

JAVA-ÜBUNGSAUFGABEN Animationen 1

Bemerkungen:

Das Programmdesign und die Lösung der Aufgaben sollen objektorientiert sein.
Erzeugen Sie für jedes Programm ein UML.

1)

a) Das Schaubild der folgenden Formel ergibt eine Ellipse:

$$\left(\frac{y}{a}\right)^2 + \left(\frac{x}{b}\right)^2 = 1 \quad \text{wobei } a \text{ und } b \text{ vorgegebene Zahlen sind, wie z.B. } a=3 \text{ und } b=2$$

Lassen Sie das Schaubild durch ein Java-Programm zeichnen.

a) Die Hochzahl 2 wird durch die Variable h ersetzt und man erhält dann die Gleichung einer verallgemeinerten Ellipse:

$$\left(\frac{y}{a}\right)^h + \left(\frac{x}{b}\right)^h = 1 \quad \text{wobei } a \text{ und } b \text{ vorgegebene Zahlen sind, wie z.B. } a=3 \text{ und } b=2$$

Was geschieht, wenn h große positive Werte annimmt?

Lassen Sie das Schaubild durch ein Java-Programm zeichnen.

2)

Voraussetzung:

Sie haben den Quellcode (Demo-Animation) einer funktionierenden Java-Animation vom Lehrer erhalten.

In dieser Animation bewegt sich ein Ball von links nach rechts.

V1) Das Programm verletzt eine Richtlinie der OOP (Attribute dürfen nicht public sein).
Ändern Sie dies ab.

V2) Wenn der Ball am rechten Rand angekommen ist, soll es wieder vom linken Rand aus weiterfahren, usw.

Dies soll unabhängig von der Länge des Fensters funktionieren.

V3) Das Programm soll in einer Mehrschichtenarchitektur entwickelt werden:

Zusätzlich zur GUI-Schicht (Klasse Zeichenflaeche) muß noch eine Fachkonzeptschicht erstellt werden.

Das wäre hier z.B. die Klasse Ball.

Die Klasse Ball muss eine eigene Zeichenmethode enthalten, die den Ball zeichnet.

Ein Ball soll vom linken zum rechten Rand des Bildschirms fliegen. Wenn er am rechten Rand angekommen ist, soll es wieder vom linken Rand aus weiterfahren.

Der Ball "befindet" sich einer Grafikdatei (ball.png)

3)

Der Ball soll die Bewegung eines Betrunkenen simulieren (wissenschaftlich: random walk), der sich nur über Zufall fortbewegt:

Mit jeweils der Wahrscheinlichkeit 0,25 einen Schritt nach oben, nach unten, nach rechts und nach links.

4)

Zusätzlich zu dem Ball soll nochmal ein Ball (aus der gleichen Grafikdatei oder ein Ball in einer selbst erstellten Grafikdatei) vom linken zum rechten Rand des Bildschirms fahren

5)

Zusätzlich zu den Bällen soll noch ein Rechteck (soll ein Auto darstellen und deshalb in einer Klasse Auto erstellt werden) vom linken zum rechten Rand des Bildschirms fahren.

Wichtig:

Jede Klasse Ball bzw. Auto muss eine eigene Zeichenmethode enthalten, die das entsprechende Fahrzeug zeichnet.

6)

Zusätzlich zu den Bällen und dem Auto soll noch ein weiteres Auto vom linken zum rechten Rand des Bildschirms fahren.

Wichtig:

Jede Klasse Ball bzw. Auto muss eine eigene Zeichenmethode enthalten, die das entsprechende Fahrzeug zeichnet.

7)

Die Geschwindigkeit der bewegten Objekte soll durch Veränderung der Schrittbreite verändert werden.

8)

a) Die Geschwindigkeit der bewegten Objekte soll durch Veränderung der Verzögerungszeit des Timers verändert werden.

b) Die bewegten Objekte sollen periodisch immer schneller und dann langsamer werden.

9)

a) Bälle sollen an der Wand reflektiert werden.

b) Mehrere Bälle werden an der Wand reflektiert. Bei einer Kollision werden sie gelöscht.

10)

A1) Ein Auto soll vom linken zum rechten Rand des Bildschirms fahren. Wenn es am rechten Rand angekommen ist, soll es wieder vom linken Rand aus weiterfahren.

b) Das Auto soll bis an den rechten Rand der Zeichenfläche fahren, auch wenn diese durch den Anwender verkleinert oder vergrößert wird (durch Ziehen des Fensters).

A2) a) Die Geschwindigkeit des Autos soll mit einem Schieberegler (JSlider) verändert werden können (siehe Beispielprogramm zu JSlider).

b) Zusätzlich soll noch die aktuelle Geschwindigkeit in einem Textfeld angezeigt werden.

A3) Die Aufgabe soll mit 2 Autos realisiert werden. Für jedes Auto soll ein eigener Schieberegler (JSlider) verändert werden können.

A4) Die Aufgabe soll mit 2 Autos und genau einem Schieberegler realisiert werden. Bei jedem Klick auf einen bestimmten Button verändert sich die Markierung eines Autos (d.h. es wird das andere Auto markiert).

Die Markierung soll an der roten Autofarbe (des aktuellen Autos) erkennbar sein.
Die Geschwindigkeit des aktuellen Auto kann dann mit Hilfe des Schieberegler verändert werden.

A5) Die Aufgabe soll mit mehreren Autos (programmtechnisch mit einem Feld) realisiert werden.

A6) Beim Anklicken eines bestimmten Buttons soll ein neues Auto auf den Bildschirm kommen, beim Anklicken eines anderen Buttons soll ein Auto auf dem Bildschirm verschwinden. Benutzen Sie dazu die als Liste die Klasse ArrayList (siehe Dokument und Beispielprogramm zu ArrayList).

Tipp:

Erstellen Sie die Klasse AutoListe, in der die Verwaltung (entfernen, hinzufügen, verändern des Listenzeiger) der Liste realisiert wird.

Die Größe, die Farbe und der Ort des Wagens kann entweder fest vorgegeben sein (wie in den vorhergehenden Programmen) oder auch vom Anwender ausgewählt werden.

A7) Es sollen nicht nur Autos auf dem Bildschirm erscheinen, sondern auch andere Fahrzeuge (LKW, Traktor). Die Liste heißt dann FahrzeugListe

A8) Aus einem Auto kann (im Gegensatz zu einem LKW oder einem Traktor) ein Vermietfahrzeug (Taxi) gemacht werden. Bringen Sie einen Gebührenzähler an das Vermietfahrzeug an.