

— Übungsblatt 3 (P) —

Aufgabe 10 (P)

(Grafikprogrammierung)

a)

Schreiben Sie eine Klasse `Punkt` zur Darstellung von Punkten in der Zeichenebene. Der Konstruktor der Klasse soll zwei `double`-Parameter erwarten (die x - und y -Koordinaten des Punkts) und in entsprechenden privaten Instanzvariablen speichern. Schreiben Sie je eine Zugriffsmethode `getX()` bzw. `getY()`, die die x - bzw. y -Koordinate des Punkts zurückliefert.

Schreiben Sie außerdem eine Methode `drehen`, die den Punkt um den übergebenen Winkel ϕ (im Bogenmaß) dreht. Die Koordinaten $(x_{\text{neu}}, y_{\text{neu}})$ eines um den Winkel ϕ gedrehten Punkts $p = (x_p, y_p)$ berechnen sich als

$$x_{\text{neu}} := x_p \cdot \cos(\phi) - y_p \cdot \sin(\phi) \quad \text{und} \quad y_{\text{neu}} := x_p \cdot \sin(\phi) + y_p \cdot \cos(\phi).$$

b)

Schreiben Sie eine Klasse `Strecke`, die das (zum Download bereitgestellte) Interface

```
import java.awt.*;

public interface GeoObjekt {
    public void drehen(double phi);
    // dreht das Objekt um den Winkel phi

    public void zeichnen(Graphics g, int xNull, int yNull);
    // zeichnet das Objekt auf der Zeichenebene
    // xNull und yNull sind die Koordinaten des Ursprungs
    // (Nullpunkts) des verwendeten Koordinatensystems
}
```

implementiert. Der Konstruktor soll zwei `Punkt`-Parameter (die Randpunkte der Strecke) akzeptieren und deren Referenzen in privaten Instanzvariablen ablegen. Die beiden abstrakten Methoden `drehen` und `zeichnen`, die von der Schnittstelle `GeoObjekt` definiert werden, sind zu implementieren. Greifen Sie dabei so weit wie möglich auf Methoden der Klasse `Punkt` zurück.

Achten Sie beim Zeichnen in der Methode `zeichnen` darauf, dass die Linie relativ zum übergebenen Nullpunkt $(x_{\text{Null}}, y_{\text{Null}})$ ausgegeben wird. Verwenden Sie beim Aufruf der Methode `drawLine` der Klasse `Graphics` daher die um x_{Null} bzw. y_{Null} erhöhten Koordinaten.

c)

Schreiben Sie eine Klasse `Dreieck`, die ebenfalls die Schnittstelle `GeoObjekt` implementiert. Der Konstruktor soll drei `Punkt`-Parameter erwarten und sie in entsprechenden `protected`-Instanzvariablen speichern. Im Konstruktor sollen mit diesen drei Punkten drei neue Objekte der Klasse `Strecke` instantiiert werden, deren Referenzen in privaten Instanzvariablen gespeichert werden.

Implementieren Sie die Methode `drehen`, indem Sie nur auf die jeweilige Methode der drei `Punkt`-Objekte zurückgreifen.

Implementieren Sie die Methode `zeichnen`, indem Sie nur auf die jeweiligen Methoden der drei `Strecke`-Objekte zurückgreifen.

d)

Schreiben Sie eine Klasse `DrehPanel`, die von `JPanel` erbt und als Zeichenfläche dienen kann, auf der fest vorgegebene zweidimensionale geometrische Objekte gedreht werden können. Die Klasse soll mit einer finalen Klassenvariablen `SCHRITTWEITE` mit Wert $\frac{\pi}{60}$ (die Schrittweite für Drehungen) und einer privaten Variable `drehObjekt` vom Typ `GeoObjekt` (das Objekt, das gedreht werden soll) ausgestattet sein.

Im Konstruktor der Klasse sollen vier Instanzen der Klasse `JButton` erzeugt, mit *Links*, *Rechts*, *Strecke* und *Dreieck* beschriftet und dem Panel hinzugefügt werden. Bei jedem dieser Buttons soll ein `ActionListener`-Objekt unter Verwendung einer anonymen Klasse registriert werden. In der jeweiligen Methode `actionPerformed` soll dabei

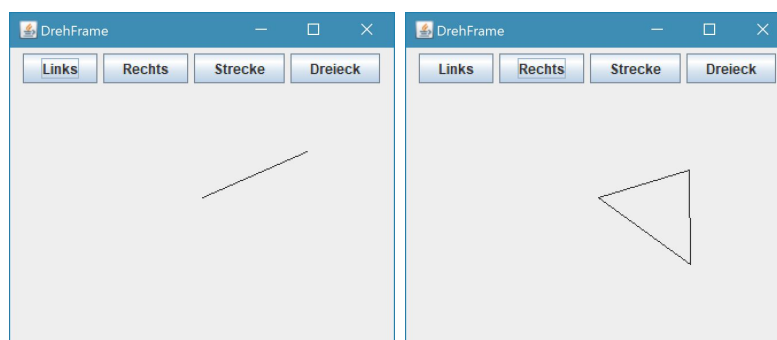
- für den mit *Links* beschrifteten Knopf das Objekt `drehObjekt` um die negative Schrittweite gedreht,
- für den mit *Rechts* beschrifteten Knopf das Objekt `drehObjekt` um die positive Schrittweite gedreht,
- für den mit *Strecke* beschrifteten Knopf das Objekt `drehObjekt` mit der Methode `erzeugeStrecke` als neue Strecke erzeugt,
- für den mit *Dreieck* beschrifteten Knopf das Objekt `drehObjekt` mit der Methode `erzeugeDreieck` als neues Dreieck erzeugt

und jeweils `repaint` zum Neuzeichnen aller Komponenten aufgerufen werden. Der Konstruktor soll auch dafür sorgen, dass die Variable `drehObjekt` zu Beginn mit einer Strecke initialisiert wird.

Zum Erzeugen einer neuen Strecke müssen Sie nun noch die Methode `erzeugeStrecke()` schreiben, die ein neues Objekt der Klasse `Strecke` mit den Endpunkten (0,0) und (100,0) erzeugt und als Ergebnis zurückliefert. Außerdem müssen Sie noch die Methode `erzeugeDreieck()` zum Erzeugen eines neuen Dreiecks ergänzen, die ein neues Objekt der Klasse `Dreieck` mit den Eckpunkten (0,0), (100,0) und (50, -66) erzeugt und als Ergebnis zurückliefert. Schließlich überschreiben Sie noch die Methode `paintComponent(Graphics g)`, indem Sie darin zunächst die entsprechende Methode der Superklasse und anschließend die Instanzmethode `zeichnen` des Objekts `drehObjekt` aufrufen. Verwenden Sie dabei die Koordinaten des Panel-Mittelpunkts für die Parameter `xNull` und `yNull`.

e)

Schreiben Sie eine Frame-Klasse, in deren Konstruktor ein Objekt der Klasse `DrehPanel` erzeugt und in die Content-Pane des Frames eingefügt wird. In der `main`-Methode der Klasse sollten Sie den Frame dann in den Dimensionen 350 × 300 Pixel erzeugen, so dass sich die grafische Oberfläche folgendermaßen präsentiert:



Aufgabe 11 (P)

(Frames)

Erstellen Sie eine grafische Oberfläche der Form

die es ermöglichen soll, unterhalb der Beschriftung Argument eine Zahl einzugeben. Diese soll dann als Argument für die Logarithmus- oder die Quadratwurzel-Funktion (je nach Markierung der entsprechenden Checkbox) verwendet werden. Durch den Button rechts oben soll die Berechnung ausgelöst und das Ergebnis direkt unter dem Button angezeigt werden.

Programmieren Sie den Konstruktor so, dass ein geeignetes Layout festgelegt wird, die benötigten Swing-Komponenten erzeugt werden (beachten Sie bei den JCheckBox-Objekten, dass jeweils nur eines davon aktiviert sein darf und zu Beginn die Logarithmus-Funktion ausgewählt sein soll), die Swing-Komponenten in der richtigen Reihenfolge eingefügt werden und der Berechnungs-Button mit einem Event-Listener verknüpft wird.

Realisieren Sie die Ereignis-Behandlung für den Berechnungs-Button in Form einer inneren Klasse ExecuteListener, die das Interface ActionListener implementiert. Bei der Betätigung des Berechnungs-Buttons soll jeweils der im Textfeld eingegebene Wert bestimmt werden, die Logarithmus- oder die Quadratwurzel-Berechnung (abhängig vom Zustand der Checkbox-Markierungen) durchgeführt und das Ergebnis in dem dafür vorgesehenen Label angezeigt werden. Bei einer unzulässigen Eingabe soll eine entsprechende Exception abgefangen und dies über den Text des Resultats-Labels dem Benutzer bzw. der Benutzerin mitgeteilt werden.