

Start mit JDraw: Erste Schritte

In diesem Dokument haben wir eine Anleitung zusammengestellt wie bei der Bearbeitung von Übung 2 am besten vorzugehen ist:

1. `StdDrawModel.getFigures` implementieren:
 - Wird bereits aus `StdDrawView` aufgerufen, aber funktioniert noch nicht
 - Braucht interne Datenstruktur, die alle Figuren speichert
 - Diese muss auch gefüllt werden können, also auch `StdDrawModel.addFigure` implementieren
 - Geeignete Datenstruktur auswählen: muss Reihenfolge sich überlappender Figuren wiedergeben können
2. `StdDrawModel.addModelChangeListener` implementieren:
 - Wird bereits aus `StdDrawView` aufgerufen, aber funktioniert noch nicht
 - Braucht interne Datenstruktur, die alle registrierten Observer speichert
 - Die registrierten Observer müssen dann bei den Änderungen des Models auch aufgerufen werden (interface `DrawModelListener` enthält die Information, wie das geht)

Resultat: Man kann ein Rechteck einfügen, evtl. bleibt es aber zufällig auf einer irgendeiner Grösse oder man sieht es zunächst gar nicht. Löst man ein Neuzeichnen des Fensters aus (z.B. vergrössern/verkleinern) sieht man das richtige Bild mit allen Figuren.

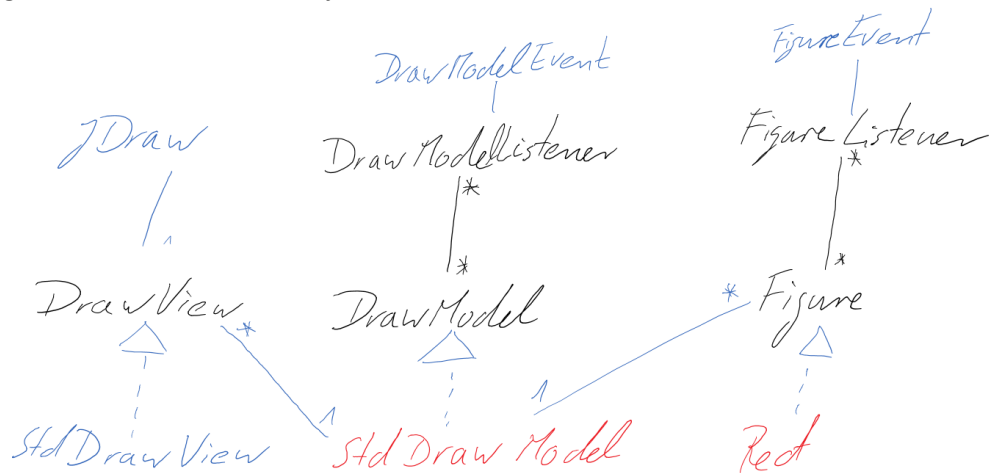
Der Grund ist, dass das Rechteck zunächst mit einer Ausdehnung von (0, 0) erzeugt und eingefügt wird. Von der Veränderung der Rechteck-Grösse erfährt die View nichts. Beim Neuzeichnen werden jedoch alle Figuren vollständig neu gezeichnet. Deswegen sieht man auch das Rechteck. Wenn man neu eingefügte Figuren grundsätzlich gar nicht sieht, funktioniert die Notifikation des Models bei `addFigure` nicht.

3. Grössen-Veränderungen von `Rect` müssen den Views mitgeteilt werden
 - `Rect` muss ebenfalls ein `Subject` (ein `Observable`) werden, in dem Observer registriert werden können
 - Eine entsprechende Methode `addFigureListener` ist definiert aber nicht implementiert und wird auch nicht aufgerufen
 - Bei Veränderungen der Objekt-Grösse oder Position (Methoden `Figure.setBounds` und `Figure.move`) müssen die registrierten Observer aufgerufen werden
 - Entsprechende Observer müssen programmiert und registriert werden. Es gibt verschiedene Möglichkeiten, wo diese Observer programmiert und registriert werden.

Resultat: Das Einfügen und das Verschieben von Rechtecken funktioniert nun korrekt, auch das gleiche Model gleichzeitig in mehreren Fenstern dargestellt wird. Rechtecke können jedoch nicht entfernt werden.

4. Vervollständigung:
 - Figuren können nicht nur in ein Model eingefügt, sondern auch wieder gelöscht werden. Die Methoden `StdDrawModel.removeFigure` und `StdDrawModel.removeAllFigures` müssen dazu implementiert werden. Was muss man dabei mit Hinblick auf die zuvor eingebauten Observer-Mechanismen alles beachten?
 - Views, die ein Model darstellen entsprechen Fenstern. Mit dem Menu-Befehl `Window -> New Window` können weitere Fenster auf das gleiche Model geöffnet werden. Beim Schliessen dieser Fenster wird die Methode `StdDrawModel.removeModelChangeListener` aufgerufen, die Sie ebenfalls implementieren müssen.
 - Ändern der Zeichenreihenfolge von Figuren: Die `StdDrawView` zeichnet die Figuren in der gleichen Reihenfolge, in der sie in der von `getFigures()` zurückgegebenen Liste enthalten sind. Mit den Menu-Befehlen unter `Edit -> Order...` möchte man die selektierte Figur an den Anfang oder das Ende der Liste versetzen können. Diese Menu-Befehle rufen dazu die Methode `StdDrawModel.setFigureIndex` auf. Implementieren auch diese Methode noch.
 - Benutzen Sie die mitgelieferten Junit-Tests, um Ihre Implementierung zu testen. Das Ziel ist, dass alle Tests korrekt laufen. Hierbei werden Sie vielleicht auch mit der einen oder anderen weiterführenden Überlegung zum Observer-Pattern aus den Unterrichtswochen 2 und 3 konfrontiert.

Wichtigste Klassen / Interfaces in jdraw:



schwarz: interfaces

blau: classes

rot: Klassen, in denen für Übung 2 programmiert werden muss (s. TODO-Tags)

Ablauf bei Einfügen eines neuen Rechtecks:

