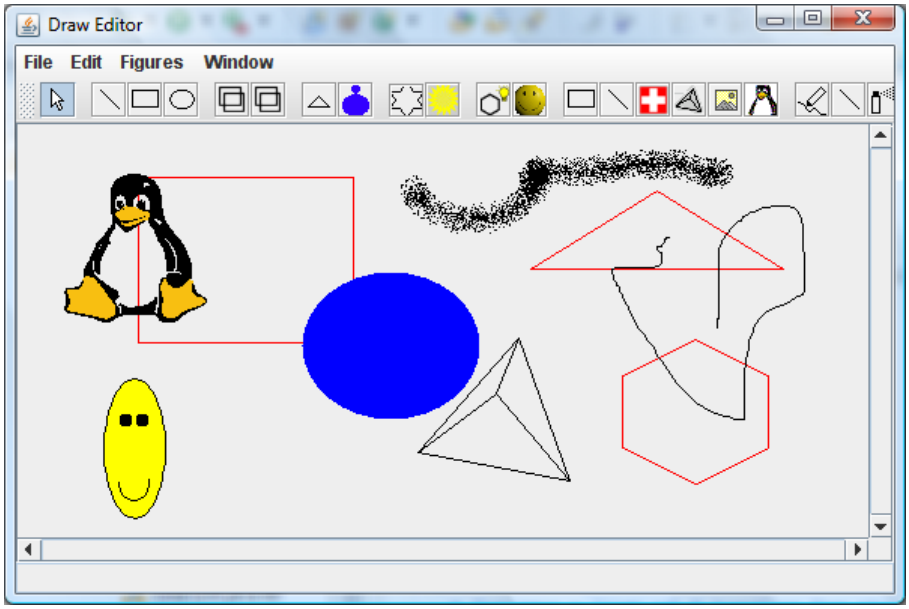
Assignment 3: Figure Palette

With the last assignment you became acquainted with the JDraw framework, you implemented the model and ensured that the views are updated upon changes. Currently, the graphics editor is rather limited as long as only rectangles can be drawn. In this assignment you will implement at least *two* additional figures. There are no limits to your imagination. Whether circles, lines or triangles or more complicated figures such as general polygons, curves, connecting arrows, etc. - anything goes.



Before you start implementing additional figures, here are some hints:

- Use already existing classes to draw your figures as much as possible. Look at the classes defined in package *java.awt.geom*. Classes *Line2D* and *Ellipse2D* might make your work easier.
- Use these classes but do not derive your figures directly from them (using inheritance).
- For each new figure you also have to provide a tool. This tool is then shown in the toolbar and allows switching into the requested figure mode. Icons for circles and lines are provided in the directory *res/images* of your JDraw project. If you implement other figures, you have to define your own icons.
- You can add your new figure tools in method *doRegisterDrawTools* with the method *addTool* to the figure menu and to the toolbar, similar to the registration of the given tool for rectangles. You find this method in class *jdraw/std/StdContext.java* at row 165 (look out for the comment // Add new figure tools here).

Have you noticed during implementation that you always have to program the same methods with exactly the same content over and over? For example, the *add-* and *removeFigureListener* methods? There are probably even more.

Consider where you can use common abstract base classes that contain the common code. This task is called extraction of code.

Summary:

1. Implement at least two additional figures which can be used in the graphics editor, i.e. a tool has to be provided for each figure as well.
2. Extract the similarities of the figure- and tool-classes in abstract base classes.

Deadline: October 16, 2018