Assignment 2: Constructors, String methods, static variables and methods

Exercise 2.1

Write a class which describes the characteristics of a rectangle in 2D coordinate system (sides are parallel to the coordinate axis). It contains instance data for the lower left corner coordinates and upper right corner coordinates (four double values). Provide next constructors:

- Constructor with no parameters, which initializes the lower left corner as (0, 0) and upper right corner as (1, 1)
- Constructor, which takes the upper right corner coordinates as parameters and the lower left corner is initialized as (0, 0)
- Constructor, which takes the lower left corner and the upper right corner coordinates as parameters

Write a method, which returns the width of the rectangle, and a method, which returns the height of the rectangle. Write also a method, which returns the area of the rectangle (use methods for width and height).

Write a main method which instantiates some rectangle objects using your constructors. Print out the width, height and area of those rectangles.

Exercise 2.2

a) Write a class, which represents a date. It contains instance data for the day, month, and year (type int). Use access modifier private for the instance variables. Write getters and setters (Hint: you get getters and setters if you choose **Source/Generate Getters and Setters** in Eclipse).

Write also a method

```
public String toString()
```

which returns the date as a String in a form dd.mm.yyyy, for example 25.10.2019 (or if you prefer yyyy/mm/dd).

Write a short main, which instantiates an object, sets some data to it, and prints out its data with help of toString method.

(Note: In standard Java API, there are already several classes for dates, for instance java.util.Date, java.util.Calendar, and java.time.LocalDate. We use our own date class for learning purposes, and we'll continue to develop this class as follows....).

b) Write a constructor with no parameters, which initializes the date as today using system date. You get the system date using java.time.LocalDate:

```
java.time.LocalDate today = java.time.LocalDate.now();
day = today.getDayOfMonth();
month = today.getMonthValue();
year = today.getYear();
```

c) Constructor, which takes a day, a month and a year as integer parameters.

d) Constructor, which takes whole date as a String parameter in the form ”dd.mm.yyyy” (or if you prefer, in form yyyy/mm/dd) and the constructor splits it and sets the instance variables. (You can split a string using, for example, methods **`split`** or **`substring`**, and you can change a string to **`int`** type using, for example, method **`Integer.parseInt(String intasstring)`**).

c) Method **`public boolean isSame(YourDate another)`**, which checks if this date object represents the same date as another date (use **your own class type** for the parameter type).

d) Write a main method which instantiates some date objects using these constructors. Ask at least one date from the user as a string. Compare some of the dates with your isSame method.

e) Add an array, which is common to all objects of this class, into your date class. This array contains the names of the months:

```
public static final String[] MONTHNAMES = {"January", "February",
"March", "April", "May", "June", "July", "August",
"September", "October", "November", "December"};
```

Write an **instance method**, which returns the date in a long form, for example "the 10th of January, 2018"

Write a **static method** (**class method**), which gets a month name as a parameter, and the method returns the number of the month. The method should work correctly regardless of whether the month name is in lower case or upper case letters.

Add statements to your main method to test these new methods.