Programming Exercises - PRO1 - Session 11

Exercise 11.01

Modify your `MyDate` class so that you:

a) Add an equals method that will return true if the object given as argument has the same values for day, month and year. The method signature should be:
`public boolean equals(MyDate obj)`
b) Add a copy method, `public MyDate copy()`, that will return a new `MyDate` object with the same date (day, month and year)
c) Add a copy constructor, `public MyDate(MyDate obj)`, that sets the fields to the same values as those in the `MyDate` object given as argument

Then write a test class to test the new constructor and methods.

Exercise 11.02

Add a new method `nextDays(int days)` to your `MyDate` class. The method is similar to the `nextDay()` method you have made previously, but instead of only increasing the date by one day, it should increase the date by the number of days given as argument.

Exercise 11.03

Add a static method named `today()` to your MyDate class. The method should return a MyDate object that is always set to the current date.

To find out what date it is today you can use Java's built-in `LocalDate` class.
So, first import that class:

```
import java.time.LocalDate;
```

Then you can get the current day, month, and year like this:

```
LocalDate currentDate = LocalDate.now();
int currentDay = currentDate.getDayOfMonth();
int currentMonth = currentDate.getMonthValue();
int currentYear = currentDate.getYear();
```

When you are finished, then also modify the no-argument constructor in the MyDate class, so it always sets the date to the current date.

Exercise 11.04

Add yet another method `isBefore(MyDate date2)` to your `MyDate` class. The method should return `true` if the date is before the other date given as argument, and if not, then the method should return `false`.

Exercise 11.05

[Gaddis] Programming Challenges 1, p. 463-464

[Gaddis] Programming Challenges 4, p. 465