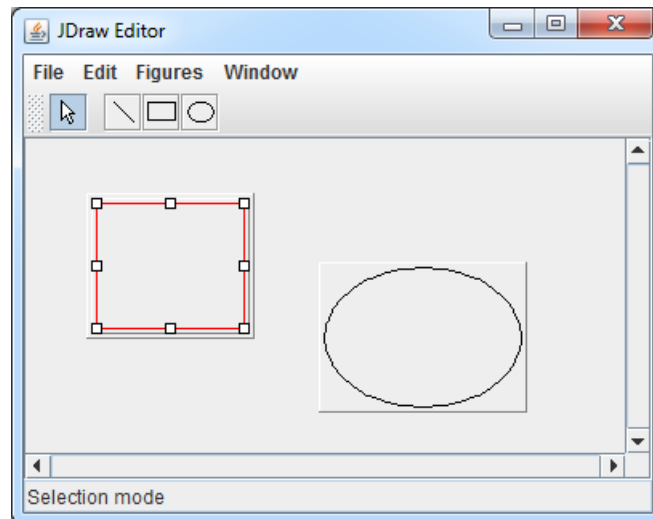


## Assignment 9: Take & Wrap: Border, Backgrounder, Bundler, Logger, ...

New figures can be provided by implementing the `Figure` interface (as you have done in assignment 3). Another possibility is to decorate existing figures.

As an example, implement a decorator that draws a border around a figure. A border is a white line above and left of the figure and a dark grey line below and right of the figure. It should be possible to draw several borders around a figure by decorating it several times.



Implement an abstract base class `DecoratorFigure` which contains a reference to the decorated figure and which forwards all methods to this inner instance. Concrete decorators can then extend this base class and override methods according to their behavior.

In order to use the decorators, you have to add the corresponding menu commands to class `StdContext`.

Besides the proposed *BorderDecorator* you could implement the following decorators:

- **BundleDecorator** A bundler prevents that the dimension of the figure can be changed (i.e. it suppresses the functionality of methods `move` and `setBounds` and it also prevents the publishing of the handles of the decorated figure (or it alternatively changes the behavior of the returned handles).
- **AnimationDecorator** An animation decorator regularly changes the position of the decorated figure in a separate thread.
- **LogDecorator** A logging decorator logs all method calls on the figure to `System.out` before they are forwarded to the decorated figure.

In order to test the functionality of your decorator we recommend that you define an additional figure test which operates on a decorated figure. Simply copy the existing test class `RectangleTest` and change method `setUp` as follows:

```
public class BorderRectangleTest {
    private Figure f;
    private int cnt;

    @Before
    public void setUp() {
        f = new BorderDecorator(new jdraw.figures.Rect(0, 0, 20, 10));
        cnt = 0;
    }
    ...
}
```

Deadline: December 11, 2018