

Oppimispäiväkirja  
Teemu Keskinen

SISÄLLYS

[1. Tehtävät 2](#_Toc188339454)

[1.1 Tehtävä 1, viikko 3 2](#_Toc188339455)

[1.2 Tehtävä 2, viikko 4 4](#_Toc188339456)

.

# 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

import java.util.Scanner;

public class NumberGuessingGame {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        int min = 1;

        int max = 100;

        int numberToGuess = (int) (Math.random() \* max) + min;

        int maxnumberOfTries = 7;

        int numberOfTries = 0;

        boolean hasGuessedCorrectly = false;

        while (numberOfTries < maxnumberOfTries && !hasGuessedCorrectly) {

            System.out.println("Guess the number between " + min + " and " + max + ":");

            int userGuess = scanner.nextInt();

            numberOfTries++;

            if (userGuess == numberToGuess) {

                hasGuessedCorrectly = true;

                System.out.println("Correct number.");

            } else if (userGuess < numberToGuess) {

                System.out.println("The number is higher.");

            } else {

                System.out.println("The number is lower.");

            }

        }

        if (!hasGuessedCorrectly) {

            System.out.println("The correct number was " + numberToGuess);

        }

        scanner.close();

    }

}

## 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

import java.time.LocalDate;

import java.time.temporal.ChronoUnit;

public class Birthday {

    public static void main(String[] args) {

        String birthdateStr = System.getenv("BIRTHDATE");

        if (birthdateStr == null || birthdateStr.isEmpty()) {

            System.out.println("Birthdate is not set, please setup environment variable BIRTHDATE");

            return;

        }

        LocalDate birthdate = LocalDate.parse(birthdateStr);

        LocalDate today = LocalDate.now();

        if (birthdate.isAfter(today)) {

            System.out.println("Your Birthdate is in the future.");

            return;

        }

        long days = ChronoUnit.DAYS.between(birthdate, today);

        LocalDate nextBirthday = birthdate.withYear(today.getYear());

        LocalDate latestBirthday = nextBirthday.minusYears(1);

        if(birthdate.getYear() == today.getYear()){

            latestBirthday = birthdate.withYear(today.getYear());

        }

        long daysFromBirthday = ChronoUnit.DAYS.between(latestBirthday, today);

        if (nextBirthday.isBefore(today) || nextBirthday.isEqual(today)) {

            nextBirthday = nextBirthday.plusYears(1);

        }

        long daysUntilNextBirthday = ChronoUnit.DAYS.between(today, nextBirthday);

        if (today.getMonth() == birthdate.getMonth() && today.getDayOfMonth() == birthdate.getDayOfMonth() && today.getYear() >= birthdate.getYear()){

            System.out.println("Happy Birthday!");

            System.out.println("You are " + days + " days old");

            if (days % 1000 == 0) {

                System.out.println("Nice round number");

            }

            return;

        }

        System.out.println("You are " + days + " days old");

        if (days % 1000 == 0) {

            System.out.println("Nice round number");

        }

        System.out.println("Your next birthday is " + daysUntilNextBirthday + " days away");

        System.out.println("Your last birthday was " + daysFromBirthday + " days ago");

    }

}

## 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