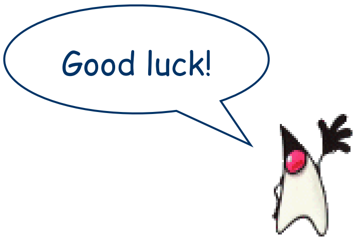
Mittuniversitetet
MID SWEDEN UNIVERSITY

Assignment 7

Java for C ++ programmers, 7,5 hp

Objective:	To create own graphical components inherited from JPanel and use classes in Java to draw 2D graphics.
To read:	Lecture 8
Tasks:	1
Submission:	Inlämningslåda 7 at Moodle



Task 1

In this task, you will start with your previous solution from Lab 6.
Create a new Java project in Eclipse, which will be a copy from your previous solution.
Your classes should belong to the package: `dt062g.studentid.assignment7` where studentid is your username in the student portal / Moodle.

The program should have a graphical user interface based on Swing. `System.in` may not be used for user input. `System.out` and `System.err` can only be used for debugging or error messages. You may not expect the user of the program to see these messages.

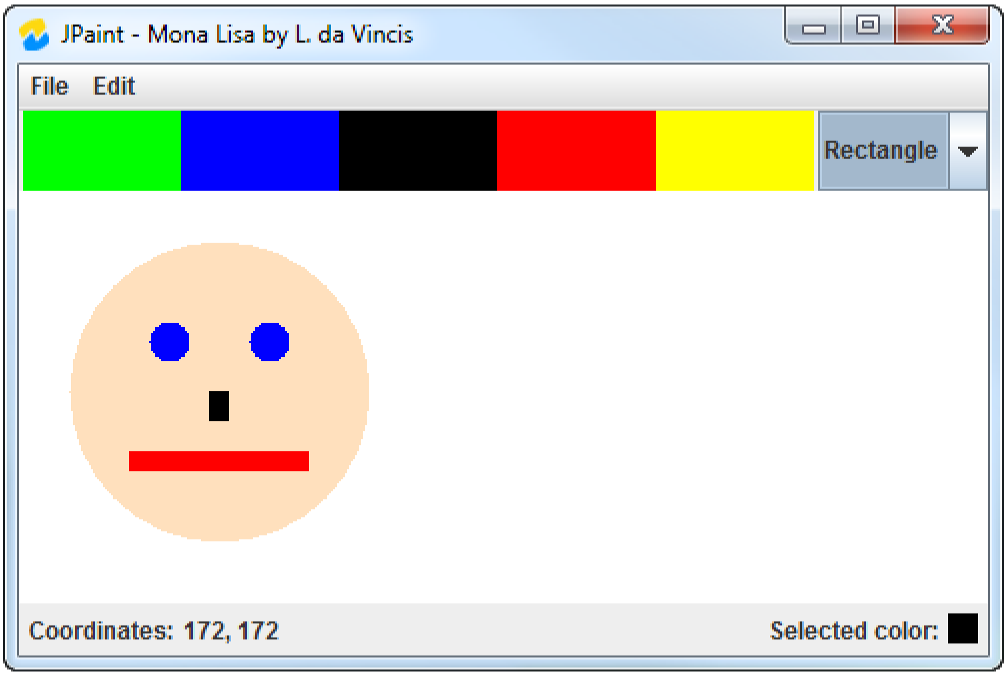
Each class and interface (interface) that you create must be documented with comments. Ex.

```
/**
* 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)
*/
```

Now you will complete the drawing program so that circles and rectangles can be drawn.

DrawingPanel

Start by creating a new class that will inherit from `JPanel`. This class will be named `DrawingPanel` and it is the class where you will draw all the figures.



The class should:

- Have an instance variable of type `Drawing`. It will contains the figures, which will be drawn in the panel.
- Have an empty/default constructor that creates a new empty `Drawing`-object.
- Have a constructor that takes a drawing-object as an argument and sets it as new current `Drawing`. The drawing area will be updated.
- Have a method `setDrawing(Drawing)` that sets new current `Drawing`. The drawing area will be updated.
- Have a method `addDrawing(Drawing)` that adds all shapes from the `Drawing,` that is sent as an argument to the method, into the current `Drawing`. In other words, the method will be used to add or put together two different `Drawing`–objects. The name and author of the Drawing will be the same as the first one.
- Have a method `getDrawing()` which returns current `Drawing`.
- Override the `paintComponent(Graphics)` method, and in this the current `Drawing` will be drawn by calling its method `draw(Graphics).` This will now call the respective figure's `draw(Graphics)`-method. If Drawing does not refer to something (i.e. it is `null`), nothing should be drawn. The background should always be white.
- If you think more methods are needed in the class you may add them.

Drawing, Circle and Rectangle

Implement the method `draw(Graphics)` in the classes `Drawing`, `Circle` and `Rectangle` so that they properly draw themselves according to the data (points) and the colors they have. The figures you are drawing should be filled with their color. Check how a circle (oval) in the `Graphics` class is drawn. Keep in mind that the first point in the `Circle` class should be the midpoint and the second point is the radius. Circles should be drawn with the antialiasing enable (if the platform supports it).

JFrame

In your class which inherits `JFrame` , `DrawingPanel` will now be used as drawing area.

When you choose **File | New**… the current `Drawing` in the drawing area (`DrawingPanel`) will be changed to a new empty `Drawing` with the name and author which the user writes in the dialog boxes.

When you choose **File | Save as**… the current `Drawing` in the drawing area will be saved to the file the user writes in the dialog box.

When you choose **File | Load**… and then you write a correct file name, the loaded `Drawing`-object will be set as current `Drawing` in the drawing area.

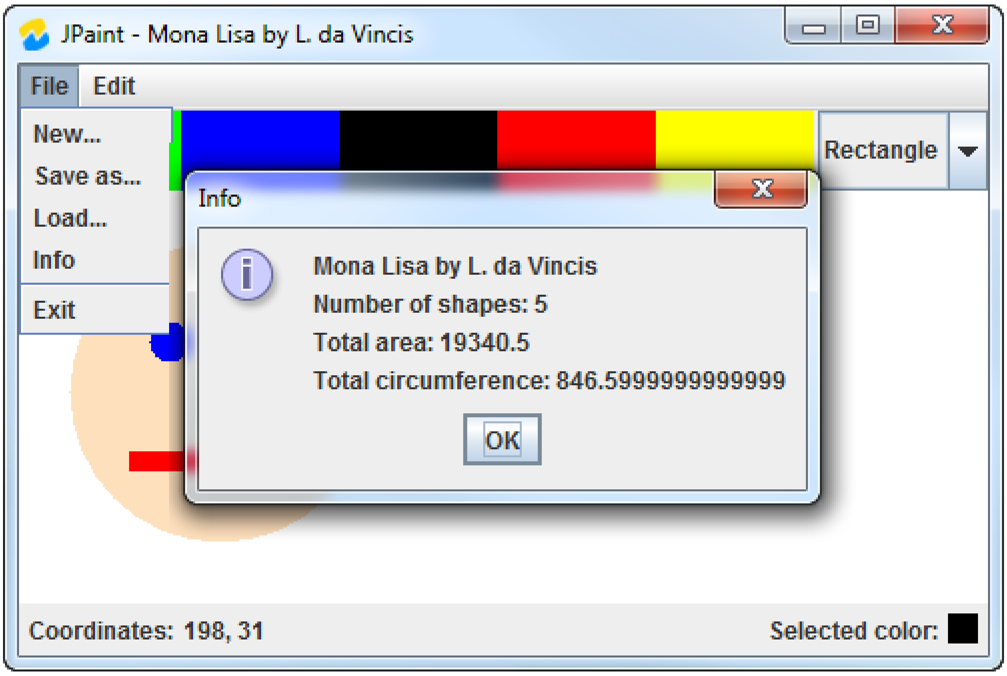
If you already have a `Drawing` in the drawing area, the loaded one will be added as the current drawing.

When you will choose **Edit | | Name** … the current `Drawing` in the drawing area will be updated with the name, the user writes in the dialog box.

When you will choose **Edit | | Author** … the current `Drawing` in the drawing area will be updated with the author, the user writes in the dialog box.

You can delete the menu option **Edit | Undo** if you don't want to do the extra task (check the last page)

Add a menu option **File | Info** that will display information about current `Drawing` in a dialog box. The information to be displayed is the name of the drawing according to the same format as the title of your JFrame. In addition, information about the number of figures, total area and total circumference of all figures in the drawing should be displayed.



Mona Lisa by L. da Vincis
Number of shapes: 5
Total area: 19340.5
Total circumference: 846.5999999999999

Coordinates: 198, 31

Selected color: ■

Assignment7

This class is attached to the task description. The class is used to create and display your JFrame. You can change the code so that the correct name of your class is used. There is some code in this class that have not been explained in any lesson. You can use the code as it is. Explanation will come in an upcoming lesson on threads.

Optional extra task (You can do it if you have time and interest)

You should make it possible to draw new circles and rectangles with the mouse. For example, the user should be able to select Circle from the combo box and then click somewhere in the drawing area, hold down the mouse, drag it, and then release the mouse and after that the circle should be complete. While the user draws the mouse, the new circle will always be drawn. The figure that the user has drawn (when the user releases the mouse button) should be added to existing `Drawing` in `DrawingPanel`.

The menu option **"Edit | Undo"** should now undo the last drawn figure by removing it from the list and repaint the drawing area.