

### Project: Sisu Unraveled

#### I – Project description

SISU is a program for handling degree data to get an easy way to keep track of where students in Tampere University are at regarding completing a degree.

##### 1. *Project functionality*

- Only English interface is implemented in this version.
- When the users open Sisu application, a first dialogue view will appear for logging in or signing up.
- A student information tab view is implemented for use after the users have logged in: Users can view their own information. In this view, they can also choose or change their degree programme.
- Only degrees that are still available for students (whose minimum credits are not 0) are fetched for users to choose.
- The users' progress over their studies can be viewed from the student information tab view.
- The users can mark courses completed as studies go forward.
- When the users select a course that has not been completed from the curriculum view, its details about completion methods will be shown.

##### 2. *Project classes*

- Key classes and their responsibilities:
  - **DegreeModule:** An abstract class for storing information of modules or courses.
  - **DegreeProgramme:** A class representing the root level of a degree structure.
  - **StudyModule:** A class representing a study module, which usually consists of courses or submodules.
  - **GroupingModule:** A class representing an intermediate level in the degree structure, which consists of courses or submodules and is superficially similar to a study module. These differ from study modules mainly in that grouping module do not have information about study credits.
  - **Course:** A class representing a single course with data like code, name, awarded study credits.
  - **Data:** A class for retrieving data from the API and processing data to store and return degree module or course objects.
  - **JsonReaderWriter:** A class for writing to or reading from a JSON file to store or read the progress and information of a student. These files are stored at `../Sisu/students`.
  - **Student:** A representing student data. During the planning, the class keeps track of the user's actions to store the degree program and completed courses.
  - **StartDialogueController** and **SelectionViewController:** Classes for controlling the flow and processing data during the use of Sisu application.
- Classes are also documented using javadocs. The documentation files are available in `../Sisu/target/site/apidocs`.
- The UML class diagram of our application is as below (The diagram can be found at `Documentation/class-diagram.jpg`):

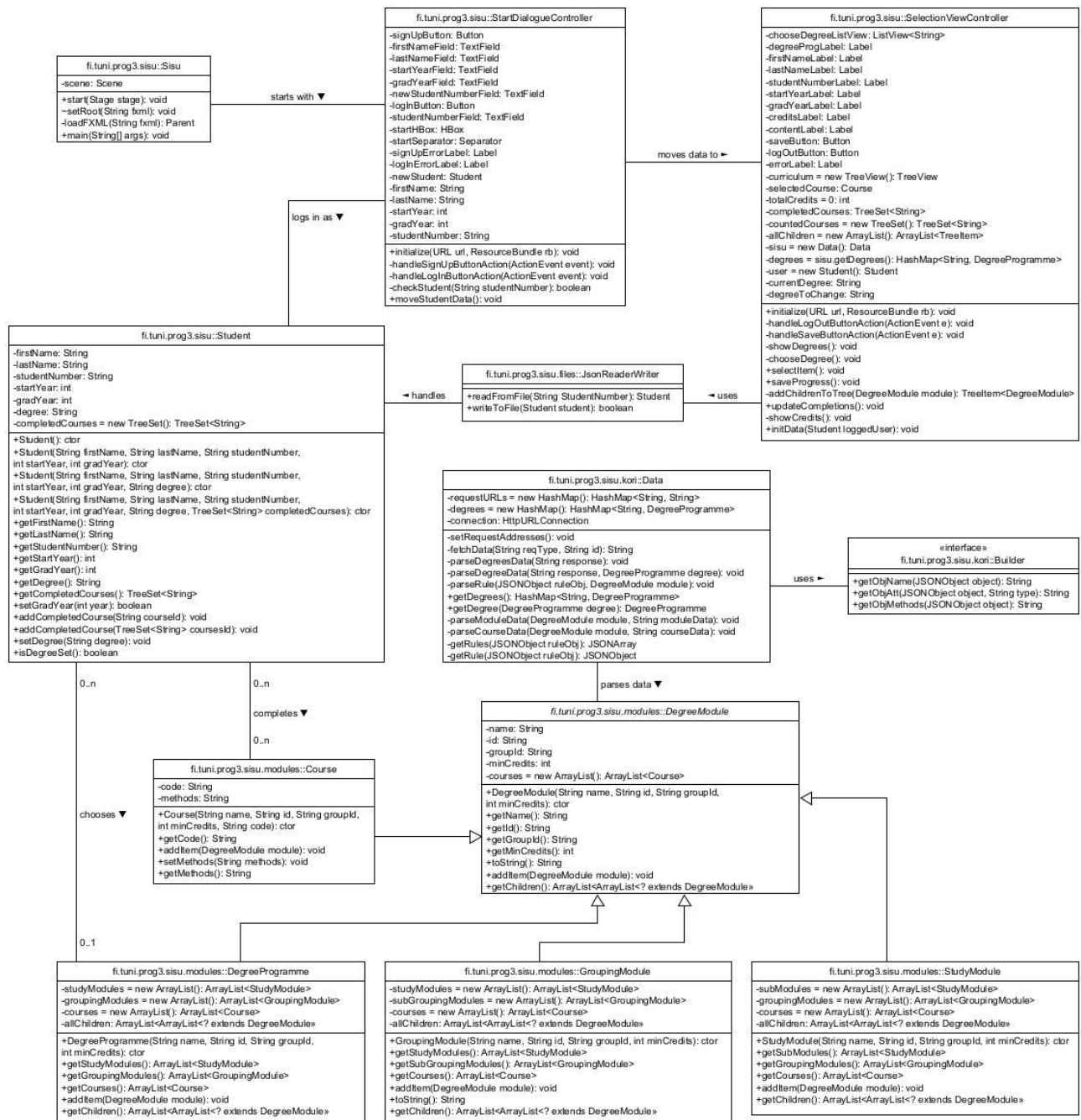


Figure 1: Class diagram for program structure

## II – Group members and workshare

Member 1: Thuy Le, [thuy.t.le@tuni.fi](mailto:thuy.t.le@tuni.fi), 150533634, responsible for:

- Implementing classes `DegreeProgramme`, `GroupingModule`, `StudyModule`, and `Course` (using the abstract class `DegreeModule`) to handle degree structures. Implementing class `Student` to handle information of users in role of a student.
- Implementing unit tests for `fi.tuni.prog3.sisu.modules` package.
- Fetching data from Kori API and getting degree modules or courses using fetched data by building class `Data` and interface `Builder`.
- Combining the two views for the application after logging in.
- Updating `README.md` and writing the documentation for the project.

Member 2: Elsa Seppänen, [elsa.seppanen@tuni.fi](mailto:elsa.seppanen@tuni.fi), H299913, responsible for:

- Setting up the first dialogue interface of the application.
- Setting up the student information view for the application after logging in.
- Writing user manual for the application.

Member 3: Eero Jormalainen, [eero.jormalainen@tuni.fi](mailto:eero.jormalainen@tuni.fi), H292260, responsible for:

- Reading and writing to a file with class `JsonReaderWriter` to save or read the progress of each student.
- Implementing unit tests for `JsonReaderWriter` class.
- Setting up the curriculum view for the application after logging in.

### III – User manual

In the first dialogue, the user can either sign up as a new user by entering their information or log in using their student number. After logging in or signing up, the user can choose or change their degree programme in the student information tab by clicking the degree programme's name and after that clicking "Save". The user can also look at their personal information there. In the curriculum tab the user can see the courses in their degree programme and add completed courses to their credits. After adding the completed courses and clicking "Save progress", the user can see their credit progress in the student info tab. In the curriculum tab, the user can also see the course completion methods by selecting a course that has not been completed yet. The user has to log out using the button in the student information tab before closing the window to save the user for the next log in.

### IV – Further improvements

- The application assumes that students will enter the valid information when signing up. In the future, information validation will be implemented.
- The application now does not allow entering the grades for the courses. When this feature is implemented, students can see their GPA in the student settings.
- The application now does not allow dropping a course after completing it. Therefore, if students accidentally save a course as completed, they cannot undo that action. The feature of dropping a course will be allowed in later versions.
- The user info cannot be edited in the student information tab. Adding this feature would allow the user to edit the expected graduation year.