

Vesa Lukkarila
Mikko Juntunen
group2735

[Luokkien vastuujako, kts kaavio dokumentin lopussa.](#)

Ohjelmassa on valmiiksi annettu abstraktista kantaluokasta DegreeModule periytetty ActualDegreeModule, GroupingModule, StudyModule ja CourseModule. Kyseiset luokat ovat keskenään samanlaisia ja periaatteessa yksi toteuttava luokka olisi riittänyt tämän kokoiseen ohjelmaan, mutta jatkokehitysmahdollisuuksien vuoksi jokaiselle moduulille päätettiin luoda oma luokka. Sisu-luokka toimii sekä View että Controller roolissa (MVC-rakenteen termin ilmaistuna). RootSearch.java tekee pyynnöt Sisu API:lle heti ohjelman käynnistyessä: hakee tiedot kaikista tutkinto-ohjelmista ja luo näistä oliot. iAPI.java taas käsittelee käyttäjän valitseman tutkinto-ohjelman ja luo kustakin moduulista olion.

Projektin loppupuolella tekijöille selvisi mitä ohjeistuksessa mainittu MVC rakenne todella tarkoittaa, mutta ohjelman koon ja jo tehtyjen ratkaisujen vuoksi sekä View- että Controller-roolit valittiin jättää Sisu-luokalle, sillä rakenteen muokkaaminen MVC:n mukaiseksi olisi ollut lähinnä muodollista ja ehkä hiukan teennäistäkin.

[Ohjelman toiminta ja käyttöohje](#)

Ohjelman toiminta on yksinkertainen. Kun sovellus avataan, hakee ohjelma automaattisesti kaikki vuoden 2023 tutkinto-ohjelmat RootSearch-rajapinnan palveluiden avulla. Paluuarvona tulee lista olioita, jotka asetetaan vasemmalle välilehdelle dropdown-valikkoon.

Dropdown-valikosta käyttäjä valitsee tutkinnon, jonka sisältöä haluaa tarkastella. Kun käyttäjä on tehnyt valinnan käy ohjelma iAPI-luokan palveluita käyttäen rekursiivisesti läpi valitun tutkinnon rakenteen ja luo olion kustakin moduulista. iAPI-luokan findModules on depth-first algoritmi ja lisää kunkin lapsimoduuliolion vanhemman säiliöön ennen seuraavaan lapsimoduuliin siirtymistä.

Oikealle välilehdelle avautuu siten TreeView-näkymä valitun tutkinnon rakenteesta. Tutkinnon osia voi tarkastella avaamalla ja sulkemalla moduuleja. Kaikista moduuleista on näkyvissä nimi, mutta vain kurssimoduuleista opintopisteet. Ohjelmaan ei ole tehty toteutettu opiskelijan opintosuunnitelman tallentamismahdollisuutta ja tämän vuoksi ei myöskään erillistä sisäänkirjautumisenäkymää. Valittua tutkintoa voi vaihtaa valitsemalla dropdown-valikosta toisen tutkinnon ja uuden tutkinnon näkymä avautuu taas automaattisesti oikealle välilehdelle. Välilehtien sulkeminen on estetty ja ohjelman oikeaan alakulmaan on ankkuroitu "Quit"-nappi, joka liikkuu ikkunan kokoa muutettaessa.

[Sovittu ja toteutunut työnjako](#)

Ohjelma suunniteltiin toteutettavaksi siten, että Vesa tutustuu json-tiedostojen parsimiseen, Mikko taas web-pyyntöjen tekemiseen Sisu-apille ja luokkien toteuttamiseen. Käyttöliittymää muokattiin annetusta pohjasta yhdessä hieman erilaiseksi, tällä tavoitellaan kolmosen

arvosanaa. Asioita tehtiin läpimatkan yhdessä ja toisilta apua kysellen, mutta projektin luonne oli sellainen, että oli helpompaa yrittää hoitaa omat osiot yksin kuin käyttää aikaa toisen briiffaamiseen. Projekti tehtiin käytännössä viimeisen kahden viikon aikana, gitin commitit viimeisen viikon aikana, kun erikseen tehdyt osiot saatiin viimein sovitettua yhteen toimivaksi kokonaisuudeksi. Projekti onnistui siis kutakuinkin ennalta sovitulla työnjaolla.

Tiedossa olevat ongelmat ja puutteet

- Ohjelma olisi pitänyt tehdä MVC rakenteen mukaiseksi, mutta tämä tyyli oli tyystin uusi tekijöille ja tieto asiasta tuli niin myöhään, ettei muutoksiin enää ruvettu.
- Testitiedostot
- Virhetarkistukset, jos kori.api:lta tulee jokin virhekoodi niin niihin olisi voinut tarttua.

Ohjelman rakenne, kaavio

