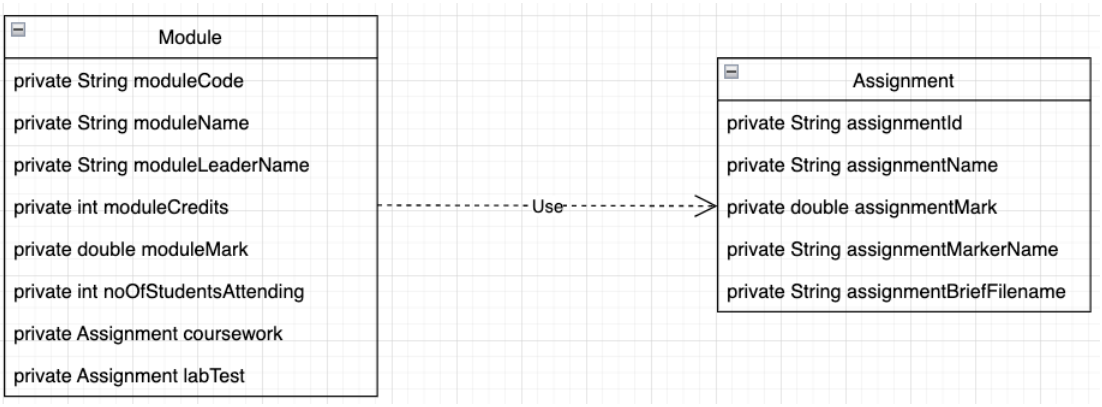
## 3.20. Lab practical: Conditional (if) statements, constructors, methods, classes using classes, and calling methods from the String class

This is a pair-programming task. The aim of this practical is to become familiar with conditional (if) statements, constructors, methods, classes using classes, and calling methods from the String class. The tasks are based on a variation (and a partial version) of the class diagram from previous week's lab.

### Tasks

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




**1.** Using the class diagram above:

**i)** Create a new project in BlueJ;

**ii)** Create new classes to match the class diagram;

**iii)** For each class, add the respective fields as they appear in the class diagram.

**2.** *Constructors*

**`Assignment` class:** The constructor takes no parameters and sets all fields to default values – 0.0 for `double` fields and the zero-length string ("") for `String` fields.

**`Module` class:** The constructor takes two parameters of type `Assignment`, which are used to set the value of the `coursework` and `labTest` fields, respectively. Furthermore, the constructor sets all other fields to default values – zero (0) for integer fields, 0.0 for `double` fields and the zero-length string ("") for `String` fields.

**3.** Getter (accessor) methods: Add the appropriate getter (accessor) methods for all fields in both `Assignment` and `Module`.

**4.** Setter (mutator) methods: Add the appropriate setter (mutator) methods for all fields in both `Assignment` and `Module`, according to the following requirements:

**`Assignment` class:**

- `assignmentId` must be at least six (6) characters long
- `assignmentMark` cannot be negative

**`Module` class:**

- `moduleCode` must be five (5) characters long
- `moduleCredits` and `noOfStudentsAttending` cannot be negative

*Hint: use if/if-else statements. In all cases, print appropriate error messages (and leave the respective field values unchanged) if the parameter values do not comply with the above requirements.*

**5.** Add a mutator method to the `Module` class: `public void calculateModuleMark()`, which calculates the module mark by combining the marks of the two `Assignment` fields and storing the result to the `moduleMark` field.
You may assume that both assignments have equal weighting; hence, you have a number of options here – decide within your pair.

**6.** Add an accessor method to the `Module` class: `public void printModuleDetails()`, which prints the details of all fields along with some descriptive text. For the `moduleCode` field, the method should print *"Module code not set!"* if its value is equal to the zero-length string.

**Make sure you share your code with your partner before leaving the lab!**

**Good luck!**