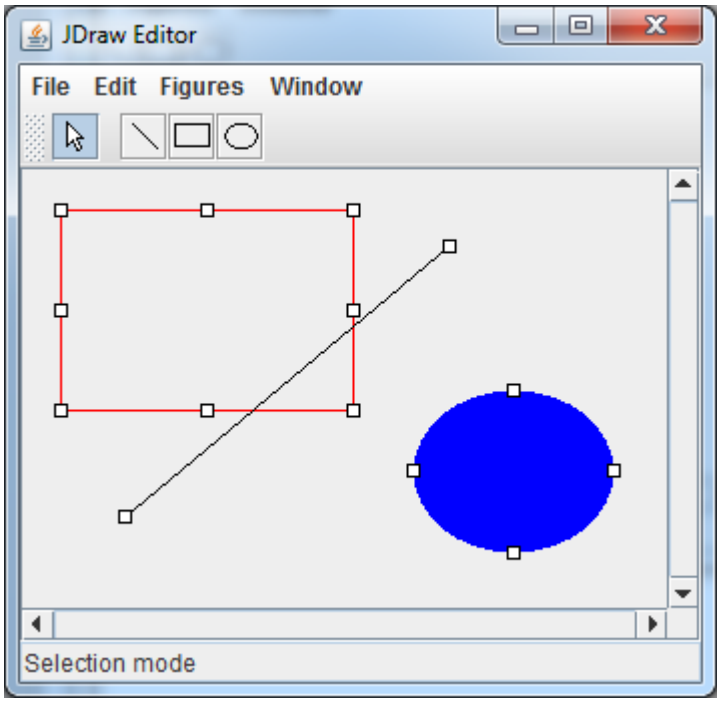
Übung 4: Figuren Handles

Mit den letzten beiden Übungen haben sie sich in das JDraw-Framework eingearbeitet, das Modell und erste Figuren entwickelt und die Aktualisierung der Views sichergestellt. Gewisse Methoden aus dem Interface `Figure` haben Sie jedoch noch nicht ausprogrammiert. In dieser Woche sollen Sie die Methode `getHandles` realisieren.

Mit der Methode `getHandles` gibt eine Figur ihre Handles zurück. Handles werden verwendet, um das Aussehen einer Figur zu verändern. In der nebenstehenden Figur bietet die Linie zwei und das Rechteck acht Handles an.



Die Handles implementieren ein State Pattern, denn je nach Handle (N, E, S, W, etc) sieht das Verhalten leicht anders aus.

Handles sind Objekte die das Interface `jdraw.framework.FigureHandle` implementieren. Analog zur Figurenklasse kennt auch dieses Interface die Methoden `draw`, `contains` und `getBounds`.

Mit der Methode `getCursor` kann ein Cursor definiert werden. Dieser wird angezeigt, wenn sich die Maus gerade über einem Handle befindet.

Die Methode `getOwner` soll für jeden Handle eine Referenz auf die zugehörige Figur zurückliefern.

Wenn die Maus in einem Handle gedrückt wird, so werden die Methoden `startInteraction`, `dragInteraction` und `stopInteraction` aufgerufen. Die zugehörige Figur soll dabei angepasst werden. Da während dem `dragInteraction` typischerweise die Methode `setBounds` der Figur aufgerufen wird, die ihrerseits einen Figure-Event an das Modell schickt, wird die View automatisch nachgeführt. Ein expliziter Aufruf der Methode `repaint` auf der View ist daher nicht nötig. Der zusätzliche Parameter vom Typ `MouseEvent` kann verwendet werden, um Modifier-Tasten auszulesen, und der Parameter vom Typ `DrawView` könnte verwendet werden, um auf den Draw-Context zuzugreifen um in der Statuszeile Text anzuzeigen.

Überlegen Sie sich auch (analog zu Übung 3) wo sie hier sinnvoll abstrakte Basisklassen einsetzen können.

Das Projekt das wir ihnen abgegeben haben ist auch vorbereitet damit leicht JUnit Tests formuliert werden können. In der Klasse `jdraw.test.JDrawTests` finden Sie eine Klasse welche zwei Testklassen zu einer Test-suite zusammenfasst. Ergänzen Sie diese Klassen laufend während (oder besser vor) der Entwicklung Ihrer JDraw Klassen.

Optionale Erweiterung:
Die Handles sollen so erweitert werden, dass die Figuren beim Drücken von Modifier-Tasten anders geändert werden. Bei gedrückter *Ctrl*-Taste soll z.B. das Zentrum der Figur fix bleiben und bei gedrückter *Shift*-Taste soll das Verhältnis von Breite zu Höhe konstant bleiben. Es handelt sich dabei wiederum um Verhalten das von einem Zustand abhängt. Der unterschiedliche Code könnte auch hier mit einem State-Pattern separiert werden.

Abgabe: 24. Oktober 2017