

Tekijät (Group2823):

- Pekka Nokelainen, 150290744
- Sampo Suokuisma, 150422473
- Anni Vierimaa, 150292771

Luokkien vastuunjako:

Graafinen käyttöliittymä:

- MainGUI.java
 - Kysyy käyttäjältä opiskelijanumeron ja nimen. Hyödyntää jsonManageria, joka tutkii, onko kyseistä opiskelijaa vielä tallennettuna json-tiedostoon. Jos on, kutsuu InfoGUI:ta oikeilla tiedoilla. Jos ei, kutsuu NewStudentGUI:ta.
 - Ei päästä opiskelijaa eteenpäin, jos opiskelijanumerossa on muita kuin numeroita tai jos nimessä muita kuin kirjaimia.
- NewStudentGUI.java
 - Pyytää käyttäjää valitsemaan pudostuslaatikosta tutkinto-ohjelman. Hyödyntää ShowContents-luokkaa.
- InfoGUI.java
 - Hyödyntää ShowContents-luokkaa tutkinnon rakenteen näyttämässä ja esittää sen käyttäjälle.
 - Esittää valitun moduulin kurssit, sekä luo niille valintalaatikot. Hyödyntää Student-luokan attribuutteja.
 - Esittää tutkinnon etenemisen suhteessa vaadittuihin opintopisteisiin.
- Sisu.java
 - Koko ohjelman main-funktio, käynnistää MainGUI:n.

Muu toteutus:

- CourseUnit.java
 - Luokka, johon talletetaan kurssit ja kursseihin liittyvät muut tiedot, kuten nimi, id, ryhmä id, opintopisteet ja tieto suorituksesta.
- DegreeModule.java
 - Luokka, johon talletetaan nimi, id, ryhmä id, opintopisteet, alikurssit ja alimoduulit. Luokan perii DegreeProgramme.java, GroupingModule.java ja StudyModule.java, joissa kaikissa on sama toteutus. Kaikki luokat myös rakentuvat json tiedostosta ja niiden tietoja voidaan muuttaa tai hakea metodien avulla.
- DegreeModuleFactory.java
 - Luokkia lajitteleva ja tekevä luokka, jonka avulla yhden DegreeProgramme olion antamalla voidaan rakentaa koko tutkinto rakenne moduulien luokilla. Luokka ja kurssi rakenne luodaan rekursiivisesti eri luokkia kutsuen. Sisältää myös metodin, jonka avulla voidaan luoda linkkejä ja hakea json muodossa tietoa SisuAPI luokan kautta. Sisältää lisäksi myös haku kutsuja muille luokille, jotka tarvitsevat olioina kaikki luokat tai kurssit.

- jsonManager.java
 - Toteuttaa rajapinnan iReadAndWriteToFile.javan. Luokan avulla kirjoitetaan ja luetaan StudentData.json tiedostoa, mihin on talletettu opiskelijoiden valinnat.
- ShowContents.java
 - Linkittää Treeltemit toisiinsa InfoGuin TreeViewia varten. Sisältää myös alikurssien ja alimoduulien haku metodit nimien avulla.
- SisuAPI.java
 - Luokka, jonka avulla haetaan tietoa linkkien avulla sisusta. Palauttaa tiedon json muodossa.
- Student.java
 - Luokka, johon tallennetaan opiskelijoiden tiedot aina nimestä ja opiskelijanumerosta suoritettuihin kursseihin ja valittuun opintokokonaisuuteen.
- StudentDeserializer.java
 - Luokka, joka avustaa jsonManageria lukemaan tallennetusta json-tiedostosta opiskelijoiden tiedot ja luomaan tarvittavat Student-oliot.
- StudentSerializer.java
 - Luokka, jonka avulla jsonManager pystyy kirjoittamaan json-tiedostoon opiskelijoiden tiedot Student-olioista, kun ikkuna suljetaan.
- TreeModule.java
 - Luokka, jonka avulla saadaan kaikki DegreeProgrammet API-pyyntöjen avulla. API-pyyntö suoritetaan DegreeModuleFactoryn kautta.
 - Selvittää käyttäjän valitseman opinto-ohjelman ja käynnistää sen avulla tutkinto-ohjelmaa vastaavien moduulien, sekä kurssien luonnin kutsumalla DegreeModuleFactory-luokkaa.
- Testiluokat
 - Testaavat eri luokkien rakentajia ja muita get- tai set-toimintoja. Ei testaa GUI luokkia tai niihin liittyviä toimintoja. Pää paino opintorakenteen testaamisessa ja tiedostoon oppilaan tietojen kirjoittamisessa.
- iAPI.java
 - Julkinen rajapinta tiedon keräämiseksi Sisusta API-pyyntöillä.
- iReadAndWriteToFile.java
 - Julkinen rajapinta, jonka avulla voidaan kutsua json tiedostoon kirjoittavia metodeja, jotka tallentavat oppilaiden tiedot StudentData.json tiedostoon käyttöliittymän sulkeuduttua ja lukevat tiedot StudentData.json tiedostosta, kun käyttäjä kirjoittaa nimensä ja oppilasnumeron.

Sovittu työnjako:

Jaettavat osiot:

Pekka Nokelainen:

- URL:in ja API:n käyttö, sekä näiden tietojen muokkaaminen ohjelmassa hyödynnettävään muotoon
- JSON-tiedostojen lukeminen ja kirjoittaminen

- Tutkinto-ohjelmien, moduulien ja kurssien puurakenteen tekeminen

Sampo Suokuisma:

- Opiskelijan tietojen lukeminen opiskelijan tiedot json- tiedostosta sekä tallentaminen sinne

Anni Vierimaa:

- Graafisen käyttöliittymän teko, eri näkymien ohjaaminen ja toiminnallisuuksien toteuttaminen

Kaikki:

Testaus ja testiluokkien tekeminen

Dokumentaatio

Toteutunut työnjako:

Jaettavat osiot:

Pekka Nokelainen:

- URL:in ja API:n käyttö, sekä muokkaaminen ohjelmassa hyödynnettävään muotoon
- Tutkinto-ohjelmien, moduulien ja kurssien puurakenteen tekeminen
- Testiluokkien tekeminen ja ohjelman testaaminen
- CourseUnit.java, DegreeModule.java, DegreeModuleFactory.java, DegreeProgramme.java, GroupingModule.java, iAPI.java, SisuAPI.java, StudyModule.java, ShowContents.java, iReadAndWriteToFile.java

Sampo Suokuisma:

- Opiskelijan tietojen lukeminen tiedostosta sekä tallentaminen sinne
- TreeModule.java, ShowContents.java, jsonManager.java, Student.java, StudentDataJson.json, StudentSerializer.java, StudentDeserializer.java

Anni Vierimaa:

- Graafisen käyttöliittymän teko, eri näkymien ohjaaminen ja toiminnallisuuksien toteuttaminen
- MainGUI.java, NewStudentGUI.java, InfoGUI.java, Sisu.java

Kaikki:

Javadoc-kommenttien tekeminen omiin luokkiin, sekä yhteisesti dokumentaatio

Käytännössä teimme koodia ja siihen liittyvää suunnittelua jonkin verran yhdessä, sekä teimme työtä eri luokkien välillä.

Käyttöohje:

Ensimmäisessä näkymässä käyttäjän tulee syöttää integer-tyyppinen opiskelijanumero, sekä opiskelijan nimi tekstikenttiin, sekä painaa "Search"-nappia. Mikäli opiskelijan tiedot on tallennettu aiemmalla käyttökerralla, avautuu suoraan opintotietonäkymä. Jos opiskelijaa ei ole tallennettu aiemmin, avautuu uusi näkymä, jossa opiskelijan tulee valita pudotusvalikosta tutkinto-ohjelma, sekä klikata "Save new student"-näppäintä.

Tämän jälkeen avautuvassa opintotietonäkymässä on kolme pystysuuntaista osiota: vasemmanpuoleisin näyttää puurakenteena valitun tutkinto-ohjelman, sen alamoduulit ja niiden sisältämät kurssit. Kun jokin alamoduuli avataan laajemmaksi, keskimmäiseen osioon avautuu sen moduulin sisältämät kurssit, sekä moduulin nimi (puurakenteena). Jokaisen kurssin nimen viereen tulee valintalaatikko, josta yksittäisen kurssin tai koko moduulin (moduulin nimen kohdalta) voi rastittaa suoritetuksi. Nämä tallentuvat opiskelijan suorittamiin kursseihin. Suorituksen voi myös poistaa laatikkoa klikkaamalla, mikä poistaa suorituksen opiskelijan tiedoista. Oikeanpuoleisin kapea osio näyttää tutkintoon vaaditut opintopisteet, opiskelijan sillä hetkellä suorittamat opintopisteet, sekä edistymispalkin perustuen näihin lukuihin.

Jokainen näkymä sisältää "Quit"-nappulan, jota klikkaamalla ohjelman suoritus päättyy. Opintotietonäkymässä on lisäksi "Change programme"-nappula, josta opiskelija voi siirtyä valitsemaan tutkinto-ohjelmansa uudelleen. Tämä toiminto uudelleenkirjoittaa opiskelijan suorittamien kurssien tiedot.

Tiedossa olevat ongelmat ja puutteet:

Jotkin moduulit sisältävät useita samannimisiä kursseja, esimerkiksi "Ammattiharjoittelu- tekniikka", jotka eroavat toisistaan vain niiden opintopisteiden perusteella. Kun moduulin kursseja käydään läpi valintalaatikkoja varten, käydään kurssit läpi loopilla. Kurssin nimeä etsitään opiskelijan suoritettuihin kursseihin tallennetusta listasta, ja sen mukaan valitaan, tuleeko laatikkoon rasti jo valmiiksi vai ei. (Onko rasti valittu aiemmalla käyttökerralla). Tässä kohtaa looppi löytää saman kurssin listasta jokaisella loopin kierroksella ja näyttää, että kaikki sen nimiset kurssit on suoritettu, vaikka todellisuudessa suorituksia voi olla vähemmän. Tämä ongelma havaittiin liian myöhään, jotta sen olisi voinut korjata järkevästi tietorakennetta muokkaamalla.

Mikäli StudentData.json-tiedosto sisältää korruptoitunutta tekstiä, ohjelma ei saa käsitellä sitä oikein. Ohjelma heittää JsonSyntaxErrorin, jota yritettiin korjata JsonSyntaxExceptionilla, sekä MalformedJsonExceptionilla, siinä kuitenkaan onnistumatta. Lopulta päätimme luottaa siihen, ettei käyttäjällä ole pääsyä StudentDataJson-tiedostoon (kuten ei todellisuudessa ole).

Toteutetut lisäominaisuudet:

- Opiskelija-asetukset. Ohjelman käynnistyessä asetusikkunaan laitetaan opiskelijan tiedot ja jos opiskelija tiedot löytyvät JSON-tiedostosta hänen suorituksensa avataan suoraan oikein.
- Opiskelijan opintojen etenemisen aikataulun esittäminen tutkinnon rakenteeseen peilaten oikeassa reunassa.
- Ylimääräisiä, hyvin toteutettuja yksikkötestejä
- Erittäin hyvä ja nopea käyttöliittymä
- Laadukkaasti dokumentoidut rajapinnat ja luokat
- Tutkinto-ohjelmaa valitessa tutkinto-ohjelmasta näytetään teksti, kun hiiren kursori viedään sen päälle.