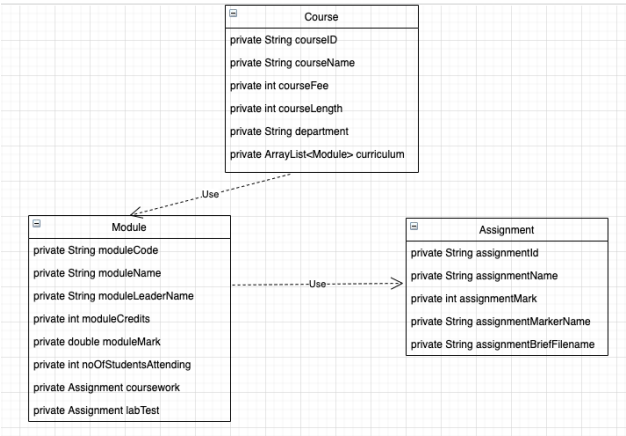
4.40. Lab practical: Conditional (if) statements, iteration, collections, and using classes from the Java library

This is a pair-programming task. The aim of this practical is to become familiar with conditional (if) statements, iteration, collections, and using classes from the Java library. The tasks are based on a variation (and a partial version) of the class diagram from previous labs.

Tasks

General advice: *Make sure you compile your code regularly to make sure you catch syntax errors early.*



1. Using your project from last week's lab (decide which one to use within your pair) and the class diagram above:

i) Create a new class `Course` to match the class diagram;

ii) Add the respective fields to the `Course` class as they appear in the class diagram.

2. Constructor: write a constructor that initializes all fields of the `Course` class to reasonable values. Make sure that you initialize the `ArrayList` to an appropriate value!

3. Getter (accessor) and setter (mutator) methods: Add the appropriate getter (accessor) and setter (mutator) methods for all fields. *Do you need a setter for the `ArrayList`?*

4. Add a method to the `Course` class: `public void addModule(Module module)`, which works according to the following requirements:
If the `module` parameter is equal to `null`, the method rejects the parameter and prints an appropriate error message.
If the `ArrayList` already contains the `module` parameter object, the method rejects the parameter and prints an appropriate error message.
Otherwise, the method adds the `module` parameter object to the `ArrayList`.

Hint: Refer to the `ArrayList` documentation to discover how you can check whether an `ArrayList` contains an element.
https://docs.oracle.com/en/java/javase/19/docs/api/java.base/java/util/ArrayList.html

5. Add an accessor method to the `Course` class: `public void printCourseDetails()`, which prints the details of all fields along with some descriptive text. For the `curriculum` field, the method should print the details of all `Module` objects contained in the `ArrayList`. If the `ArrayList` is empty, then the method should print an appropriate message.

Hints: *1. Refer to the `ArrayList` documentation to discover how you can check whether an `ArrayList` is empty.*
2. Use a loop (e.g. a for-each loop) to iterate over the `ArrayList`.

6. Add an accessor method to the `Course` class: `public void printAssignmentsByMark(int mark)`, which prints the details of all assignments with a mark greater or equal to the `mark` parameter.

Hint: *Use a loop (e.g. a for-each loop) to iterate over the `ArrayList`.*

Make sure you share your code with your partner before leaving the lab!
This is a very important lab practical, so make sure you complete all tasks during this week's lab sessions!

Good luck!