

Harjoitustyön dokumentaatio COMP.CS.140 Ohjelmointi 3: Rajapinnat ja tekniikat

Tekijät:

Elisabet Dalskaja

Miia Honkanen

Essi Huhtanen

Ohjelman toiminta ja käyttöohje

Ohjelmamme hakee kurssi- ja opintokokonaisuustietoja Sisun API:sta. Opiskelija voi käyttää ohjelmaa seuratakseen ja tallentaakseen etenemistään valitsemassaan tutkinto-ohjelmassa.

Ohjelma koostuu ikkunasta ja kolmesta välilehdestä. Ensimmäisenä avautuu Student-välilehti, johon opiskelija voi antaa nimensä ja id:nsä käyttäessään ohjelmaa ensimmäistä kertaa, tai jos hän on jo tallentanut tietojaan, ID:n antamalla ohjelma palauttaa opiskelijan tallentamat tiedot (*lisäominaisuus*). Ohjelma kertoo, jos opiskelija on tallentanut tietojaan mutta on silti lisäämässä nimeään uutena opiskelijana ("Existing student!"). Toisaalta jos opiskelija yrittää hakea ID:llä tietojaan, mutta niitä ei ole tallennettu, ohjelma ilmoittaa "Student not found!". Student-välilehden alareunassa näkyy opiskelijan sen hetkinen suoritus tilanne (*lisäominaisuus*) ja muut opiskelijan tiedot. Kun opiskelija käyttää ohjelmaa ensimmäistä kertaa ja tallentaa siihen nimensä ja ID:nsä, ohjelma muistuttaa opiskelijaa tallentamaan tietonsa alareunan Save progress –painikkeella.

Kun opiskelijan tiedot on syötetty ohjelmaan (joko uutena opiskelijana tai aikaisemmasta tallennuksesta), Degree Program –välilehti avautuu. Tältä välilehdeltä opiskelija voi valita oman tutkinto-ohjelmansa napsauttamalla ohjelman nimen vieressä olevaa radiopainiketta ja vahvistamalla valintansa Choose degree program –painikkeella. Valitun ohjelman nimi ilmestyy yläreunaan, jonne tulee myös uusi muistutus opiskelijalle tietojen tallennuksesta. Tietojen lataamisessa voi kestää hetki tutkinnon laajuudesta riippuen.

Kun tutkinto on valittu ja ohjelma on hakenut kurssitiedot Sisun API:sta, Modules and courses –välilehti avautuu. Välilehdellä on kaksi toimintoruutua. Vasemmanpuoleisesta ruudusta opiskelija voi moduuleita klikkaamalla selata tutkinnon kurssirakennetta. Rakenteessa näkyvät tutkinnon alla olevat opintomodulit ja kurssit. Nimen perässä näkyy kurssin tai moduulin vähimmäislaajuus opintopisteissä. Opintopisteiden minimimäärä oli selkeä valinta rajalliseen tilaan, ja tämän vuoksi laajuutena on joissakin moduuleissa 0 cr. Kun opiskelija valitsee moduulin, jonka alla on kurseja, kurssit tulevat näkyviin oikeanpuoleiseen ruutuun. Tästä ruudusta opiskelija voi täpätä suorittamansa kurssit. Kun opiskelija tallentaa valinnat painamalla Save changes –painiketta, suoritettut kurssit näkyvät vasemmanpuoleisessa ruudussa vihreällä värillä korostettuna ja "(Completed)"-tekstillä varustettuna. Samalla myös eteneminen päivittyy alareunaan ja kurssien ja muiden modulien nimien perään. Kurssista saa lisätietoa pitämällä hiirtä kurssin nimen päällä oikeanpuoleisessa ruudussa (*lisäominaisuus*).

Opiskelija voi tallentaa etenemisensä missä välilehdessä tahansa Save progress -painikkeella. Kun painiketta on napsautettu, se menee harmaaksi, kunnes opiskelija tekee uusia muutoksia kurssitietoihinsa. Opiskelija

voi halutessaan vaihtaa tutkinto-ohjelmaansa Degree program –välilehdessä koska vain. Muutokset tallentuvat aina edellisten päälle.

Toteutetut lisäominaisuudet

Toteutimme nämä Ohjelman toiminta ja käyttöohje -osiossa kuvatut lisäominaisuudet:

- 1) Kurssin katselunäkymä. Vietäessä hiiren kursori jonkin kurssin yläpuolelle, kyseisestä kurssista näytetään lisätietoja näytöllä.
- 2) Opiskelija-asetukset. Ohjelman käynnistyksen yhteydessä avataan asetusikkuna, josta voi asettaa ohjelmaa käyttävän opiskelijan tiedot. Suorituksen tilanne haetaan ja tallennetaan JSON-tiedostoon.
- 3) Opiskelijan etenemisen aikataulun esittäminen tutkinnon rakenteeseen peilaten.

Ryhmän työnjako

Alla olevassa taulukossa on kuvattuna suurpiirteinen työnjakomme. Pidimme aloituspalaverin, jossa suunnittelimme ohjelman rakennetta yhdessä. Kävimme tiivistä keskustelua ryhmätyön aikana ja tarjosimme ja saimme toisiltamme (ja myös kurssihenkilökunnalta) apua työn eri vaiheissa. Pysyimme sovitussa ja kaikki tekivät parhaansa.

Elisabet Dalskaja	Tietojen haku Sisun API:sta DegreeBuilder-luokalla, DegreeBuilder-luokan yksikkötestaus
Miia Honkanen	Tutkinto-ohjelmien rakennetta kuvaavien luokkien Sisumodule, DegreeProgramme, Specialisation, StudyModule, GroupingModule ja Course toteutus, opiskelijaluokan Student toteutus ja opiskelijatietojen tallennusluokan DegreeSaver toteutus sekä luokkien yksikkötestaus
Essi Huhtanen	käyttöliittymän toteutus App ja PrimaryController -luokkien avulla sekä TestFX-testaus

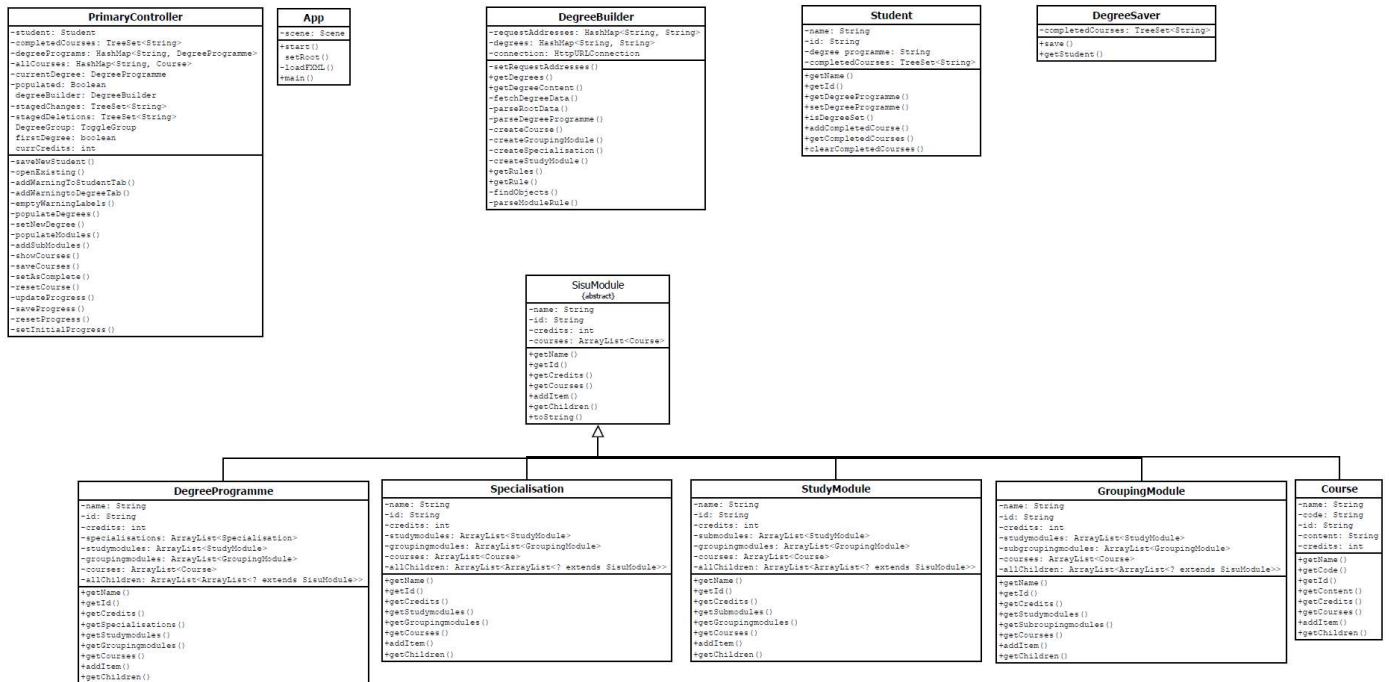
Luokkarakenne ja luokkien vastuujako

Seuraavassa kaaviossa on kuvattuna suunnittelemamme luokat ja niiden rakenne.

Yläreunassa ovat käyttöliittymäluokat PrimaryController ja App, sekä API:n lukemiseen tarvittava DegreeBuilder-luokka. Niiden vieressä on opiskelijan tiedot säilyttävä Student-luokka, ja tiedot erilliseen JSON-tiedostoon kirjoittava ja niitä lukeva DegreeSaver-luokka.

Alempana kuvatut luokat DegreeProgramme, Specialisation, StudyModule, GroupingModule ja Course periytyvät abstraktista luokasta Sisumodule. Käsittelimme kaikenlaisia opintokokonaisuuksia ja kursseja Sisumodulen aliluokkina, minkä vuoksi jouduimme ratkaisussamme toteuttamaan Course-luokassa muutaman dummy-funktion.

Kaaviossa on kuvattu vain moduulien periytymissuhde, mutta kaikki projektin luokat liittyvät toisiinsa ja tarvitsevat toisiaan. Tällaiset assosiaatiomerkinnot kaaviosta kuitenkin puuttuvat.



Ohjelmassamme moduuliolioihin tallennetaan kunkin tyyppisten opintokokonaisuuksien nimet, ID:t ja opintopisteet, sekä kurssin osalta koodi ja kurssin sisältö. Moduulien hierarkia noudattaa Sisun rakennetta, eli ylemmän tason oliot sisältävät säiliöitä (ratkaisussamme ArrayList) alemman tai saman tason olioille.

Moduuliluokkien lisäksi toteutimme Student-luokan, joka pitää sisällään opiskelijan tunnistetiedot ja tallennetun tutkinto-ohjelman ja suoritetuiksi merkityt kurssit.

DegreeSaver-luokka sisältää json-tiedostojen kirjoittamisen ja lukemisen mahdollistavat funktiot.

DegreeBuilder-luokka on vastuussa Sisun API:n lukemisesta ja käsittelee luettua dataa kahdella eri tavalla: metodia getDegrees kutsuttaessa luokka palauttaa vain tutkinnot eli DegreeProgramme-oliot säiliörakenteessa, ja metodia getDegreeContent kutsuttaessa luokka täydentää valitun tutkinnon eli yksittäisen DegreeProgramme-olion tiedot.

PrimaryController-luokassa toteutetaan itse käyttöliittymä, joka hyödyntää muita luokkia.

Tiedossa olevat ongelmat tai puutteet

Abstraktista SisunModule-luokasta periytyvien metodien dummytoteutukset Course-luokassa eivät ole tyyllisesti paras mahdollinen ratkaisu. Ongelman olisi voinut ratkaista esimerkiksi käyttämällä ParentModule-rajapintaluokkaa, jota implementoivat vain Course-luokkaa ylemmät opintomodulit.

Eri moduulien luokkien metodeja olisi voinut testata laajemminkin, mutta niiden samankaltaisuuden vuoksi testit on suoritettu vain osalle moduuleista. Samoin DegreeSaverin IO-toimintojen testit puuttuvat. Lisäksi luokan DegreeBuilder sisäisesti kutsuttavat metodit on jouduttu asettamaan julkisiksi yksikkötestejä varten, mikä luo turhia metodeja julkiseen rajapintaan.

Käyttöliittymän käytettävyyttä olisi vielä voitu parantaa monin tavoin, joista tutkintojen sekä kurssien hakutoiminto olisi helpottanut järjestelmän käyttöä huomattavasti.

Käyttöliittymässä kertaalleen tehdyn etenemisen tallennuksen jälkeen opinto-ohjelman vaihtaminen mitätöi opiskelijan aiemmin valitussa tutkinnossa suoritetuiksi merkityt kurssit. Järjestelmästä jäi puuttumaan mahdollisuus lisätä opintoja oman tutkinto-ohjelman ulkopuolelta.