The Student Management App, part 1

Gradle homework

Important note

- This homework will be used by several subsequent homeworks and is a baseline and every subsequent one will upgrade on the previous one
- If required to submit in a subgroup in git, **use** a single group, like `/student-management/` because all the subsequent homeworks will be in the same repo and can be differentiated by branches or tags.
- Advice: write it well, so that when you upgrade it you won't have a bad experience

Overview

The task for this homework assignment is to create a multi-project repository that build with gradle. The expected projects that you need to have are:

- ☑ ***Student Management CLI***
- ☑ ***Student Management Business Layer (a.k.a. services)***
- ☑ ***Student Management DAL (Data Access Layer) (a.k.a. repositories)***
- ☑ ***Student Management Domain Model (a.k.a. entities, business objects, java beans)***

Hint: use `gradle init` - it will greatly decrease your suffering.

Projects Overview

There will be four projects for the entire solution.

- ☑ ***Student Management CLI***
  - this project will handle user interaction via the command line interface. It will serve as a ***presentation layer*** for the application, which means it will just handle input/output and user interaction.
  - this project will ***only*** use the ***Business Layer*** (service methods) defined in the ***Student Management Business Layer*** and the objects defined in ***Student Management Domain Model***.
- ☑ ***Student Management Business Layer (a.k.a. services)***
  - this project should implement ***services***, which will define the business logic in the application, which will be used in the ***presentation layer***.
- ☑ ***Student Management DAL (Data Access Layer) (a.k.a. repositories)***
  - ***database layer***, which should define interface through which the domain models can be persisted in a *** data structure***, ***file/s***, or a real ***database***.
  - you're not required to make all of the above, but extending/implementing the interface for a specific data storage should be an easy process.
  - ***CRUD*** (create, read, update, delete) functionality in the form of ***repositories***.
- ☑ ***Student Management Domain Model***
  - this project should include the ***domain models*** - the objects used throughout the project

  - the object should not have functionality except for one that handles relationships between the object themselves

Core business logic functionality (functional requirements)

As a user I should be able to:

- add a new course.
- add a new student.
- add a new teacher.
- add/remove a teacher to a specific course (max 1).
- add a student to a specific course.
- add a grade for a student in a specific course. (grade 2.0-6.0)
- show all students grouped by course (alphabetically) and then by their average grade for the course (ascending).
- show all courses and their teachers and students (without grades).
- show the average grade for all students in a specific course.
- show a total average grade for a student (between all of his courses using the average grade).

Models

```
Teacher
- id
- name
- degree (MSc, BSc, PHD)

Student
- id
- name
- age

Course
- id
- name
- total hours
```