Object-Oriented Programming – Exercises Week 5

This week, we are going to create two different applications based solely on UML diagram.

The learning goal for the exercises

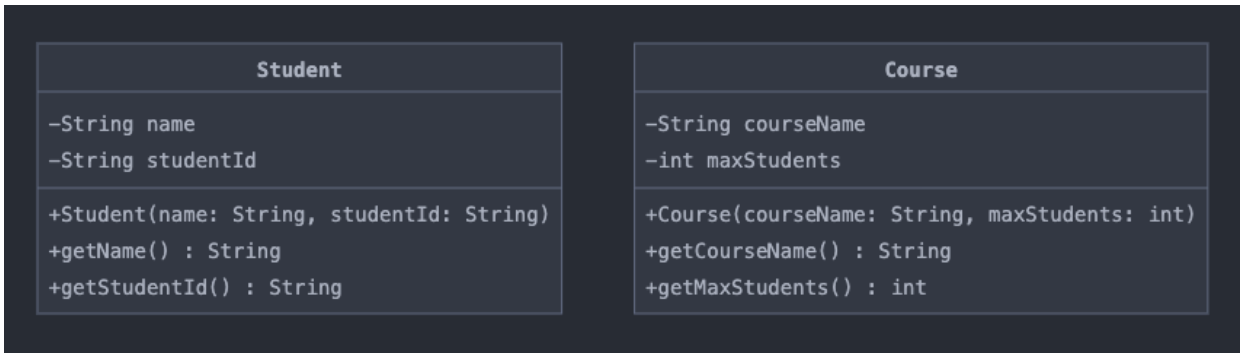
1. Creating program code from UML
2. Refreshing memory from previous exercises. In the end, repetition makes you perfect!

Note on submissions:

This week, we are testing out new ways to submit via CodeGrade.
You must submit the separate .java files to CodeGrade. No .zip files this week.

Exercise 1 – Students on a course (1 point for full implementation)

Implement the Java code according to the following UML diagram. This is such a simple implementation that we only create students and the courses, no need to implement the relationship this time.



Class descriptions

Student.java

- The student's attributes are all private. The name and student ID are set in the constructor.
- The student has two getter methods: getName(), which returns the student's name, and getStudentId(), which returns the student's ID.

Course.java

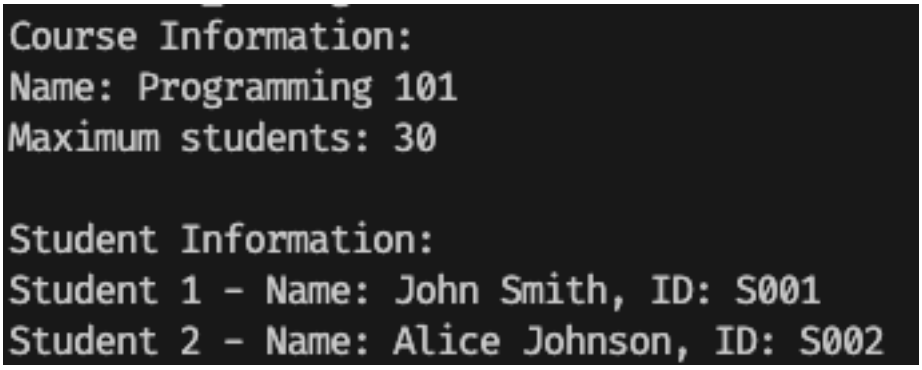
- The course's attributes are all private. The courseName and maxStudents are set in the constructor.
- The course has two methods: getCourseName(), which returns the name of the course, and getMaxStudents(), which returns the maximum amount of students on that course.

Main.java

- This is the main program that is started. It will create the courses and the students and then print them on the command line
- For this exercise, no user input is required
- NOTICE: We are checking to ensure that the required Course.java, Course.java, and Main.java are included in the code. In addition, CodeGrade will check that all the methods exist in the classes
- All code is checked for the printing machine. If you just try to print the expected output, your code will not be accepted.

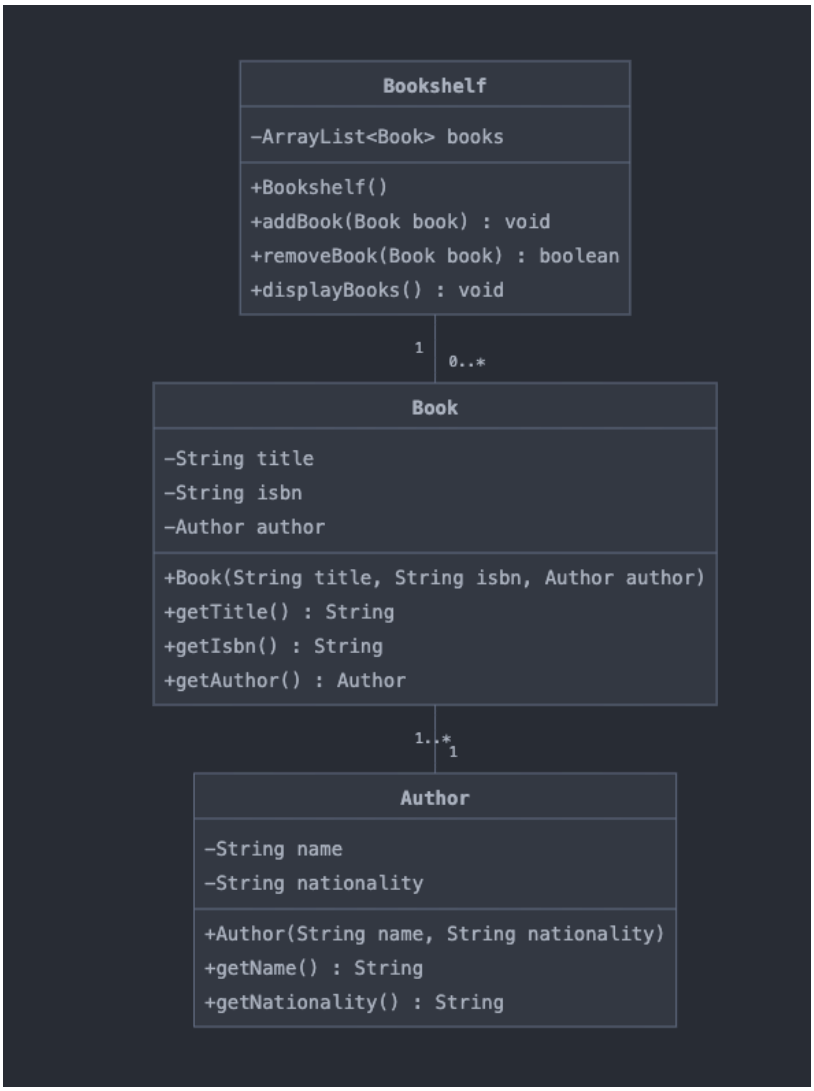
Expected output

The expected output when running main.Java is the following:



Exercise 2 – Book on a shelf (1 point for full implementation)

Implement the Java code according to the following UML diagram:



Class descriptions

Author.java

- The author's attributes are all private. The name and nationality are set in the constructor.
- The author has two getter methods: getName(), which returns the author's name, and getNationality(), which returns the author's nationality.

Book.java

- The course's attributes are all private. The values are set in the constructor.
- The book only has getter methods for private attributes.

Bookshelf.java

- The Bookshelf has a private list of books on the shelf.
- The bookshelf has methods for adding books to the shelf and removing books from the shelf.
- The bookshelf has a method for displaying the books on the shelf.

Main.java

- This is the main program that is started. It will create the courses and the students and then print them on the command line
- For this exercise, no user input is required
- NOTICE: We are checking that the required .java files are included in the code. In addition, CodeGrade will check that all the methods exist in the classes
- All code is checked for the printing machine. If you just try to print the expected output, your code will not be accepted.

Expected output

The expected output when running main.Java is the following:

```
Initial bookshelf contents:
Books on the shelf:
Title: 1984, Author: George Orwell, ISBN: 978-0451524935
Title: Animal Farm, Author: George Orwell, ISBN: 978-0451526342

Removing 1984...

Updated bookshelf contents:
Books on the shelf:
Title: Animal Farm, Author: George Orwell, ISBN: 978-0451526342
```