MY FIRST JAVA CLASSES
LAB 2

1. Objectives:

- Write my first simple Java classes
  - Rectangle and Circle class
- Controlling access to members of a class (*private* / *public* keywords and no keywords)
- Understanding the difference between Instance and Class members (*static* keyword*)*
- Understand the use of the *final* keyword with variable.
- Use Math library

2. Lab Practice:

2.1. The “ Rectangle ” class

- As described in “LAB 1”, create a new Java Project in Eclipse and name it: **Rectangle.**
- Add a new **Rectangle** class to this project. The Rectangle class defines a rectangle geometric shape with specific size. It provides methods to compute the area and perimeter.
- The **Rectangle** class contains:
  - A *public* attribute representing the height of the rectangle (type: double).
  - A *public* attribute representing the width of the rectangle (type: double).
  - A constructor to create instances of Rectangle with specific size.
  - A *public* method calculating and returning the area of the rectangle.
  - A *public* method calculating and returning the perimeter of the rectangle.
- Write the **Rectangle** class and the required comments to generate the corresponding Javadoc documentation. Generate the Javadoc.
- Test your Rectangle class by writing a simple program that creates a Rectangle instance and print its height, width, area and perimeter on the standard output (add a new **TestRectangle** class - implementing the **main** method - to the Rectangle project).
- Run the **TestRectangle** program.
- Complete the **TestRectangle** program to change the height and width of the **Rectangle** object after its creation. The program should now prints the object's height, width, perimeter and area just after its creation and after user has modified its height/width properties.
- Create a new class “`RectangeWithFinalKW`” by modifying the **Rectangle** class adding the ***final*** keyword to height and width attributes.
  - What do you observe?
  - What would be the interest of using the *final* keyword in our `RectangeWithFinalKW` class?

- Create a new class “`RectangeWithPrivateKW`” by Modifying the **Rectangle** class. Modify the **Rectangle** class to make the height and width *private* attributes.
- Run the **TestRectangle** program again. What do you observe? Explain.
- What is missing to the `RectangeWithPrivateKW` class? Please complete the Rectangle class with missing elements and run **TestRectangle** again.

- Create a new class “`RectangeWithNoKW` ” by modifying the **Rectangle** class, modify the **Rectangle** class to make the height and width attributes with not access modifier
- Instantiate `myRectangle5` from `RectangeWithNoKW` class

```
RectangeWithNoKW myRectangle5 = new RectangeWithNoKW(3,5);
        myRectangle5.height = 6;
        myRectangle5.width = 7;
```

- Run the **TestRectangle** program again. What do you observe? Explain what is happening

2.2. Rectangle with UID

- We want to assign a unique ID that can't be changed to each rectangle object created with the **Rectangle** class. Propose a solution based on the usage of a class member (static keyword).
  - Create a `RectangleFull Class,` that will assign an unique UID (1,2,…) for each instance –
    - use 2 variables
      - a class variable: cnt
      - an instance variable: uid
  - Instantiate `myRectangle6` from `RectangleFull` class
  - Modifiy the TestRectangle as follow:

```
            RectangleFull myRectangle6 = new RectangleFull(3,5);
            System.out.println("rec6 cnt= "+myRectangle6.getRectCnt()+" uid=
"+myRectangle6.getUid());
            RectangleFull myRectangle7 = new RectangleFull(3,5);
            System.out.println("rec7 cnt= "+myRectangle7.getRectCnt()+" uid=
"+myRectangle7.getUid());
            System.out.println("rec6 cnt= "+myRectangle6.getRectCnt()+" uid=
"+myRectangle6.getUid());
```

Output expected:

```
rec6 cnt= 1 uid= 1
rec7 cnt= 2 uid= 2
rec6 cnt= 2 uid= 1
```

2.3. The “ Circle” class

- Create a new Java Project in Eclipse and name it **Circle.**
- Write the **Circle** class and **TestCircle** program, following the **Rectangle** example developed in previous section.
    - In the Circle Class add:
        - Instance variable: radius (private)
        - a method to get the area
        - a method to get the perimeter
        - The Circle class should define or use the constant PI.

Create a circle of radius 3.0 from `Circle class` and print its characteristic:

```
“My new Circle is created with radius X and have an area of Y and a
perimeter of P”
```

- Write a new Circle class: **CercleWithMath** with using Math methods/constant
- For the area use “Pow” formula.

Create a circle of radius 3.0 from `CircleWithMath class` and print its characteristic:

```
“My new Circle is created with radius X and have an area of Y and a
perimeter of P”
```