä System.getenv() komentoa. Syntymäpäivää käytetään tämän jälkeen laskemaan käyttäjän ikä päivinä, lasketaan kulunut aika käyttäjän viime syntymäpäivästä ja lasketaan myös aika käyttäjän seuraavaan syntymäpäivään. Ohjelmassa on myös lisättynä tarkistus, onko käyttäjän ikä jaollinen 1000 ja tästä onnitellaan käyttäjää. Päivämäärää käsitellään javan java.time.localDate ja java.time.temporal.ChronoUnit kirjastojen avulla. Ensimmäisenä puretaan ympäristömuuttujasta pihalle localDate muuttujaksi sopiva arvo localDate.parse() komennon avulla. Tämän lisäksi tarvitsemme localDate.now(), jota käytämme arvona mihin syntymäpäivää verrataan. Vertailun jälkeen kerrotaan käyttäjälle hänen ikänsä päivinä, milloin hänen seuraava syntymäpäivä on ja milloin hänen viimeisin syntymäpäivänsä oli. Mikäli käyttäjän ikä päivinä on jaollinen tuhannella tulostetaan hänelle vielä sen lisäksi onnittelut tästä.

Kuva, joka sisältää kohteen teksti, kuvakaappaus, Fontti

Kuvaus luotu automaattisesti

<https://github.com/TeemuKeskinen/ohjelmoinnin-syventavat/blob/main/tehtavat/week%204/Birthday.java>

## Tehtävä 3, viikko 5

Tehtävässä luodaan ohjelma, jossa tallennetaan taulukkoon tehtävässä annetut tiedot ja tämän jälkeen luetaan luodusta taulukosta tietoa for-loopin avulla. GitHubissa olleiden valmiiden luokkien avulla saamme taulukosta helposti tiedon käsiteltyä valmiina olevien get()-kutsujen avulla. Tekstinä olevaa kuvausta emme halua kokonaan, joten jaamme sen split()-komennolla ja poimimme sieltä vain haluamamme osat. Päivämäärän saamme tapahtumasta pihalle localDate-kirjastossa olevalla getdate()-kutsulla. Teksti kirjotetaan käyttäjälle kutsumalla printf, jossa tekstiin on asetettu placeholdereita (”%s”) ja nämä placeholderit korvataan komennon lopussa annetuilla muuttujilla (”teksti”, muuttuja, muuttuja, muuttuja), tässä tapauksessa. Viimeisenä tallennetaan kaikki tapahtumat taulukkoon ja puretaan sieltä pihalle vain version nimi getDescription() kutsulla Event-luokasta. Tämän jälkeen taulukko muokataan aakkosjärjestykseen Arrays.sort()-komennolla ja printataan tämä sama pihalle println(”teksti” + Arrays.toString())) komennolla

Kuva, joka sisältää kohteen teksti, kuvakaappaus, Fontti

Kuvaus luotu automaattisesti

<https://github.com/TeemuKeskinen/ohjelmoinnin-syventavat/blob/main/tehtavat/week%205/MacOSRelease.java>

## Tehtävä 4, viikko 6

Tehtävässä luodaan ohjelma, joka pystyy näyttämään annettujen kriteerien perusteella valitut tapahtumat. Tapahtumia edustavat aiemmin tehdyn Event-luokan oliot. Tapahtumat tulevat varastosta, joka on toteutettu esimerkissä. Ohjelma käyttää komentoriviparametreja suodattaakseen tapahtumia päivämäärän ja kategorian perusteella. Komentoriviltä pyydetään kaksi parametria, päivämäärä ja kuukausi muodossa –mm-dd ja tapahtuman primary/secondary kuvaus. Myös pelkästään primary kuvauksella on mahdollista hakea tapahtumaa. Mikäli parametrit ovat virheellisiä palautetaan käyttäjälle virhe, jossa häntä ohjeistetaan oikeasta muotoilusta argumentteja varten. Mikäli tiedot on annettu oikein, suodatetaan tapahtumat annetun päivämäärän ja kategorian mukaan. Tapauksessa, jossa pelkästään primäärikategorian arvo on annettu, asetetaan toinen kategoria wildcard-arvoksi (\*). Tehtävän toteuttamista varten on Today.java tiedostoon pitänyt lisätä metodit: report(LocalDate filterDate, Category filterCategory): joka suodattaa ja tulostaa tapahtumat ja

isSameDate(LocalDate someDate, LocalDate otherDate): joka tarkistaa, ovatko päivämäärät samat (ilman vuotta). Näiden metodien lisäksi on myös category.java tiedostoon tehty muutoksia lisäämällä sinne matches-metodi, joka tarkistaa vastaavatko kategoriat toisiaan. Jos sekundaariarvo on wildcard, se hyväksytään mös.

Kuva, joka sisältää kohteen teksti, kuvakaappaus, Fontti

Kuvaus luotu automaattisesti



Lähdekoodi:

<https://github.com/TeemuKeskinen/ohjelmoinnin-syventavat/blob/main/tehtavat/week%206/Today.java>