



# Einführung in die Informatik

für Hörer alle Fakultäten

(WS-03/04)

## Übungsblatt 6

**Ausgabe:** Do. 2003-11-27

**Abgabe:** Do. 2003-12-04

**Besprechung:** Di./Do. 2003-12-09/11

Beachten Sie auch für dieses Aufgabenblatt die Hinweise auf Übungsblatt 4 zur Abgabe der Lösungen mit Praktomat unter:

<https://jop.informatik.uni-wuerzburg.de/infohaf/praktomat.cgi>

### Aufgabe 6.1: Kreis Klasse (2 Punkte)

Schreiben Sie eine öffentliche Klasse `Circle`.

Die Klasse soll private Attribute für Kreismittelpunkt und Radius verwalten.

Die Klasse soll mit folgenden Konstruktoren erzeugt werden können:

- `public Circle()` erzeugt einen Kreis mit Mittelpunkt (0,0) und Radius 1.
- `public Circle(double x, double y, double r)` erzeugt einen Kreis mit Mittelpunkt (x,y) und Radius r.

Darüber hinaus soll Sie folgende Methoden bereitstellen:

- `public void setCenter(double x, double y)` zum Setzen des Mittelpunkts.
- `public double getCenterX()` zum Auslesen der x-Koordinate des Mittelpunkts.
- `public double getCenterY()` zum Auslesen der y-Koordinate des Mittelpunkts.
- `public void setCenterX(double x)`  
zum Setzen der x-Koordinate des Mittelpunkts.
- `public void setCenterY(double y)`  
zum Setzen der y-Koordinate des Mittelpunkts.
- `public double getRadius()` zum Auslesen des Radius.
- `public double setRadius(double r)` zum Setzen des Radius.
- `public double area()` liefert die Fläche des Kreises.
- `public double circumference()` liefert den Umfang des Kreises.
- `public String toString()`  
liefert eine textuelle Repräsentation des Kreises in der Form: "Circle(x,y,r)"  
also z.B. "Circle(0.0,0.0,1.0)".

Schreiben Sie weiterhin eine Klasse CircleTest mit folgenden Methoden:

- `public static void main(String[] args)`  
Eine Main-Methode, die die Funktionalität der Klasse Circle testet und demonstriert. Dies wird nicht automatisch geprüft, Sie sollten jedoch trotzdem eine sinnvolle Demonstration implementieren, da das bei der manuellen Abschlusskorrektur berücksichtigt wird. Darüber hinaus dient es für Sie selbst zur Überprüfung Ihrer Klassen und Methoden.

## Aufgabe 6.2: Polymorphe Figuren (4 Punkte)

Gegeben ist folgendes Interface:

```
/**
 * Das Interface Figure definiert die gemeinsame
 * Schnittstelle für verschiedene geometrische Figuren.
 * @author fischer
 * @version 1.0
 */
public interface Figure {
    /**
     * Berechnet die Fläche der geometrischen Figur.
     * @return Die Fläche der Figur
     */
    public double area();

    /**
     * Berechnet den Umfang der Figur, d.h. die Länge der "Außenkanten".
     * @return Der Umfang der Figur.
     */
    public double circumference();

    /**
     * Berechnet, ob die Figur den Punkt mit den angegebenen Koordinaten
     * überdeckt.
     * @param x Die x-Koordinate des zu überprüfenden Punktes
     * @param y Die y-Koordinate des zu überprüfenden Punktes
     * @return true wenn die Figur den Punkt überdeckt,
     *         false sonst.
     */
    public boolean overlaps(double x, double y);

    /**
     * Liefert einen, die Figur beschreibenden String, mit dem Namen der Figur
     * und den wichtigsten Attributen.
     * @return Eine textuelle Beschreibung der Figur.
     */
    public String toString();
}
```

Schreiben Sie die Klassen Rectangle, Circle und Point, die alle das Interface Figure sinnvoll implementieren.

Darüber hinaus müssen die Klassen folgende Konstruktoren bereitstellen:

- `public Rectangle(double xLowerLeft, double yLowerLeft, double width, double height)` für die Klasse Rectangle.
- `public Circle(double x, double y, double r)` für die Klasse Circle.
- `public Point(double x, double y)` für die Klasse Point.

Schreiben Sie weiterhin eine Klasse FigureTest mit folgenden Methoden:

- `public static int countOverlappingFigures(Figure[] figures, double x, double y)`  
Diese Methode liefert die Anzahl der Figuren aus figures, die den Punkt (x,y) überdecken.
- `public static void main(String[] args)`  
Eine Main-Methode, die die Funktionalität der Klassen Rectangle, Circle, Point testet und demonstriert. Dies wird nicht automatisch geprüft, Sie sollten jedoch trotzdem eine sinnvolle Demonstration implementieren, da das bei der manuellen Abschlusskorrektur berücksichtigt wird. Darüber hinaus dient es für Sie selbst zur Überprüfung Ihrer Klassen und Methoden.

## Aufgabe 6.3: Figur Vergleich (4 Punkte)

Erstellen Sie eine Klasse `Figures`, die eine Menge von `Figure`-Objekten (siehe Übungsaufgabe 6.2) als eine Folge verwaltet.

Dazu soll ein privates Array der Größe 100 von `Figure` angelegt werden. Ein unbenutzter Speicherplatz wird mit `null` gekennzeichnet. Zu Beginn sind also alle Werte des Arrays auf `null` gesetzt.

Erstellen sie weiterhin ein folgende Methoden:

- `public void pushFigure(Figure f)`  
Hängt die Figur `f` an, indem es den ersten, noch freien Speicherplatz damit belegt. Ist kein Speicherplatz mehr vorhanden, wird die Aktion einfach verworfen, also keine Änderungen vorgenommen.
- `public Figure popFigure()`  
Entfernt die letzte Figur aus der Folge und liefert sie zurück. Ist kein Element mehr in der Folge enthalten, so soll `null` zurückgeliefert werden.
- `public void sortByArea()`  
Sortiert das Array so um, dass die geometrischen Figuren mit der größten Fläche am Anfang stehen. Für das Sortieren können Sie einen beliebigen Sortieralgorithmus verwenden, z.B. aus der Vorlesung oder vom letzten Übungsblatt.
- `public void sortByCircumference()`  
Sortiert das Array so um, dass die geometrischen Figuren mit dem größten Umfang am Anfang stehen. Für das Sortieren können Sie einen beliebigen Sortieralgorithmus verwenden, z.B. aus der Vorlesung oder vom letzten Übungsblatt.
- `public static void main(String[] args)`  
Die Main-Methode soll zunächst einen Integer einlesen. Dieser ist folgendermassen zu interpretieren:
  - 1: Erstelle einen neuen Kreis.
  - 2: Erstelle ein neues Rechteck.
  - 0: Keine weiteren Objekte erstellen.

Je nach Auswahl werden anschließend die Daten für die zu erzeugenden Objekte eingelesen:

### **Kreis:**

- X-Koordinate des Mittelpunkts
- Y-Koordinate des Mittelpunkts
- Radius

### **Rechteck:**

- X-Koordinate der linken unteren Ecke.
- Y-Koordinate der linken unteren Ecke.
- Breite.
- Höhe.

Mit diesen Daten wird dann ein Objekt der entsprechenden Klasse erzeugt und in die Folge der Figuren eingefügt (z.B. mit `pushFigure()`).

Zur Konstruktion verwenden sie dabei die auch in der Aufgabe 6.2 verlangten Konstruktoren der Klassen `Rectangle` und `Circle`. Diese Klassen brauchen nicht unbedingt implementiert und abgegeben zu werden, es wird bei der Abgabe automatisch eine Implementierung zur Verfügung gestellt. Sie können aber natürlich auch selbst eine erstellen, oder Ihre Implementierung aus Aufgabe 6.2 verwenden.

Wenn keine weiteren Objekte mehr erstellt werden sollen, dann sollen die Figuren nach der Fläche sortiert ausgegeben werden (aufsteigend).