



Technische Universität Darmstadt
 Fachbereich Informatik
 Prof. Johannes Fürnkranz

Allgemeine Informatik II im SS 2007

Übungsblatt 7

Bearbeitungszeit: 13.06. bis 19.06.2006

Aufgabe 1: Aufrufzähler mit Klassenvariablen

Erstellen Sie in BlueJ folgende Klasse **Statics**:

```
public class Statics {
    private static int anzahl = 0;

    public Statics() {
        anzahl++;
    }

    public int anzahl() {
        return anzahl;
    }
}
```

Schreiben Sie dann eine Klasse **StaticsTest**, die drei Objekte vom Typ **Statics** anlegt und anschließend den Rückgabewert der Methode **anzahl()** (Aufruf auf einem beliebigen der drei Objekte) auf dem Bildschirm ausgibt.

Was beobachten Sie?

Haben Sie dieses Ergebnis erwartet?

Aufgabe 2: Kreisberechnung mit Klassen- und Objektmethoden

Aus dem Radius eines Kreises können Durchmesser, Umfang und Fläche berechnet werden.

- a) Schreiben Sie eine Klasse **PlayCircle**, die **public**-Methoden **durchmesser(...)**, **umfang(...)** und **flaeche(...)** zur Berechnung von Durchmesser, Umfang und Fläche enthält. Jede dieser Methoden soll dabei als Parameter den Radius übergeben bekommen und als Rückgabewert das entsprechende Berechnungsergebnis liefern. Da es sich um Fließkommazahlen handelt, verwenden Sie bitte immer den Datentyp **double**. Die Methoden sollen außerdem mit **static** gekennzeichnet werden. Überlegen Sie sich warum!

Zu Berechnung von Kreisfläche und -umfang wird die Zahl π benötigt. Definieren Sie diese als Konstante (mit mindestens 5 Nachkommastellen) in der Klasse **PlayCircle**: **public static final double PI = ...**

- b) Testen Sie die Methoden Ihrer Klasse mit BlueJ.
- c) Ergänzen Sie nun die Klasse **PlayCircle** um folgende nicht-statische Methoden und Variablen:

- Eine **double**-Variable **radius** für den Radius.
- Ein Konstruktor, der einen **double**-Parameter übergeben bekommt und damit den Radius initialisiert.
- Eine Methode **public double radius()**, die den Radius zurückgibt.
- **public**-Methoden **durchmesser()**, **umfang()** und **flaeche()**, die wie in Aufgabenteil a) die entsprechenden Werte abhängig von der Objektvariable **radius** zurückgeben.
- Eine Methode **public boolean isEqual(PlayCircle circle)**, die **true** zurückgibt, wenn beide Kreise den gleichen Radius haben.
- Eine Methode **public void print()**, die eine Ausgabe folgender Art auf dem Bildschirm ausgibt:

```
Kreis mit Radius: 5.232
Durchmesser: 10.464
Umfang: 32.87362552716356
Flaeche: 85.99740437905989
```

- d) Testen Sie die neuen Methoden, indem Sie in BlueJ mehrere Objekte der Klasse **PlayCircle** anlegen und darauf die verschiedenen Methoden aufrufen.

Aufgabe 3: Polynome (FREIWILLIGE FLEISSAUFGABE)

Ein kubisches Polynom hat die Gestalt $ax^3 + bx^2 + cx + d$. Die Werte a , b , c und d sind die „Koeffizienten“ des Polynoms.

- a) Definieren Sie eine Klasse **Polynom** zur Speicherung eines maximal kubischen Polynoms mit **double**-Koeffizienten.
- b) Schreiben Sie vier passende Konstruktoren, mit denen Polynome auf folgende Art definiert werden können:
- **Polynom p = new Polynom(2)** für $p(x) = 2$
 - **Polynom q = new Polynom(1, 0.5)** für $q(x) = x + 0.5$
 - **Polynom r = new Polynom(3, -3, 2)** für $r(x) = 2x^2 - 3x + 2$
 - **Polynom s = new Polynom(4, 0, -1, -3)** für $s(x) = 4x^3 - x - 3$
- c) Definieren Sie eine Methode **Polynom add(Polynom poly)**, mit dem ein **Polynom** auf das aktuelle **Polynom**-Objekt addiert und als neues **Polynom** zurückgegeben wird.
Beispiel: **Polynom t = q.add(r)** für $t(x) = 2x^3 - 2x + 2.5$
- d) Schreiben Sie eine Methode **double apply(double x)**, die den Wert eines Polynoms für ein gegebenes **x** berechnet.
Beispiel: **double y = s.apply(2.0)** für $y = 4 \cdot 2^3 - 2 - 3 = 27$
- e) Schreiben Sie eine Methode **Polynom dx()**, die das nach **x** abgeleitete **Polynom** des aktuellen **Polynom**-Objekts liefert.
Beispiel: **Polynom u = s.dx()** für $u(x) = s'(x) = 12x^2 - 1$

Testen Sie Ihre Klasse mittels BlueJ.

Viel Spaß!