

Präsenzaufgaben 8

23./24.11.2022

Die Lösung der Aufgaben wird am Ende der Übung von Ihnen vorgestellt.

Geometrische Figuren

In dieser Aufgabe sollen geometrische Figuren grafisch auf dem Schirm angezeigt werden. Laden Sie sich dazu bitte von der Veranstaltungsseite die Quelldatei `Grafik.java` herunter. Sie liefert einen Rahmen zur Anzeige Ihrer geometrischen Figuren auf dem Bildschirm und besteht aus den Klassen `Grafik` und `Figur`. `Grafik` hat diese Methoden und Konstruktoren:

- `/* Oeffnet ein Grafikfenster mit einer Zeichenflaeche der angegebenen Breite und Hoehe */`
`public Grafik(int width, int height)`
- `/* Fuegt die genannte geometrische Figur in die Zeichenflaeche ein */`
`public void add(Figur f)`
- `/* Entfernt die geometrische Figur aus der Zeichenflaeche */`
`public void remove(Figur f)`
- `/* Aktualisiert die Zeichenflaeche. Aenderungen werden erst sichtbar, nachdem diese Methode aufgerufen wurde. Wartet vorher ms Millisekunden. Damit ist eine einfache Animation moeglich */`
`public void aktualisiere(int ms)`

Ihre geometrischen Klassen müssen die Klasse `Figur` ableiten. `Figur` besitzt zwei abstrakte Methoden. Abstrakte Methoden *müssen* in den Unterklassen überschrieben werden. Dies sind die Methoden:

```
public abstract int getPunktCount()  
public abstract Point getPunkt(int i)
```

`getPunktCount` gibt die Anzahl der zur `Figur` gehörenden Punkte zurück. (Die Klasse `Point` ist eine Java-Klasse aus dem Paket `java.awt.`). `getPunkt` gibt den zu einem Index gehörenden Punkt zurück. In der `Grafik` werden Verbindungslinien zwischen den jeweils benachbarten Punkten der `Figur` gezeichnet.

Außerdem besitzt `Figur` die Methode

```
public void verschiebe(int x, int y)
```

die alle Punkte der `Figur` um den Vektor (x, y) verschiebt.

a) **Linie**

Laden Sie die Klassen `Linie` und `LinieTest` von der Veranstaltungsseite und machen Sie sich anhand des Beispiels die Funktionsweise des Programms klar.

b) **Rechteck**

Schreiben Sie eine Klasse für ein achsenparalleles Rechteck. Fügen Sie die folgenden Konstruktoren und Methoden hinzu:

- `/* Konstruktor aus zwei gegenueberliegenden Eckpunkten */`
`public Rechteck(Point p1, Point p2)`
- `/* Kopierkonstruktor */`
`public Rechteck(Rechteck r)`
- `/* Skaliert das Rechteck mit dem Faktor d. Der Mittelpunkt des Rechtecks bleibt erhalten */`
`public void skaliere(double d)`
- `/* Gibt das kleinstmoegliche Rechteck zurueck, das sowohl this als auch r enthaelt */`
`public Rechteck vereinige(Rechteck r)`
- `/* --> Rechteck: Hoehe = ..., Breite= ... */`
`public String toString()`

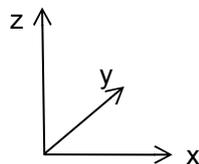
c) **Kreis**

Schreiben Sie eine Klasse für einen Kreis. Konstruieren Sie einen Kreis aus 100 kleinen Sekanten. Benutzen Sie die Sinus- und Cosinus-Funktion zur Berechnung der Endpunkte der Sekanten. Fügen Sie folgende Methode hinzu:

```
/*-> Kreis: Mittelpunkt = ..., Radius = ...*/
public String toString()
```

d) **Zusatzaufgabe: Quader**

Schreiben Sie eine Klasse für einen dreidimensionalen Quader. Die Projektion der y-Achse soll in Richtung (1,1) gehen, so wie in der Zeichnung dargestellt. Die Skalierung der y-Achse ist gegenüber der x- und der z- Achse um den Faktor $\sqrt{2}$ gestaucht.



Schreiben Sie einen Konstruktor:

```
public Quader(Rechteck x, int tiefe)
```

Die Dimensionen des Rechtecks ergeben die x- und die z-Achse des Quaders, die Tiefe ergibt die y-Achse des Quaders. Beachten Sie, dass Sie einen Würfel nicht in einem Zug zeichnen können, sondern Linien mehrfach zeichnen müssen.

Hinweise:

Sie können den Code von `Grafik.java` gerne modifizieren, falls Sie Swing beherrschen und wissen, was Sie tun. Lassen Sie die Vorlage ansonsten lieber unangetastet (dass das möglich ist, ist ein großer Vorteil der objektorientierten Programmierung). Die Methode `paint` aus `Figur` ist `public`, um den Code kurz zu halten. Die Methode ist aber nur zur internen Benutzung gedacht.