Tampereen ammattikorkeakoulu

Oppimispäiväkirja

5G00EU62-3005

Kuusisto Jaakko

Tietotekniikan tutkinto-ohjelma
Ohjelmistotekniikka

Sisällysluettelo

1 Tehtävä 1

Tehdään yksinkertainen arvauspeli, jossa aluksi alustetaan muuttujat ja loput ohjelmasta pyörii komentorivi while() -silmukassa.

Numeroiden alustus ja satunnaisluku math.random() -funktiolla:

```
Scanner scanner = new Scanner(System.in);
int targetNumber = (int) (Math.random() * 100) + 1;
int attempts = 7;
boolean guessedCorrectly = false;
```

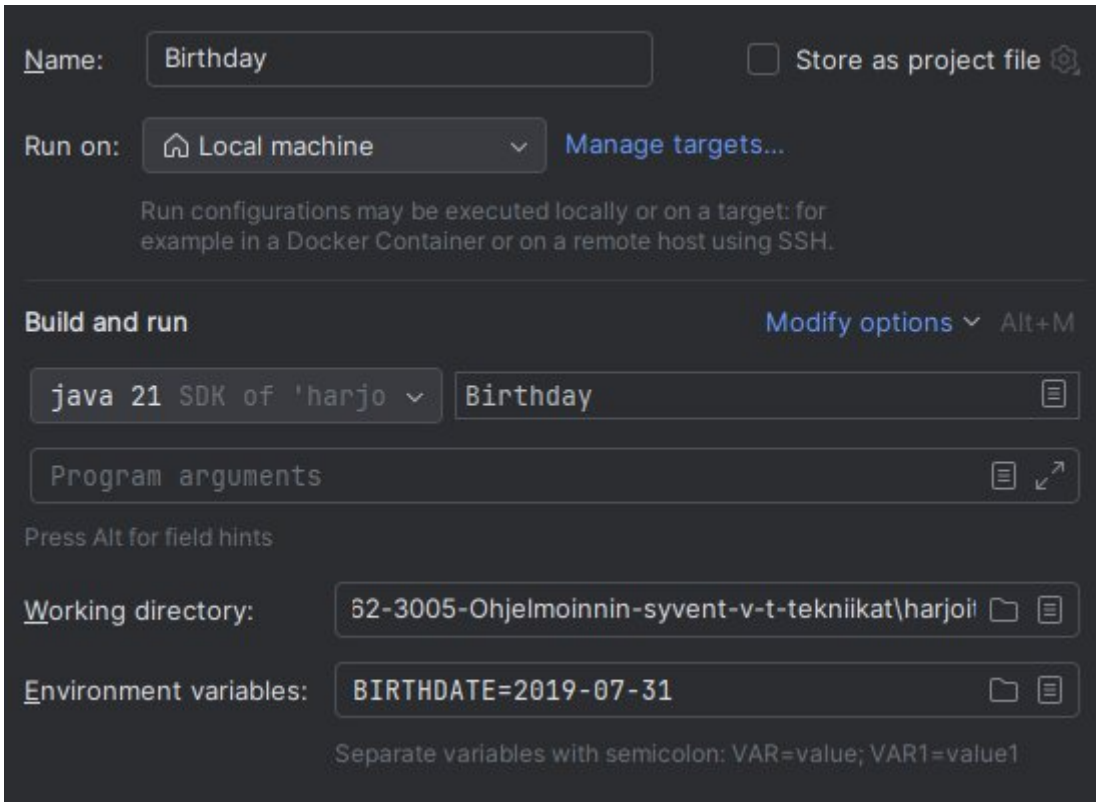
Alustuksen jälkeen silmukka, joka päättyy yritysten loppuun tai oikeaan arvaukseen.

```
while (attempts > 0 && !guessedCorrectly) {
    System.out.print("Enter your guess: ");
    int userGuess = scanner.nextInt();

    if (userGuess == targetNumber) {
        System.out.println("Congratulations! You guessed the correct number!");
        guessedCorrectly = true;
    } else if (userGuess < targetNumber) {
        System.out.println("Too low!");
    } else {
        System.out.println("Too high!");
    }

    attempts--;
    if (attempts > 0 && !guessedCorrectly) {
        System.out.println("Attempts left: " + attempts);
    }
}
```

Linkki projektin lähdekoodiin osoitteessa:
https://github.com/MikeDanton/5G00EU62-3005-Ohjelmoinnin-syvent-v-t-tekniikat/tree/main/harjoitus1/untitled/src

2 Tehtävä 2

Tehtiin Birthday-ohjelma, joka tulostaa syötteitä luetun BIRTHDATE nimisen ympäristömuuttujan perusteella.
Aluksi luotiin ympäristö ja asetettiin muuttuja ideen:



Aluksi asetetaan paikalliset muuttujat ja tehdään null-tarkistus BIRTHDATE:lle.

```
public class Birthday {
    public static void main(String[] args) {
        // Try to read the BIRTHDATE environment variable
        String birthdateEnv = System.getenv( name: "BIRTHDATE");

        if (birthdateEnv == null || birthdateEnv.isEmpty()) {
            System.out.println("Please set the BIRTHDATE environm
            return;
        }
```

Tämän jälkeen ohjelma on pitkä lista if-lauseita, jotka käyvät läpi ohjelman vaadittuja toiminnallisuuksia. Ohjelma on try-catch -blokin sisällä, jos DateTimeParseException heittää virheen väärästä tallennusformaatista jäsentelyn aikana.

```java
try {
    // Convert the birthday to a LocalDate object
    LocalDate birthdate = LocalDate.parse(birthdateEnv);
    LocalDate today = LocalDate.now();

    // Check if today is the user's birthday
    if (birthdate.getMonth() == today.getMonth() && birthdate.getDayOfMonth(
        System.out.println("Happy Birthday!");
    }

    // Calculate the difference in days between the birthday and today
    long daysBetween = ChronoUnit.DAYS.between(birthdate, today);

    if (daysBetween < 0) {
        System.out.println("Your birthday is in the future. Please check the
    } else {
        System.out.println("You are " + daysBetween + " days old.");

        // Check if the number of days is divisible by one thousand
        if (daysBetween % 1000 == 0) {
            System.out.println("That's a nice round number!");
        }
    }
} catch (DateTimeParseException e) {
    System.out.println("Invalid date format. Please use YYYY-MM-DD for the B
}
```

Ohjelma käyttää ChronoUnit ja LocalDate kirjastoja ajan käsittelyyn ja esim. jakojäännöstä jaollisuuden tarkistamiseen.

```
"C:\Program Files\Eclipse Adoptium\jdk-
You are 2001 days old.

Process finished with exit code 0
```

Yllä asetettu syntymäpäivä pitäisi antaa 2000, mutta voi olla, että LocalDate ja ChronoUnit eivät ota aikavyöhykkeitä huomioon täysin sellaisinaan tms. Bugi vaikuttaa olevan tämän tehtävänannon ulkopuolella.

Linkki lähdekoodiin:
[5G00EU62-3005-Ohjelmoinnin-syvent-v-t-tekniikat/harjoitus2 at main · MikeDanton/5G00EU62-3005-Ohjelmoinnin-syvent-v-t-tekniikat]

3 Tehtävä 3

Tehtävänä oli tehdä ohjelma, joka käytti kurssireposta löytyviä Event- ja Category-luokkia.
Aluksi muuttujien alustus käytettyjen luokkien rakentajien ja tehtävänannon datan perusteella:

```
Category macosCategory = new Category( primary: "apple", secondary: "macos");

Event[] events = {
        new Event(LocalDate.of( year: 2024, month: 9, dayOfMonth: 16), description: "macOS 15 Sequoia released", macosCategory),
        new Event(LocalDate.of( year: 2023, month: 9, dayOfMonth: 26), description: "macOS 14 Sonoma released", macosCategory),
        new Event(LocalDate.of( year: 2022, month: 10, dayOfMonth: 24), description: "macOS 13 Ventura released", macosCategory),
        new Event(LocalDate.of( year: 2021, month: 10, dayOfMonth: 25), description: "macOS 12 Monterey released", macosCategory),
        new Event(LocalDate.of( year: 2020, month: 11, dayOfMonth: 12), description: "macOS 11 Big Sur released", macosCategory)
};
```

For-silmukassa tehdään tulostuksen käsittely. LocalDate tulostaa isoilla kirjaimilla muuttujan, joten tarvitaan ylimääräistä käsittelyä tehtävänannon tulostukseen.

```
// Print each event in the specified format
for (Event event : events) {
    LocalDate date = event.getDate();
    String description = event.getDescription();
    String version = description.split( regex: " ")[1]; // e.g., "15"
    String name = description.substring(description.indexOf( str: " ", fromIndex: description.indexOf(" ") + 1) + 1, description.lastIndexOf( str: " released"));
    String weekday = date.format(DateTimeFormatter.ofPattern("EEEE")); // Directly get "Monday", "Tuesday", etc.

    System.out.printf("macOS %s %s was released on a %s%n", version, name, weekday);
}
```

Seuraavaksi kerätään käyttöjärjestelmien nimet String-taulukkoon, järjestetään se Arrays.sort()-metodilla ja tulostetaan.

```
    // Extract operating system names and sort alphabetically
    String[] osNames = new String[events.length];
    for (int i = 0; i < events.length; i++) {
        String description = events[i].getDescription();
        osNames[i] = description.substring(description.indexOf( str: " ", fromIndex: description.indexOf(" ") + 1) + 1, description.lastIndexOf( str: " released"));
    }

    Arrays.sort(osNames); // Sort alphabetically
    System.out.println("In alphabetical order: " + Arrays.toString(osNames));
}
```

Linkki lähdekoodiin:
https://github.com/MikeDanton/5G00EU62-3005-Ohjelmoinnin-syvent-v-t-tekniikat/tree/main/harjoitus3

4 Tehtävä 4

Tehtiin Today-ohjelma, johon voi lisätä/tarkastella päivän tapahtumia komentoriviargumenttien perusteella. Tehtävään annettiin valmiina Category- ja Event-luokat. Lisäksi valmiina oli pohja Today-luokka, jota tuli muokata tehtävänantoa vastaavaksi.

Category-luokkaan lisättiin tehdasmetodi, joka lajitteli String-objektin ja palautti uuden Category-objektin. Kaikki isot kirjaimet muutettiin pieniksi käsittelyn helpottamiseksi.

```
// Static factory method for parsing categories from strings
public static Category parse(String categoryString) {  1 usage  MikeDanton
    if (categoryString == null || categoryString.trim().isEmpty()) {
        throw new IllegalArgumentException("Invalid category string");
    }

    String[] parts = categoryString.split( regex: "/");
    String primary = parts[0].trim().toLowerCase();
    String secondary = (parts.length == 2) ? parts[1].trim().toLowerCase() : null;

    return new Category(primary, secondary);
}
```

Alla yhteenveto Today-luokan toiminnasta

```
public class Today {  MikeDanton
    private final List<Event> events;  7 usages

    public Today() {  1 usage  MikeDanton
        this.events = new ArrayList<>();
        loadEvents();
    }

    public static void main(String[] args) {...}

    private void loadEvents() {...}

    private void findAndPrintEvents(LocalDate filterDate, Category filterCategory) {...}

    private boolean isSameDay(LocalDate eventDate, LocalDate filterDate) {...}

    private boolean matchesCategory(Category eventCategory, Category filterCategory) {...}
}
```

loadEvents(), metodi esimerkkidatan alustukseen rakentajassa.

findAndPrintEvents(), metodi tapahtumien tulostukseen päivän ja kategorian perusteella. findAndPrintEvents () käyttää isSameDay() metodia sisäisesti, joka on apumetodi tarkastamaan, onko päivä sama.
matchesCategory() on myös apumetodi, jota findAndPrintEvents() käyttää.

Pääohjelman toiminta:

```
public static void main(String[] args) {  MikeDanton
    if (args.length != 2) {
        System.err.println("Usage: java Today --MM-DD category");
        System.exit( status: 1);
    }

    String dateInput = args[0];
    String categoryInput = args[1];

    if (!dateInput.matches( regex: "--\\d{2}-\\d{2}")) {
        System.err.println("Invalid date format. Use --MM-DD (e.g., --01-31)");
        System.exit( status: 1);
    }

    LocalDate filterDate;
    try {
        filterDate = LocalDate.parse( text: LocalDate.now().getYear() + dateInput.substring( beginIndex: 1));
    } catch (DateTimeParseException e) {
        System.err.println("Error parsing date: " + e.getMessage());
        System.exit( status: 1);
        return;
    }

    Category filterCategory;
    try {
        filterCategory = Category.parse(categoryInput);
    } catch (IllegalArgumentException e) {
        System.err.println("Invalid category format: " + e.getMessage());
        System.exit( status: 1);
        return;
    }

    Today app = new Today();
    app.findAndPrintEvents(filterDate, filterCategory);
}
```

Poimitaan aluksi komentoriviargumentit omiin String-objekteihinsa ja tehdään virheenkäsittely.
Tämän jälkeen muutetaan input-String -objektit localDate- ja Category-objekteiksi.
Lopuksi alustetaan uusi Today-objekti esimerkkidatalla ja kutsutaan findAndPrintEvents() argumentteina localDate- ja Category-objektit ja annetaan ohjelman tulostaa syöte komentoriville.

Esimerkkitulosteita:

```
PS C:\Koulu\5G00EU62-3005-Ohjelmoinnin-syvent-v-t-tekniikat\harjoitus4\src> java Today --09-20 oracle/java
2022: Java SE 19 released
PS C:\Koulu\5G00EU62-3005-Ohjelmoinnin-syvent-v-t-tekniikat\harjoitus4\src> java Today 09-20 oracle/java
Invalid date format. Use --MM-DD (e.g., --01-31)
PS C:\Koulu\5G00EU62-3005-Ohjelmoinnin-syvent-v-t-tekniikat\harjoitus4\src> java Today --09-20 java
No events found for the given date and category.
PS C:\Koulu\5G00EU62-3005-Ohjelmoinnin-syvent-v-t-tekniikat\harjoitus4\src>
```

Linkki lähdekoodiin:
https://github.com/MikeDanton/5G00EU62-3005-Ohjelmoinnin-syvent-v-t-tekniikat/tree/main/harjoitus4/src

5 Tehtävä 5

Oppimistavoitteina tällä viikolla oli ainakin ulkoisten kirjastojen käyttö, standardoitujen merkkijonojen käsittelly, sekä hakemistojen käyttä OS:ssä.

Ympäristöksi valittiin Maven, koska AI kertoi siihen kirjastojen lisäämisen olevan helpointa ja generoi seuraavan:

```
<dependencies>
    <dependency>
        <groupId>org.apache.commons</groupId>
        <artifactId>commons-csv</artifactId>
        <version>1.10.0</version>
    </dependency>
</dependencies>
```

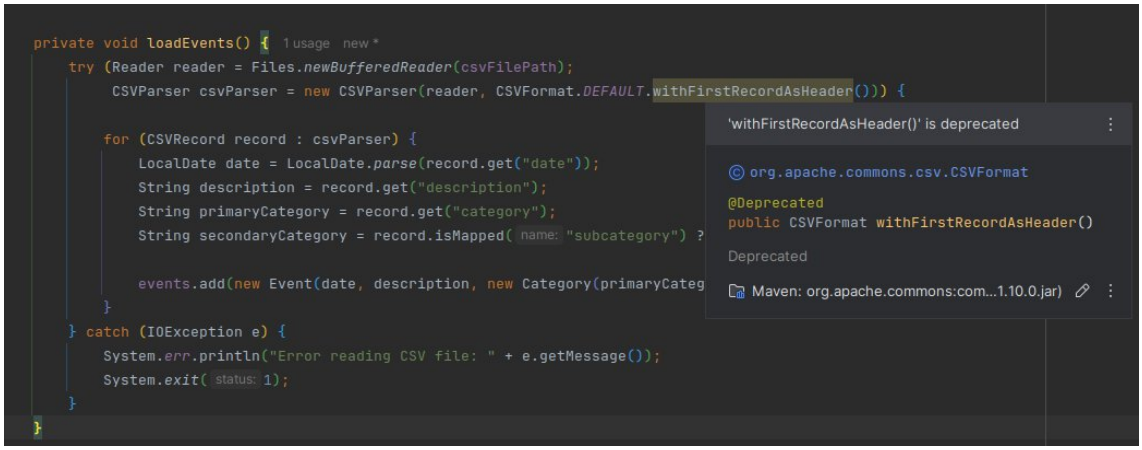
Idenä toimi intelliJ IDEA ja projekti saatiin toimimaan.

Standardoiduista merkkijonoista .csv ei tue sisäkkäisyyttä, (=nesting) (ilman kikkoja) joten käsittely on melko simppeliä. Huomautuksena kaikki toimivat kuitenkin samalla tavalla, kirjasto kerää jonkin taulukon kirjaston omaa luokkaa käyttäviä objekteja ja nämä pitää muuttaa haluttuun muotoon ja jatkokäsitellä. Täten, tekoälyn annettiin hoitaa ongelma, johon menisi muuten turhaa aikaa perehtyä kirjaston metodeihin jms. ”Comma separated values” on yksinkertainen, mutta näissäkin voi olla eksoottisempia syntakseja.

Rakentajassa asetettiin polku ja tarkastettiin onko tiedosto olemassa. AI:n mukaan tämä Javan tapa asettaa kotihakemisto on ”crossplatform” ja ainakin Linuxissa tämä toimi.

```
public CSVEventProvider(String fileName) {  1 usage  new *
    Path homeDir = Paths.get(System.getProperty("user.home"),  ...more: ".today");
    this.csvFilePath = homeDir.resolve(fileName);
    this.events = new ArrayList<>();

    if (!Files.exists(csvFilePath)) {
        System.err.println("Error: CSV file not found at " + csvFilePath);
        System.exit( status: 1);
    }

    loadEvents(); // Load events on initialization
}
```

Tämän jälkeen luettiin jäsentelyn lähteestä (record-termi yleinen), jotka saatiin sijoitettua suoraan Stringiin. Ide näyttää Deprecated -varoitusta, joten varmasti uudempi tapa olisi olemassa, mutta tämäkin toimi.

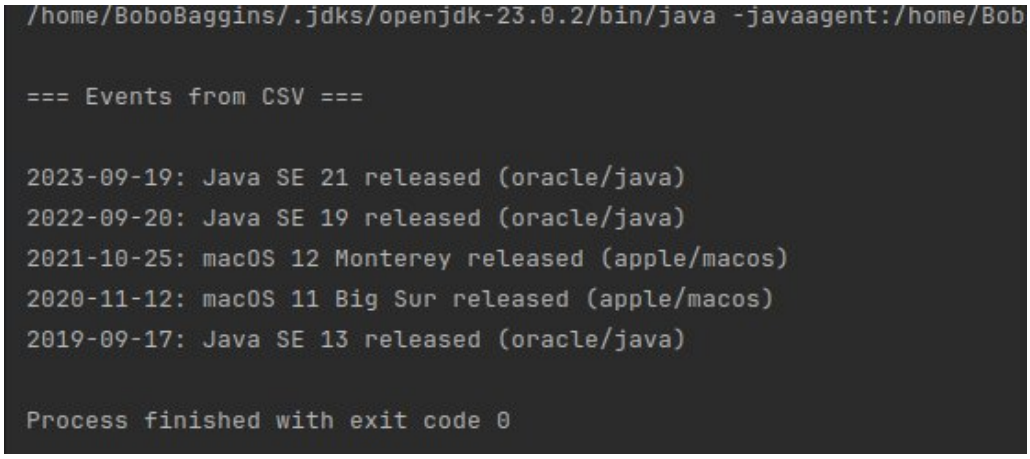


GitHubista löytyi Interface-tiedosto joten puskettiin haluttu data sen läpi:

Ja pääohjelmassa tulostettiin tiedot:

```java
public class Today {  new *
    public static void main(String[] args) {  new *
        EventProvider provider = new CSVEventProvider( fileName: "events.csv");
        List<Event> events = provider.getEvents();

        if (events.isEmpty()) {
            System.out.println("No events found.");
            return;
        }

        System.out.println("\n=== Events from CSV ===\n");
        for (Event event : events) {
            System.out.println(event);
        }
    }
}
```

Esimerkkitulostus:



Linkki lähdekoodiin:
https://github.com/MikeDanton/5G00EU62-3005-Ohjelmoinnin-syvent-v-t-tekniikat/tree/main/harjoitus5/src/main/java

6 LÄHTEET

https://chatgpt.com/