Exercises, SDJ1

Exercise 2.01

Create a new module in IntelliJ and name it `Exercise2_01`. Write an Application that reads your name and age and displays it on the screen (the method `nextLine()` on a `Scanner`-object is used for reading a line of text and method `nextInt()` is used for reading integers). Name the class `MyNameAndAge`

Exercise 2.02

Create a new module in IntelliJ and name it `Exercise2_02`. Write an Application that reads your name, age and address and displays it on the screen (almost like the previous exercise). Name the class `MyPersonalData`

Note: If the application is not waiting for you to type in the address, then consult Code Listing 2-31 (code line 27 and the comments after the code listing) – p. 118-119

Exercise 2.03 – String

Create a module in IntelliJ with the name `Exercise2_03` and create a class with a `main` method, `StringTest` in which you do the following in the `main` method:

a) Create a `String` variable storing a line you read from keyboard
b) Use String method `length` to print out the length of the string
   Hint: calling method `length` on a string variable named `line` could be done the following way:
   ```
   int lineLength = line.length();
   ```
c) Use String method `charAt` to print out the 1st character (at index 0)
   Hint: calling method `charAt` on a string variable named `line` could be done the following way:
   ```
   char firstCharacter = line.charAt(0);
   ```
d) Use String method `charAt` to print out the last character
   Hint: First, find the index of the last character (with the use of the line length)
e) Use String method `toUpperCase` to print the string in all uppercase letters
f) Use String method `toLowerCase` to print the string in all lowercase letters
g) Use String method `substring` to print the first 5 characters
h) Use String method `substring` to print the last 5 characters
i) Use String method `endsWith` to print if the last part ends with "`abc`"
j) Use String method `indexOf` to print out the index of "`a`" – is it what you expected?
k) Use String method `indexOf` to print out the index of "`b`" – is it what you expected?
l) Use String method `indexOf` to print out the index of "`X`" – is it what you expected?
m) Use String method `replace` to replace '`a`' by '`X`' and print out the result.

Run the program (at least) four times, with the following keyboard input

- First run: "`abcde ghijk`"
- Second run: "`abc abc`"
- Third run: "`aaaaaaaaaabbbbbbbbbb`"
- Fourth run: "`123456789aab123456789`"

Exercise 2.04

a) Download the file http://ict-engineering.dk/jar/RentalCompany.jar to a folder on your computer (if you don't already have it from last session)

b) Create a new module in IntelliJ with the name `Exercise2_04`

c) Setup IntelliJ to get access to the downloaded file, the following way:
   → File (top menu) → Project Structure…
   → Libraries → [Right-click the library `RentalCompany`] → Add to Modules…
   → [Choose the Module] `Exercise2_04` → OK → OK

d) Create a class called `DateTest` with a `main` method.

e) In line 1 (before: `public class DateTest`), write the following statement:
```
import rentalcompany.model.*;
```

f) Read 3 integers from keyboard, and store them in the variables: `day`, `month` and `year`.

g) Create a `MyDate` object with the integers read from keyboard
   Hint: to create a `MyDate` object for the date 22/12/2012, do the following:
```
MyDate date1 = new MyDate(22, 12, 2012);
```
   Note: use variables instead of literals (hard-coded values)

h) Print out the `MyDate` object using `toString`

i) Read one more integer (in a variable named `days`) representing how many days you want to step forward.

j) Call method `stepForward(days)` on the `MyDate` object and print out the date again.
   Hint: to call a void method (not returning anything) you cannot store it in a variable or put it in a print statement, instead do something like the following:
```
date1.stepForward(17);
```

Exercise 2.05

Create a new module in IntelliJ and name it `Exercise2_05` and setup the module to get access to the file `RentalCompany.jar` like in the previous exercise. Create a class with a `main` method (name the class `EngineTest`) and insert the statement in top of the file (before the `main` method):
```
import rentalcompany.model.*;
```

The purpose for this exercise is to create an object of a class `Engine` storing information of a car engine (with type, horsepower, volume and a boolean variable storing the information if it is a diesel engine or not)

Example: An engine for a Porche 911:
`Type = 6-cylinder, twin-turbo`
`Horsepower = 420`
`Volume in m³ = 3000`
`isDiesel = false` (it is a petrol engine)

Among other methods, the class `Engine` has a constructor (to be used when creating an object) and a `toString` method (to be used when getting all information in a single string). The two methods are used the following way:
```
Engine engine1 = new Engine(type, horsePower, volume, true);
System.out.println("Engine 1: " + engine1.toString());
```

```
Engine engine2 = new Engine(type, horsePower, volume, false);
System.out.println("Engine 2: " + engine2.toString());
```

Do at least the following in the body of the `main` method:

a) Read a String from keyboard (using `nextLine`) and store it in a variable named `type`.
b) Read an `int` from keyboard (using `nextInt`) and store it in a variable named `horsePower`.
c) Read an `int` from keyboard (using `nextInt`) and store it in a variable named `volume`.
d) Create an `Engine` object using the 3 variables and the value `false` for the last parameter.
e) Print out the object using method `toString`

Exercise 2.06

Create a new module in IntelliJ and name it `Exercise2_06`

a) Create a new class `Rectangle` and copy the entire contents of class `Rectangle` given below
b) Create another class `RectangleTest` with a `main` method (in the same module such that you now have two classes in the src folder). In the body of the `main` method do the following:
    1. Read a `double` from keyboard representing the length of a rectangle.
    2. Read another `double` from keyboard representing the width of the rectangle.
    3. Create a `Rectangle` object using the two values
    4. Print out the length of the `Rectangle` object, using method `getLength`
    5. Print out the width of the `Rectangle` object, using method `getWidth`
    6. Print out the area of the `Rectangle` object, using method `getArea`
c) Run the program a few times entering some decimal numbers for length and width.
    Note: Depending on your Window settings, you either have to use dot (.) or comma (,) as a separator for the decimal number.
    Either: `Please enter length: 2.5`
    Or: `Please enter length: 2,5`

```
public class Rectangle
{
   private double length;
   private double width;

   public Rectangle(double length, double width)
   {
      this.length = length;
      this.width = width;
   }

   public void setLength(double length)
   {
      this.length = length;
   }

   public void setWidth(double width)
   {
      this.width = width;
   }

   public void set(double length, double width)
   {
      this.length = length;
      this.width = width;
   }

   public double getLength()
```

```
    {
       return length;
    }

    public double getWidth()
    {
       return width;
    }

    public double getArea()
    {
       return length * width;
    }
}
```