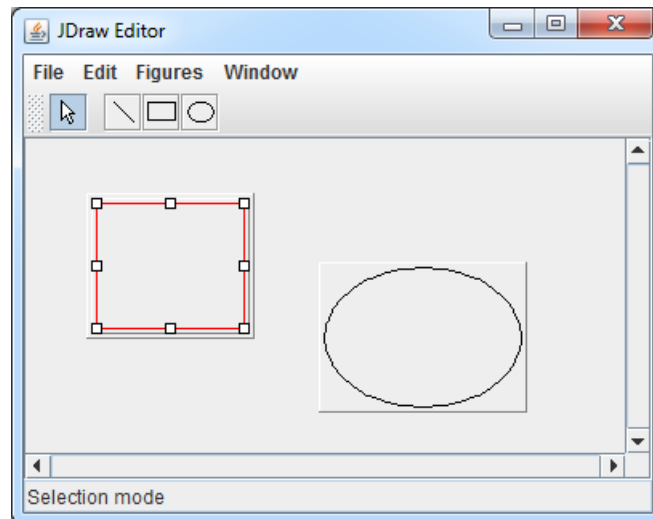


## Übung 9: Take & Wrap: Rahmen, Hintergrund, Logger, ...

Die Menge der Figuren kann einerseits durch neue Implementationen des Figure-Interface erweitert werden, andererseits aber auch dadurch, existierende Figuren zu dekorieren.

Als Beispiel soll ein Decorator implementiert werden, welcher einen Rahmen um eine Figur zeichnet. Ein Rahmen ist eine weisse Linie oben und links und eine graue Linie rechts und unterhalb der Figur. Ein Rahmen soll auch mehrfach um eine Figur gelegt werden können.



Stellen Sie zunächst eine abstrakte Klasse *DecoratorFigure* bereit, deren Objekte eine Referenz auf eine Figur enthalten und alle Methodenaufrufe an diese innere Instanz weiterleiten. Konkrete Decorator können diese Klasse erweitern und einzelne Methoden entsprechend ihrem Verhalten überschreiben.

Um den Decorator auch benutzen zu können, müssen Sie entsprechende Menu-Kommandos in *StdContext* einbauen.

Neben dem vorgeschlagenen *BorderDecorator* könnte man sich noch folgende Decorator vorstellen:

- *BundleDecorator* ein Decorator der verhindert, dass die Grösse der Figur geändert werden kann (verhindert Herausgabe der Original-Handles der dekorierten Figur).
- *AnimationDecorator* ein Decorator, welcher (in einem eigenen Thread) die Position der Figur verändert.
- *LogDecorator* ein Decorator, welcher sämtliche Methoden die aufgerufen werden auf `System.out` protokolliert.

Um die Funktionalität Ihres Dekorators zu prüfen empfehlen wir Ihnen, einen weiteren Figurentest zu definieren in welchem eine dekorierte Figur getestet wird. Sie können dazu die Klasse *RectangleTest* kopieren und die Methode *setUp* wie folgt anpassen:

```
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;
    }
    ...
}
```