

In dieser Aufgabe wird das Whiteboard aus der vorhergegangenen Aufgabe erweitert. Kopiere alle Dateien aus Aufgabe 1 in das Verzeichnis `aufgabe2`. Es kann auch die Musterlösung verwendet werden, die unter `/proj/i4oovs/pub/aufgabe1` (ab 27.04.2002) verfügbar ist.

- a) Nachdem die Methoden `move`, `getX` und `getY` bei allen Graphikobjekten identisch sind (oder zumindest sein können), bietet es sich an, diese in einer gemeinsamen Oberklasse zu implementieren. Dazu ist das Interface `Shape` in eine abstrakte Klasse umzuwandeln. In den Unterklassen soll jeweils nur noch Code, der spezifisch für ein einzelnes Graphikobjekt notwendig ist, vorhanden sein. (Die Test-Klassen aus Aufgabe 1 sollen weiterhin erfolgreich anwendbar sein.)
- b) Beim Auftreten von Exceptions (z.B. Analyse der Kommandozeilenparameter) ist darauf zu achten, dass eine sinnvolle Fehlermeldung ausgegeben wird.
- c) Jetzt soll der Anwender interaktiv Objekte verschieben können. Hierzu muss die Anwendung einen `MouseListener` und einen `MouseMotionListener` im `Whiteboard`-Frame installieren. Der `MouseListener` soll auf das `mousePressed`-Ereignis an der Stelle  $(x,y)$  reagieren, indem es mit Hilfe von `isInside` nach einem `Shape`-Objekt sucht, das den Punkt  $(x,y)$  enthält. Wurde ein solches gefunden, so soll der `MouseMotionListener` bei `mouseDragged`-Ereignissen die Position des Objekts mit `move` anpassen.  
Um Informationen zwischen den Event-Handlern auszutauschen, können Instanzvariablen des `Whiteboard` verwendet werden. Hierzu sollten die Handler innere Klassen davon sein. Die Position des Maus-Klicks relativ zur Position des Objekts ist zu beachten, damit das Objekt nicht zu Beginn des Verschiebens auf die Mausposition "springt".
- d) Nun soll der Anwender zusätzlich Objekte auch löschen können, indem er mit der rechten Maustaste in ein Objekt klickt. Liegt der Punkt innerhalb mehrerer Objekte, so sollen alle gelöscht werden.

**Bearbeitung: bis zum 3.5.2002**