Medientechnik - Programmierung 2

Aufgaben

Aufgabe 1: Affine Transformationen und Koordinatensysteme (6 Punkte)

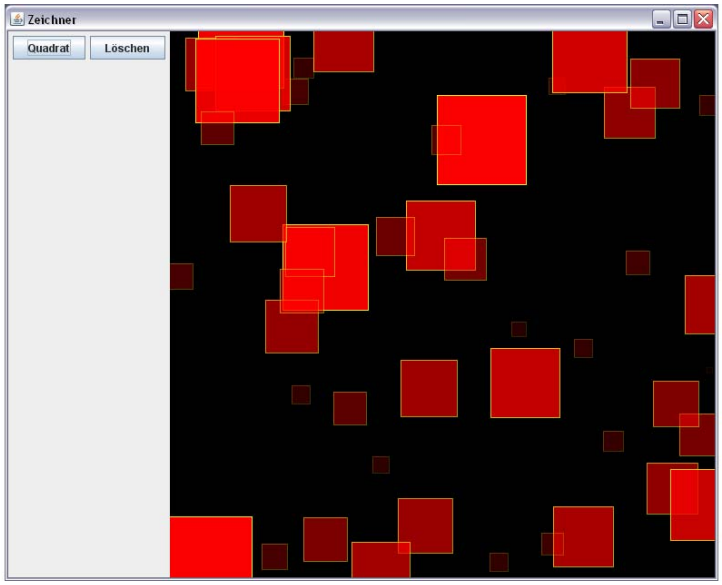
Java2D ermöglicht die einfache Nutzung der drei Transformationsarten Translation, Rotation und Skalierung für beliebige geometrische Objekte.

a) Recherchieren Sie, was Translation, Rotation und Skalierung bedeuten und erklären Sie jede dieser Transformationen in jeweils mindestens einem Satz.
b) Transformationen erfolgen in Java2D entweder über die Klasse *AffineTransform* oder direkt über den Aufruf der Methoden *translate(), rotate()* und *scale()* für ein *Graphics2D*-Objekt. Bei Nutzung dieser Klasse oder einem Aufruf der Methoden werden der Ursprung und die Rotation des *gesamten* Koordinatensystems verändert, um die jeweilige Transformation zu erzeugen, was sich natürlich auf alle danach gezeichneten Objekte auswirkt. Wie müssen diese Methoden genutzt werden, dass sich Transformationen nur auf einzelne Objekte und nicht auf alle auswirken?
c) Probieren Sie Folgendes entweder direkt in Java2D oder auf einem Blatt Papier aus: Gegeben ist ein Quadrat mit einer Seitenlänge von 50 Pixeln. Dieses wird im Ursprung des Koordinatensystems (linke obere Ecke im Punkt 0,0) gezeichnet.
    a. Davor wird der Befehl *translate(50, 50)* gefolgt von *scale(2, 2)* ausgeführt. An welchen Koordinaten befindet sich in der finalen Darstellung die linke obere Ecke des Quadrats?
    b. Jetzt wird die Reihenfolge dieser Transformationsbefehle vertauscht. Landet die linke obere Ecke des Quadrats diesmal an einer anderen Stelle oder hat die Vertauschung der Transformationen keinen Einfluss darauf?

Speichern Sie Ihre Antworten in der Textdatei "aufgabe1.txt" und fügen Sie sie Ihrer Abgabe bei.

Aufgabe 2: Primitive mit Java 2D (8 Punkte)

Implementieren Sie eine graphische Benutzeroberfläche mit Java2D, die folgendes leistet:



Es wird ein Fenster geöffnet, das zwei nebeneinander liegende Panels enthält. Auf dem linken werden zwei Knöpfe, "Quadrat" und "Löschen" angezeigt. Das rechte zeigt zu Beginn nur einen schwarzen Hintergrund. Durch Klick auf den Knopf "Quadrat" wird ein neues Quadrat im rechten Panel erzeugt, das eine zufällig generierte Seitenlänge zwischen 10 und 100 Pixeln hat und sich an einer ebenfalls zufällig erzeugten Position befindet. Das Quadrat wird mit roter Farbe gefüllt und hat einen gelben Rand. Je kleiner das Quadrat ist, desto durchscheinender ist es. Wiederholtes Klicken auf "Quadrat" fügt weitere Quadrate hinzu, deren Größe und Position ebenfalls zufällig ausgewählt werden. Durch Klick auf den Knopf "Löschen" werden alle Quadrate entfernt.

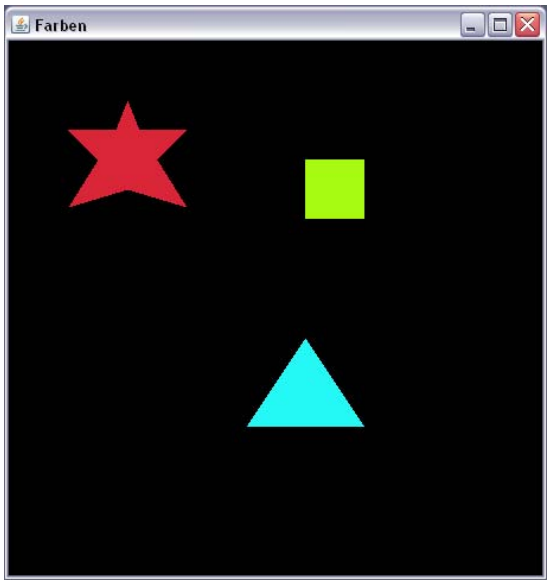
Benutzen Sie die Model-View-Controller- und Observer-Entwurfsmuster um diese Anwendung zu erstellen.

Weitere Vorgaben sind:

- Nutzen Sie zur Speicherung der Quadrate im Modell die Klasse *java.awt.Rectangle* oder *java.awt.geom.Rectangle2D*.
- Benutzen Sie die Klasse *java.util.Vector* um eine sich verändernde Zahl von Quadraten im Modell zu speichern.
- Machen Sie Quadrate durchscheinend mithilfe des Alphawerts in *java.awt.Color*.
- Nutzen Sie die statische Methode *Math.random()* zur Erzeugung von Zufallszahlen.

Legen Sie alle so erstellten Java **Quellcodedateien** in einem Verzeichnis namens "aufgabe2" ab, erstellen Sie außerdem eine lauffähige JAR-Datei und fügen Sie beides Ihrer Abgabe bei. **Achtung: Abgaben ohne Quellcode werden nicht gewertet!**

Aufgabe 3: Nutzerinteraktion mit Java2D (6 Punkte)



Erstellen Sie eine weitere Java2D-Anwendung: Diese soll nur ein einzelnes Panel mit schwarzem Hintergrund zeigen, das drei verschiedenfarbige geometrische Figuren enthält (nutzen Sie dafür vordefinierte Java2D- oder AWT-Primitive oder entwerfen Sie selbst Polygone). Sobald der Benutzer eine der Figuren anklickt, wird die Farbe der Figur auf einen zufälligen Wert geändert.

Weitere Anforderung:

- Benutzen Sie zur Behandlung von Mouseclicks das Interface *MouseListener* oder die Klasse *MouseAdapter*.

Legen Sie alle so erstellten Java **Quellcodedateien** in einem Verzeichnis namens "aufgabe3" ab, erstellen Sie außerdem eine lauffähige JAR-Datei und fügen Sie beides Ihrer Abgabe bei. **Achtung: Abgaben ohne Quellcode werden nicht gewertet!**

Abgabe

Packen Sie die beiden JAR-Dateien und Verzeichnisse und die Textdatei in eine ZIP-Datei und geben Sie diese Lösung bis zum 28.05.08, 10:00 Uhr im UniWorx Portal (http://www.pst.ifi.lmu.de/uniworx) ab.
Wichtig: Abgaben ohne Quellcode werden nicht gewertet! Abgaben außerhalb von UniWorx (z.B. per Email) werden ebenfalls nicht bewertet!