



Oppimispäiväkirja

Teemu Keskinen



SISÄLLYS

1. Tehtävät.....	2
1.1 Tehtävä 1, viikko 3	2
1.2 Tehtävä 2, viikko 4	3
1.3 Tehtävä 3, viikko 5	4
1.4 Tehtävä 4, viikko 6	5
1.5 Tehtävä 5, viikko 7	6
1.6 Tehtävä 6, viikko 8	8
1.7 Tehtävä 7, viikko 10	9
1.8 Tehtävä 8, viikko 11	11
1.9 Tehtävä 9, viikko 12	12
1.10 Tehtävä 10, viikko 13.....	13
2 Projekti.....	14
2.1 Projekti oppimispäiväkirja.....	14

1. Tehtävät

1.1 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.

```
    Guess the number between 1 and 100:
→ 50

    The number is lower.
    Guess the number between 1 and 100:
→ 25

    The number is higher.
    Guess the number between 1 and 100:
→ 40

    The number is lower.
    Guess the number between 1 and 100:
→ 30

    The number is higher.
    Guess the number between 1 and 100:
→ 35

    The number is lower.
    Guess the number between 1 and 100:
→ 33

    The number is lower.
    Guess the number between 1 and 100:
→ 32

    The number is lower.
    The correct number was 31
```

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

1.2 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 lisätty 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.

```
PS C:\Users\temek\ohjelmoinnin-syventavat\tehtavat> javac Birthday.java
PS C:\Users\temek\ohjelmoinnin-syventavat\tehtavat> Java Birthday
You are 7891 days old
Your next birthday is 145 days away
Your last birthday was 220 days ago
```

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

1.3 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 kirjoitetaan 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

```
PS C:\Users\temek\ohjelmoinnin-syventavat\tehtavat\week 5> javac MacOSRelease.java Event.java Category.java
PS C:\Users\temek\ohjelmoinnin-syventavat\tehtavat\week 5> java MacOSRelease
macOS 15 Sequoia was released on a Monday
macOS 14 Sonoma was released on a Tuesday
macOS 13 Ventura was released on a Monday
macOS 12 Monterey was released on a Monday
macOS 11 Big was released on a Thursday
In alphabetical order: [Big, Monterey, Sequoia, Sonoma, Ventura]
```

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

1.4 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.

```
PS C:\Users\temek\ohjelmoinnin-syventavat\tehtavat\week 6> javac Today.java
PS C:\Users\temek\ohjelmoinnin-syventavat\tehtavat\week 6> java Today --05-25 apple
No events found for the given date and category.
PS C:\Users\temek\ohjelmoinnin-syventavat\tehtavat\week 6> java Today --05-25 oracle/apple
No events found for the given date and category.
PS C:\Users\temek\ohjelmoinnin-syventavat\tehtavat\week 6> java Today --09-25 oracle
2018: Java SE 11 released
PS C:\Users\temek\ohjelmoinnin-syventavat\tehtavat\week 6> java Today --09-20 apple/macos
2016: macOS 10.12 Sierra released
PS C:\Users\temek\ohjelmoinnin-syventavat\tehtavat\week 6> |

PS C:\Users\temek\ohjelmoinnin-syventavat\tehtavat\week 6> java Today --20-5 apple/macos
Invalid date format. Use --mm-dd.
```

Lähdekoodi:

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

1.5 Tehtävä 5, viikko 7

Tehtävässä muokataan githubissa olevia valmiita tiedostoja lisäämällä mavenin avulla projektiin ulkoisen kirjaston avulla Apache Commons CSV. Tämä tapahtuu lisäämällä projektin pom.xml tiedostoon dependency.

```
<dependencies>
  <dependency>
    <groupId>org.apache.commons</groupId>
    <artifactId>commons-csv</artifactId>
    <version>1.9.0</version>
  </dependency>
</dependencies>
```

Tämän lisäksi täytyy tiedostoissa valmiiksi annettua CSV-tiedoston lukemista ja käsittelyä muokata käyttämään uutta lisättyä kirjastoa. Tärkeimmät lisäykset ovat `import org.apache.commons.csv importit`, joilla ulkoiset CSV-tiedoston käsittelyyn käytetyt kirjastot lisätään.

```
import org.apache.commons.csv.CSVFormat;
import org.apache.commons.csv.CSVParser;
import org.apache.commons.csv.CSVRecord;
```

Tämän lisäksi on tiedostoon CSVEventProvider.java nyt lisätty CSVParser lisätyistä kirjastoista, joka koittaa lukea CSV-tiedoston path-muuttujasta, joka sisältää tiedoston sijainnin. Mikäli tiedosto löytyy ja siihen on lukemisoikeus, luetaan jokainen rivi ja kutsutaan event-funktiota ja muokataan rivi tapahtumaksi. Tämän jälkeen tallennetaan juuri luotu tapahtuma listaan ja lopuksi tulostetaan tämä lista pihalle. Myös private event makeEvent funktiota on täytynyt muokkaa, jotta CSVParser toimii oikein. Tänne on tarvinnut muokata syötteen tyyppi CSVRecord tyypiksi, jotta tallennettu CSVRecord-muuttuja voidaan käsitellä String muuttujan sijasta. Tämän jälkeen poistettiin ei tarpeellinen string-split koodista tässä kohtaa, koska record-muuttujassa ne on purettu jo valmiiksi. Today.java tiedostossa on lisätty tiedostoon dynaaminen sijainnin haku ja käytetään Path muuttujaa, jolla voidaan hakea käyttäjän kotikansio today kansio ja tämän kansion sisältä events.csv tiedosto. Tämän lisäksi lisättiin tarkistus, tiedoston löytämiseen. Mikäli tiedostoa ei löydy tai sitä ei saada avattua palautetaan käyttäjälle virhe tästä ja suljetaan ohjelma hallitusti.

```
cd C:\Users\temek\ohjelmoinnin-syventavat\Today; JAVA_HOME=C:\\Users\\temek\\OpenJDK-23 cmd /c "%C:\\Users\\temek\\.vscode\\extensions\\oracle.oracle-java-23.1.0\\nbcode\\java\\maven\\bin\\mvn.cmd" -Dexec.vmArgs=-Dfile.encoding=UTF-8 \"% -Dexec.args=${exec.vmArgs}.classpath %classpath ${exec.mainClass} ${exec.appArgs}\" -Dexec.executable=C:\\Users\\temek\\OpenJDK-23\\bin\\java.exe -Dexec.mainClass=tamk.ohsyte.Today -Dexec.classpathScope=runtime -DskipTests=true -Dexec.appArgs= -Dmaven.ext.class.path=C:\\Users\\temek\\.vscode\\extensions\\oracle.oracle-java-23.1.0\\nbcode\\java\\maven-nb\\lib\\netbeans-eventspy.jar --no-transfer-progress process-classes org.codehaus.mojo:exec-maven-plugin:3.1.0:exec
Scanning for projects...

-----< tamk.ohsyte:Today >-----
Building Today 1.0-SNAPSHOT
from pom.xml
-----[ jar ]-----

--- resources:3.3.1:resources (default-resources) @ Today ---
skip non existing resourceDirectory C:\Users\temek\ohjelmoinnin-syventavat\Today\src\main\resources

--- compiler:3.13.0:compile (default-compile) @ Today ---
Nothing to compile - all classes are up to date.

--- exec:3.1.0:exec (default-cli) @ Today ---
Read 10000 events from CSV file
2022-02-10: Uber introduces real-time data optimization for fleet (test/null)
2022-02-10: Google debuts AI in search for underwater data mapping (test/null)
2021-02-10: Huawei presents AI-based dynamic signal adjustment (test/null)
2020-02-10: Apple reveals hybrid analog-digital input devices (test/null)
2018-02-10: Samsung introduces low-energy e-ink for mobile devices (test/null)
2018-02-10: Intel debuts neural processing units for AR devices (test/null)
2014-02-10: Microsoft rolls out context-aware AI for document editing (test/null)
2014-02-10: Adobe launches real-time translation for graphic design (test/null)
2013-02-10: NASA tests bio-integrated computer systems for space missions (test/null)
2013-02-10: IBM introduces neuromorphic computing prototypes (test/null)
2013-02-10: Adobe reveals VR environment creation tools (test/null)
2010-02-10: Samsung reveals transparent flexible display technology (test/null)
2009-02-10: Samsung introduces low-energy e-ink for mobile devices (test/null)
```

Lähdekoodi:

<https://github.com/TeemuKeskinen/ohjelmoinnin-syventavat/blob/main/Today/src/main/java/tamk/ohsyte/CSVEventProvider.java>

<https://github.com/TeemuKeskinen/ohjelmoinnin-syventavat/blob/main/Today/src/main/java/tamk/ohsyte/Today.java>

1.6 Tehtävä 6, viikko 8

Tämän viikon tehtävässä oli tarkoitus lisätä EventManager-luokka ja toteuttaa sen sisällä olevat add- ja removeEventProvider funktiot. Päätin toteuttaa tämän ratkaisun viime viikolla tekemäni CSVEventprovider päälle, koska tämä koodi oli minulle tuttu verrattuna githubissa tarjottuun openCSV versioon. Lisäsin projektiini valmiiksi tarjotut tiedostot EventManager ja FirstEventProvider sekä SecondEventProvider. EventManager tiedostossa toteutin add- ja removeEventProvider funktiot try-catch avulla, jolloin virheellisessä suorituksessa tulisi edelleen virheilmoitus. Lisäämisen tein this.eventProviders.add(provider) koodilla, joka lisää kutsuttaessa tapahtuman järjestäjän listaan järjestäjistä. Poistamista varten tein boolean muuttujan, jossa removeIf kutsutaan ja tarkistetaan mikäli annettu provider on yhtäsuuri kuin listassa jo olemassa oleva providerId. Tätä testiä toteutetaan provider.getIdentifier().equals(providerId) koodilla, joka vertaa provider muuttujaan tallennettua arvoa getIdentifier kutsulla saatuun providerId. Mikäli nämä ovat yhtäsuuret poistetaan annettu provider listasta. Today.javaan lisäsin vain githubissa olevalla mallilla kutsut, joilla testataan, toimiiko koodi oikein.

```
PS C:\Users\temek\ohjelmoinnin-syventavat\Today> c:; cd 'c:\Users\temek\ohjelmoinnin-syventavat\Today'; & 'C:\Users\temek\OpenJDK-23\bin\java.exe'
'\Users\temek\AppData\Local\Temp\cp_723o3zsiha174tbdnm7u9r02.argfile' 'tamk.ohsyte.Today'
Read 10000 events from CSV file
Event providers: [First, Second, CSV]
Manager has 3 event providers,
with a total of 10002 events.
Event providers: [First, Second]
Manager has 2 event providers,
with a total of 2 events.
```

Lähdekoodit:

<https://github.com/TeemuKeskinen/ohjelmoinnin-syventavat/tree/main/Today/src/main/java/tamk/ohsyte>

<https://github.com/TeemuKeskinen/ohjelmoinnin-syventavat/blob/main/Today/src/main/java/tamk/ohsyte/Today.java>

<https://github.com/TeemuKeskinen/ohjelmoinnin-syventavat/blob/main/Today/src/main/java/tamk/ohsyte/EventManager.java>

1.7 Tehtävä 7, viikko 10

Tässä tehtävässä laajennettiin ohjelman suodatusmahdollisuuksia lisäämällä EventFilter-abstrakti luokka ja sen aliluokat. Näiden avulla voidaan suodattaa tapahtumia päivämäärän ja kategorian perusteella. Ensimmäiseksi luotiin EventFilter-abstrakti luokka, joka määrittää metodin accepts(Event event). Tämä metodi tarkistaa, täyttääkö annettu tapahtuma tietyt suodatuskriteerit.

Tämän jälkeen toteutettiin suodatusluokat DateFilter, joka suodattaa tapahtumat päivämäärän perusteella ja tarvittaessa ottaa huomioon myös vuoden riippuen halutaanko tarkistaa, onko kyseessä singular vai annual tapahtuma. CategoryFilter suodattaa tapahtumat niiden kategorian perusteella. DateCategoryFilter, joka yhdistää molemmat edellä mainitut suodatusehdot ja suodattaa tapahtumat sekä päivämäärän että kategorian mukaan. Näiden lisäksi EventManager-luokkaan lisättiin metodi getFilteredEvents(EventFilter filter), joka palauttaa listan tapahtumista, jotka vastaavat annettua suodatinta. Lopuksi Today.java-tiedostossa testattiin uudet suodatusmetodit lisäämällä esimerkkikutsut DateFilter, CategoryFilter ja DateCategoryFilter -luokkien avulla.

```
Today in history:
2023: IBM introduces neuromorphic computing prototypes (test)
2021: IBM debuts photonic quantum interconnects (test)
2017: Oracle releases advanced predictive analytics toolkit (test)
2015: SpaceX tests miniaturized solar-powered drones (test)

Events filtered by date:
Amazon launches voice-controlled virtual marketplace: test (--03-04)
Intel showcases neural network accelerators for mobile: test (--03-04)
Uber introduces real-time data optimization for fleet: test (--03-04)
Google releases AI-driven search customization tools: test (--03-04)
Apple unveils nanotechnology-enhanced screens: test (--03-04)
Facebook introduces virtual marketplace with AI advisors: test (--03-04)
MIT reveals biologically inspired exoskeleton prototypes: test (--03-04)
Intel debuts neural processing units for AR devices: test (--03-04)
Amazon releases AI for predictive financial modeling: test (--03-04)
Adobe launches AI-generated art customization tools: test (--03-04)
IBM presents quantum parallel processing cores: test (--03-04)
Sony announces eco-powered VR headsets: test (--03-04)
IBM unveils holographic data chips: test (--03-04)
Intel debuts ultra-slim processors for quantum devices: test (--03-04)
Amazon launches AI-augmented shopping experience: test (--03-04)
Microsoft introduces subsonic menu accelerators: test (--03-04)
Intel presents 3D chip stacking for power efficiency: test (--03-04)
Google introduces real-time speech analysis for meetings: test (--03-04)
```

(tässä ei riittänyt visual studio coden sisäisessä terminaalissa muistia näyttää alkua)

```
IBM launches quantum-proof encryption standards: test (--02-22)
Microsoft rolls out context-aware AI for document editing: test (--02-25)
Huawei announces AI-optimized network frequency allocation: test (--02-28)
Sony introduces real-time emotion-responsive gaming: test (--02-29)
NASA announces collaborative AI for international space: test (--03-07)
Google releases AI-driven search customization tools: test (--03-09)
Facebook introduces virtual marketplace with AI advisors: test (--03-10)
Nokia unveils low-power IoT devices for rural areas: test (--03-10)
SpaceX tests miniaturized solar-powered drones: test (--03-10)
Apple releases adaptive AR app development suite: test (--03-13)
Microsoft introduces subsonic menu accelerators: test (--03-13)
```

```
Events filtered by date and category:
Amazon launches voice-controlled virtual marketplace: test (--03-04)
Intel showcases neural network accelerators for mobile: test (--03-04)
Uber introduces real-time data optimization for fleet: test (--03-04)
Google releases AI-driven search customization tools: test (--03-04)
Apple unveils nanotechnology-enhanced screens: test (--03-04)
Facebook introduces virtual marketplace with AI advisors: test (--03-04)
MIT reveals biologically inspired exoskeleton prototypes: test (--03-04)
Intel debuts neural processing units for AR devices: test (--03-04)
```

Lähdekoodit:

<https://github.com/TeemuKeskinen/ohjelmoinnin-syventavat/blob/main/week8/src/main/java/tamk/ohsyte/CategoryFilter.java>
<https://github.com/TeemuKeskinen/ohjelmoinnin-syventavat/blob/main/week8/src/main/java/tamk/ohsyte/DateCategoryFilter.java>
<https://github.com/TeemuKeskinen/ohjelmoinnin-syventavat/blob/main/week8/src/main/java/tamk/ohsyte/DateFilter.java>
<https://github.com/TeemuKeskinen/ohjelmoinnin-syventavat/blob/main/week8/src/main/java/tamk/ohsyte/EventFilter.java>

1.8 Tehtävä 8, viikko 11

Tehtävässä lisättiin toiminnallisuus manuaaliseen tapahtuman lisäämiseen. Tätä varten lisätään CSVEventProvider tiedostoon uusi komento "addevent" joka lisää tapahtuman käyttäjän antamaan .csv tiedostoon. Tämä toteutettiin @command annotaatiolla. @command kutsutaan ensimmäisen kerran CSVEventProviderin alussa, jolla nimetään tulevat komennot. seuraavalla @command annotaatiolla nimetään addevent ja määritetään tarvittavat argumentit. Voidaan nimetä tavat, joilla voidaan kirjoittaa komentorivillä komento. Käytän itse tässä yhtä kokonaan kirjoitettua ja yhtä aliasta, jossa on lyhenteet näille. Komentoriviltä tarvitaan tapahtuman päivämäärä, tapahtuman kategoria, tapahtuman kuvaus ja tiedosto, johon tapahtuma lisätään. Nämä on määritelty vaadittavaksi koodissa lisäämällä required = true @option perään, jolloin nämä tiedot vaaditaan, jotta koodi toimisi oikein. Provider tietokenttä ei ole pakollinen vaan sen voi jättää auki, tällöin se täytetään standard vakionimikkeellä. Tämän jälkeen try-catch avulla yritetään kirjoittaa tietoja csv-tiedostoon FileWriter objektin avulla ja mikäli tämä ei toteudu oikein palautetaan virhe käyttäjälle.

```
public void run() {  
    try (FileWriter writer = new FileWriter(fileName, true)) {  
        writer.append(date).append(",").append(description).append(",").append(category).append("\n");  
        System.out.println("Event added successfully.");  
    } catch (IOException e) {  
        System.err.println("Error writing to CSV file: " + e.getMessage());  
    }  
}
```

```

PS C:\Users\temek\ohjelmoinnin-syventavat\week11> java -jar target/Today-1.0-SNAPSHOT.jar addevent -d 2028-06-14 -c birthday -desc "My 25th birthday" -f c:\Users\temek\..today\new_events.csv
Event provider 'standard' is already registered
Event added successfully.

PS C:\Users\temek\ohjelmoinnin-syventavat\week11> java -jar target/Today-1.0-SNAPSHOT.jar addevent --date 2038-01-19 --category computing/unix --description "Unix clock rolls over" --file c:\Users\temek\..today\new_events.csv
Event provider 'standard' is already registered
Event added successfully.

PS C:\Users\temek\ohjelmoinnin-syventavat\week11> java -jar target/Today-1.0-SNAPSHOT.jar addevent -d --06-14 -c birthday -desc "My 25th birthday" -f c:\Users\temek\..today\new_events.csv
Event provider 'standard' is already registered
Event added successfully.

```

	A	B	C	D
1	1/19/2038	Unix clock rolls over	computing/unix	
2	6/14/2028	My 25th birthday	birthday	
3				
4				

Lähdekoodit:

<https://github.com/TeemuKeskinen/ohjelmoinnin-syventavat/tree/main/week11/src/main/java/tamk/ohsyte>

1.9 Tehtävä 9, viikko 12

Tehtävässä tarvitsi luoda WebEventProvider luokka, joka implementoi EventProvider luokan ja Today.javassa olevan HTTPS kutsun. Tämä kutsu siis kopioitiin WebEventProvider luokkaan ja sen toiminnallisuus kirjoitettiin WebEventCommand luokkaan. Kokonaan Today.javassa olevaa koodia ei tarvinnut kopioida, koska päätin tehdä lisäksi luokan WebEventCommand, jossa pyydän osoitteen, josta haetaan ja sen lisäksi päivämäärän tarkentamaan hakua. WebEventCommand on pieni luokka, jossa alustetaan uusi WebEventProvider ja picocliä käyttämällä pyydetään käyttäjältä tiedot (osoite ja päivämäärä muodossa –MM-DD) ja sen jälkeen ne yhdistetään URI muuttujaksi, joka annetaan WebEventProviderille käytettäväksi. Tämän jälkeen koodi hakee tapahtumat nettisivulta annetun päivämäärän mukaan ja tulostaa ne esille käyttäjälle WebEventProvider luokassa olevalla koodilla. Tätä varten tarvitaan Jackson object mapperia, jonka avulla puretaan JSON tieto, mitä nettisivu palauttaa javan

List <Event> objekteiksi. Tämän jälkeen tiedot tulostetaan käyttäjälle. Tämän lisäksi Today.javaan lisättiin uusi komento Webeventcommand.class.

```
--03-17: Sony introduces real-time emotion-responsive gaming (test/fake)
--03-17: Google debuts spatial search optimization tools (test/fake)
--03-17: Sony introduces real-time emotion-responsive gaming (test/fake)
--03-17: Intel launches AI for precise cancer detection (test/fake)
--03-17: Facebook rolls out autonomous personal data control systems (test/fake)

web-events -s https://todayserver-89bb2a1b2e80.herokuapp.com/api/v1/events?date= --date 03-17
Customize arguments (Press 'Enter' to confirm or 'Escape' to cancel)
```

<https://github.com/TeemuKeskinen/ohjelmoinnin-syventavat/blob/main/week12/src/main/java/tamk/ohsyte/providers/WebEventProvider.java>

<https://github.com/TeemuKeskinen/ohjelmoinnin-syventavat/blob/main/week12/src/main/java/tamk/ohsyte/commands/WebEventCommand.java>

<https://github.com/TeemuKeskinen/ohjelmoinnin-syventavat/blob/main/week12/src/main/java/tamk/ohsyte/Today.java>

1.10 Tehtävä 10, viikko 13

Tehtävässä lisättiin valmiin sqleventprovider-luokan toteutus omaan projektiin. Tämän toteuttamiseksi lisättiin listevents komentoon sql-tietokannan lukeminen osaksi komentoa. Tämän lisäksi tehtävässä kuului lisätä sql-tietokantaan uuden tapahtuman lisääminen. Tätä varten luotiin uusi luokka addevent, jossa käyttäjältä pyydetään päivämäärä, kuvaus ja päivämäärä. Näiden tietojen pohjalta lisätään uusi tapahtuma sql-tietokantaan käyttäen javan sisäistä JDBC-apia, jonka avulla java sovellus pääsee suoraan käsiksi tietokantaan. Uuden tapahtuman lisääminen ja tapahtumien hakeminen tietokannasta tapahtuu sql-kyselyillä samalla lailla, kuin käyttäisi normaalia sql-tietokantaa ja sovellusta. listevents komentoon lisäsin -db vaihtoehtoon, jonka avulla voidaan valita mihin tietokantaan data lisätään, mutta tämä vakioituu events.sqlite3 tiedostoon, jos -

db ei ole annettu komennossa. Viimeisenä lisättiin addevent komento alikomennoksi todayn lisäksi.

```
today addevent -date 2025-03-29 -desc "my test event" -cat 1
```

```
--- exec:3.1.0:exec (default-cli) @ Today ---  
--11-27: Thanksgiving (U.S.A.) (usa/holiday)  
--05-11: itienpiv (Suomi) (finland/flagday)  
--12-10: Second Wednesday in December (test/test)  
Event added successfully.
```

```
Today in history:  
2025: my test event (test/fake)
```

<https://github.com/TeemuKeskinen/ohjelmoinnin-syventavat/blob/main/week13/src/main/java/tamk/ohsyte/commands/ListEvents.java>
<https://github.com/TeemuKeskinen/ohjelmoinnin-syventavat/blob/main/week13/src/main/java/tamk/ohsyte/commands/AddEvent.java>
<https://github.com/TeemuKeskinen/ohjelmoinnin-syventavat/blob/main/week13/src/main/java/tamk/ohsyte/Today.java>
<https://github.com/TeemuKeskinen/ohjelmoinnin-syventavat/blob/main/week13/src/main/java/tamk/ohsyte/providers/SQLiteEventProvider.java>

2 Projekti

2.1 Projekti oppimispäiväkirja

Projekti työnä kurssilla oli toteuttaa viikkotehtävinä annettu kokonaisuus yhtenä ohjelmana, jossa kaikki toiminnallisuudet ovat mahdollisia käyttää samaan aikaan. Projekti kokonaisuutena oli helppo toteuttaa, koska kaikki yksittäiset osat

olivat jo valmiina kurssin aikana annetuissa tehtävissä, joten kokonaisuuden kasaaminen ei kestänyt kovin pitkää aikaa. Projektissa haasteena minulle oli rakentaa universaali tapa lisätä tapahtumia sql ja csv tietokantoihin, jotka valitsin lisäämistä tukeviksi tyypeiksi. Viikko tehtävissä olin rakentanut nämä pelkästään yhteen tiedosto tyyppiin keskittyen, unohtaen katsoa kokonaisuutta, jossa providereita olisi useampia ja tiedosto tyyppejä olisi useampi. Tämän takia jouduin tekemään koodissa uudestaan eventmanageriin `addEventToProvider` kutsun, joka saa tiedot tapahtumasta ja myös providerin tunnuksen, tarkistaa onko kyseinen provider olemassa ja tukeeko kyseinen provider lisäämistä. Tämän jälkeen kutsutaan providerilta `getFilename` ja tallennetaan tämä, minkä jälkeen kutsutaan providerin omaa `addEvent` tapahtumaa tapahtuman tiedoilla ja tiedostonimellä. Lisäsin myös tuen `listproviders` kutsuun hakea pelkästään tapahtuman lisäämistä tukevat providerit antamalla vaihtoehdon `-add` tai `-a listproviders` jälkeen. Tämän avulla voi käyttäjä tarkistaa, mitkä provider mallit tukevat tapahtumien lisäämistä jälkikäteen.

Ohjelma sisältää kirjastot `OpenCSV`, `Picocli`, `Jackson Core` `Jackson Annotations`, `Jackson Databind` ja `SQLite JDBC`. `OpenCSV`, `Jackson` kirjastot ja `SQLite JDBC` tarvitaan tukemaan tapahtumien lukemista eri providereilta. `OpenCSV` tukee CSV tiedostojen lukemista, `SQLite JDBC` tukee `sqlite3` tiedostoja ja `Jackson` kirjastot tukevat netistä lukemista. `Picocli` kirjastoa käytetään komentorivi argumenttien lukemiseen, joita ohjelma käyttää toiminnallisuuden määrittämiseen.

Ohjelma käännetään `bat` tiedoston avulla kutsumalla komentorivillä `today` ja sen jälkeen alikomento. Tästä ja myös ohjelman toiminnallisuudesta todisteet alla

```
C:\Users\temek\ohjelmoinnin-syventavat\project>today listevents
```

```
Today in history:
```

```
2024: Somebody listen cold million herself international. (culture)
2023: Intel debuts ultra-slim processors for quantum devices (test/fake)
2023: Adobe debuts AI-driven text-to-image synthesis (test/fake)
2023: Sony introduces real-time emotion-responsive gaming (test/fake)
2023: Intel debuts ultra-slim processors for quantum devices (test/fake)
2023: Adobe debuts AI-driven text-to-image synthesis (test/fake)
2023: Sony introduces real-time emotion-responsive gaming (test/fake)
2023: Intel debuts ultra-slim processors for quantum devices (test)
2023: Adobe debuts AI-driven text-to-image synthesis (test)
2023: Sony introduces real-time emotion-responsive gaming (test)
2022: Sony introduces real-time emotion-responsive gaming (test/fake)
2022: Sony introduces real-time emotion-responsive gaming (test)
2018: Sony reveals adaptive lighting for virtual concerts (test/fake)
2018: Exactly practice general feeling sound. (health)
2018: Sony reveals adaptive lighting for virtual concerts (test/fake)
2018: Sony reveals adaptive lighting for virtual concerts (test)
2017: Oracle announces self-healing cloud databases (test/fake)
2017: Oracle announces self-healing cloud databases (test)
2016: SpaceX tests miniaturized solar-powered drones (test/fake)
2016: SpaceX tests miniaturized solar-powered drones (test)
2015: SpaceX introduces cargo capsules with AI load balancing (test/fake)
2015: SpaceX introduces cargo capsules with AI load balancing (test/fake)
2015: SpaceX introduces cargo capsules with AI load balancing (test)
2014: Nokia unveils low-power IoT devices for rural areas (test/fake)
2014: Nokia unveils low-power IoT devices for rural areas (test/fake)
```

```
C:\Users\temek\ohjelmoinnin-syventavat\project>today listevents -provider csv -date 06-14 -cat "birthday"
Events for --06-14:
```

```
Printing events for provider: csv
```

```
Today in history:
```

```
2003: my birthday (birthday)
2003: my birthday (birthday)
```

```
C:\Users\temek\ohjelmoinnin-syventavat\project>today listproviders
```

```
csv
sqlite
text
web
```

```
C:\Users\temek\ohjelmoinnin-syventavat\project>today listproviders -a
Listing event provider IDs that support adding events
```

```
csv
sqlite
```

```
C:\Users\temek\ohjelmoinnin-syventavat\project>today addevent -provider sqlite -date 2025-05-31 -desc "Back from military" -cat "test/fake"
Event added to sqlite provider.
C:\Users\temek\ohjelmoinnin-syventavat\project>
```

```
sqlite> select * from event where event_date = "2025-05-31";
10004|2025-05-31|Back from military|1
```


<https://github.com/TeemuKeskinen/ohjelmoinnin-syventavat/tree/main/project>