# Refaktorointiraportti

Sisällysluettelo

[1. Refaktorointiraportti 1](#_Toc531784799)

[2. Bad Smellit OTP-2 kurssilla 2](#_Toc531784800)

[2.1. Long Method 2](#_Toc531784801)

[2.2. Duplicated Code 3](#_Toc531784802)

[2.3. Comments 3](#_Toc531784803)

[3. Suunnittelumallien käyttö 4](#_Toc531784804)

[4. Muuta refaktorointia 4](#_Toc531784805)

[4.1. DAO-Luokille yläluokka 4](#_Toc531784806)

[4.2. Controllereiden yläluokka 5](#_Toc531784807)

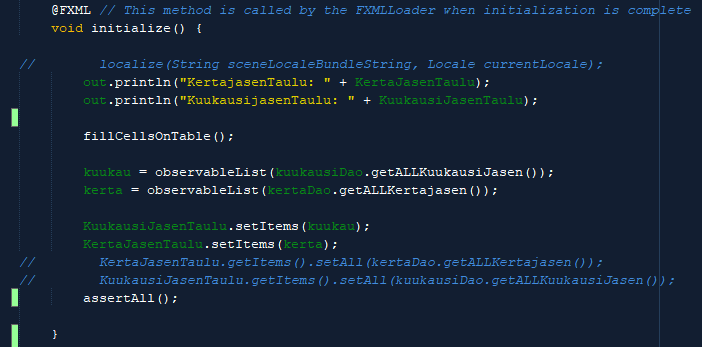
# Bad Smellit OTP-2 kurssilla

# Long Method

Kuva FXMLController-luokan initialize-metodista refaktorointien jälkeen.

Alun perin metodi oli n. 200 riviä pitkä, sisältäen mm. assertit kaikille FXML elementeille ja taulujen cellfactoreiden luonnille/asetuksille.

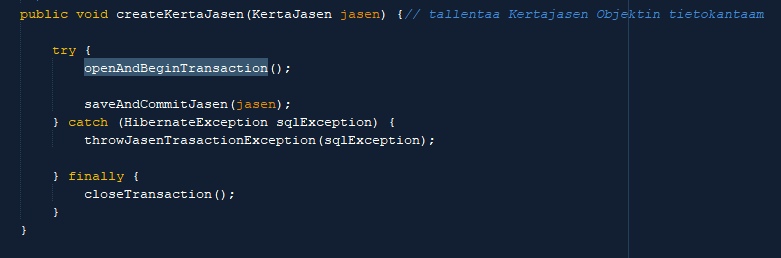
Long Method korjattiin Exctract Method- tekniikalla.



Kuva 1: Refaktoroitu Long Method

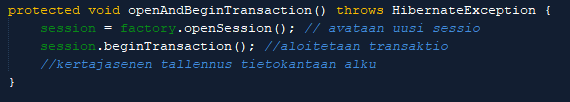
# Duplicated Code

Refactoroitiin Data Access Object (DAO)- luokkien funktiot kompaktimmiksi ja uudelleen käytettäviksi Extract Method- menetelmällä. Tällä korjattiin usein esiintynyttä duplikaattikoodia.



Kuva 2: Refaktoroitua koodia

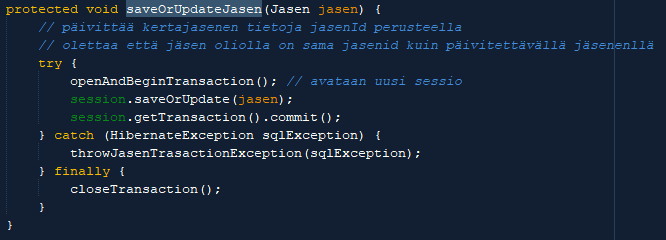
Näin toistuvia koodirakenteita saatiin korvattua yhdellä metodilla.



Kuva 3: Exctracted Method

# Comments

Extractattiin koodista kommentoituja pätkiä, ja luotiin niistä selkeästi nimettyjä metodeita. Tällä saavutettiin parempaa koodin luettavuutta, ja päästiin korvaamaan duplicated codea yksittäisillä metodeilla.



Kuva : Kommentteja metodiksi

# Suunnittelumallien käyttö

# Mediator

Koodissa käytetty paljon kontrollereita scenejen hallintaan. Kontrollerit on toteutettu Mediator-mallin mukaisesti.

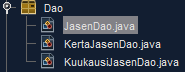
# Singleton

Session Factoryn käyttö hyödyntää Singleton-mallia: Siitä luodaan vain yksi instanssi, jota käytetään koko ohjelman laajuisesti.

# Muuta refaktorointia

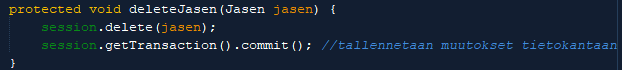
# DAO-Luokille yläluokka

DAO-luokilla oli metodeja, joita voitaisiin käyttää molemmissa JäsenDAO-luokissa. Luokista yhteiset metodit siirrettiin yläluokkaan Pull Up-menetelmällä.



Kuva 5: Exctract Superclassilla uusi yläluokka

Metodit käyttävät jäsen-yläluokkaa, josta voidaan castata kumpi tahansa jäsen-alaluokista tarpeen mukaan.



Kuva 6: JasenDAO:n metodi

# Controllereiden yläluokka

AbstractController-luokka sisältää defaulLocalen, sekä luokkamuuttujan, joka osoittaa jokaiselle scenelle oman bundlen. Lisäksi luokassa on näihin liittyvät metodit.



Kuva 7: Abstrakti yläluokka controllereille



Kuva 8: Abstractin yläluokan käyttö