



Aufgabe 8.1 (Ü) Kreise

In dieser Aufgabe soll eine Klasse für Kreise entworfen werden.

- Schreiben Sie dazu zunächst eine Klasse `Vector2d`, die einen zweidimensionalen Vektor repräsentiert. Er soll die Koordinaten `x` und `y` vom Typ `double` als Objektvariablen enthalten. Diskutieren Sie mit Ihrem Tutor, welche Objektmethoden hilfreich sein könnten, um in den folgenden Teilaufgaben mit Vektoren rechnen zu können und implementieren Sie diese.
- Schreiben Sie anschließend eine Klasse für Kreise (`Circle`). Überlegen Sie sich hierfür zunächst, welche Eigenschaften diese Objekte haben. Diskutieren Sie, welche Eigenschaften als Objektvariablen und welche als Objektmethoden implementiert werden sollten.
- Schreiben Sie anschließend die Objektmethoden:

```
public boolean isIncluded(Circle c)
```

```
public boolean includes(Circle c)
```

zu der Klasse `Circle` die testet, ob der jeweilige Kreis in einem anderen enthalten ist, bzw. ob er einen anderen Kreis enthält.

Testen Sie Ihre Implementierung in einer geeigneten Testklasse mit einer `main`-Methode mit verschiedenen geeigneten Ein- und Ausgaben.

Aufgabe 8.2 (Ü) Einfache verkettete Liste

Einfach verkettete Integer-Listen bestehen aus einer Kette von Elementen. Jedes Element besteht aus einer Zahl (`info`) und der Referenz (`next`) auf den Rest der Liste. Die leere Liste wird durch `null` repräsentiert.

- Erstellen Sie eine Klasse `IntList`. Implementieren Sie einen passenden Konstruktor, der die Attribute `info` und `next` initialisiert.
- Implementieren Sie eine Methode `append(int info)`, die eine neue Liste zurückgibt, die eine Kopie der aktuellen Liste ist, an deren Ende ein neues Element mit Zahl `info`, angefügt wurde.
- Implementieren Sie eine Methode `public String toString()`, die alle Elemente der Liste als String zurückgibt.

Aufgabe 8.3 (H) Einfache verkettete Liste (Fortsetzung)

Die Hausaufgabe setzt Aufgabe 8.3 fort.

- a) Implementieren Sie eine Methode `sum()`, die die Summe aller Listenelemente zurückgibt.
- b) Implementieren Sie eine Methode `reverse()`, die eine neue Liste zurückgibt, welche die Zahlen der aktuellen Liste in umgekehrter Reihenfolge enthält.

Hinweis: Überlegen Sie sich zunächst eine Lösung für einelementige Listen. Danach überlegen Sie sich, wie Sie mit Listenelementen verfahren, auf die noch weitere Elemente folgen.