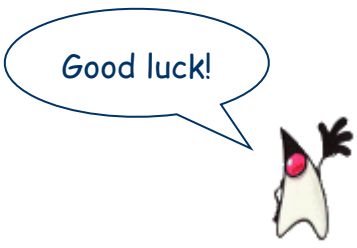
Mittuniversitetet
MID SWEDEN UNIVERSITY

Assignment 3

Java for C ++ programmers, 7,5 hp

Objective:	To practice throw, catch and handle exceptions in Java.
To read:	Lecture 4
Tasks:	1
Submission:	Inlämningslåda 3 at Moodle



Task 1

In this task, you will start with your solution from Lab2.
Create a new Java project in Eclipse. Your classes should belong to the package: `dt062g.studentid.assignment3` where studentid is your username in the student portal / Moodle.
The program should be completely text-based (use System.in and System.out for input and output from / to the user).
NOTE! Inputs and outputs may only be made in the class containing the main method (unless otherwise stated in the text).

Each class and interface (interface) that you create must be documented with comments. Ex.

```
/**
* A short description (in Swedish or English) of the class.
*
* @author  Your Name (your student id)
* @version 1.0
* @since   yyyy-mm-dd (last edited)
*/
```

Rectangle and Circle

The following methods from assignment 2 returned the value -1 if, for example, circumference could not be calculated:

- Shape
  - `getCircumference`
  - `getArea`
- Rectangle
  - `getWidth`
  - `getHeight`
  -
- Circle
  - `getRadius`

Instead of returning -1, these methods now should throw an exception if the data requested can not be calculated. Create an exception class, a separate class inherited from `java.lang.Exception` that will throw exception if the data cannot be calculated because the endpoint of the figure is not available. Give an appropriate error message that you can convey to the user.

In the classes `Rectangle` and `Circle` you need to make changes in the method `toString` method to catch and handle all the expected errors.

Assignment3

You will find this class attached to the task description. The class demonstrate the use of the classes from assignment 2. That is, to check if a calculation cannot be performed, the return value is -1.

Your task is to adjust the class `Assignment3` so that it may use the try-catch method to handle the exceptions. You have to make changes only in `printArea` method. Do not forget to change the package declaration and information about the author, version and date.

Below is an example of printout while running the test program. Your printout should look exactly same before and after the changes in the class `Assignment3`.

```
Drawing a newly created rectangle...
Drawing a Rectangle[start=0.0, 0.0; end=N/A; width=N/A; height=N/A;
color=blue]

Printing area of a Rectangle
The area cannot be calculated, end point is missing!

Changing end point of rectangle to 5.0, 5.0...
Printing area of a Rectangle
The area is: 25.0

Drawing a newly created circle...
Drawing a Circle[start=5.0, 5.0; end=N/A; radius=N/A; color=black]

Printing area of a Circle
The area cannot be calculated, end point is missing!

Changing end point of circle to 8.0, 9.0...
Printing area of a Circle
The area is: 78.5
```