

Programmierkurs Anwendungsentwicklung

Praktikum 01

Aufgabe 1

- Implementieren Sie eine Klasse `Rectangle` mit den Attributen `breite` und `hoehe`. `Rectangle` soll die Methoden `equals`, `hashCode` und `toString` sinnvoll implementieren.
 - `toString` soll den Hashcode des Objekts und die Attributwerte ausgeben
 - `hashCode` soll folgende Implementierung haben:
`final int prime = 31;`
`int result = 1;`
`result = prime * result + hoehe; result = prime * result + breite;`
 - Implementieren Sie `equals` sinnvoll
- Schreiben Sie eine Textklasse. Diese soll zwei `Rectangle`-Objekte erzeugen und vergleichen und jeweils die Methode `toString` aufgerufen werden

Aufgabe 2

- Implementieren Sie eine Klasse `Circle` mit dem Attribut `radius`
- Die Klasse `Rectangle` aus Aufgabe 1 und die eben implementierte Klasse `Circle` soll nun das Interface `Geometry` implementieren.
 - Das Interface besitzt folgende Methoden:

```
double berechneUmfang()  
double berechneFlaeche()
```
- Implementieren Sie eine Testklasse. Hier soll ein Array mit 4 Rechtecken und 4 Kreisen angelegt werden. Geben Sie in einer `for`-Schleife nacheinander zuerst die Objektspezifischen-Attribute (`laenge/breite` bzw. `radius`) und danach die berechnete Fläche und den berechneten Umfang aus.

Aufgabe 3

- Implementieren sie zu gegebener Klasse (siehe nächste Folie) die „equals“-Methode und entsprechende Hilfsmethoden/getter/setter. Testen sie ihr Programm, indem sie zwei Personen anliegen und diese „vergleichen“. „Spielen“ sie verschiedene Testfälle durch.

Aufgabe 3 cont.

```
1 package P01;
2
3 import java.util.Objects;
4
5 public class Person {
6     private final String name;
7     private final String city;
8     private String street; // Optional => null-Wert erlaubt
9     private String zipcode; // Optional => null-Wert erlaubt
10
11     public Person(final String name, final String city, final String street, final String zipcode) {
12
13         Objects.requireNonNull(name, "parameter 'name' must not be null!");
14         Objects.requireNonNull(city, "parameter 'city' must not be null!");
15
16         this.name = name;
17         this.city = city;
18
19         this.street = street;
20         this.zipcode = zipcode;
21     }
22 }
```

Aufgabe 4

Hinweis: Prüfung auf semantische Gleichheit Für Subklassen müssen in der Regel nicht nur die eigenen Attribute, sondern auch alle relevanten Bestandteile der Klassenhierarchie geprüft werden.

Aufgabe: Implementieren sie eine Klasse RefinedPerson. Diese soll von der Klasse Person aus Aufgabe 4 erben. Die Klasse RefinedPerson bekommt das Attribut nickName. Erweitern sie ihre Testklasse um zwei RefinedPersonen. Testen sie die semantische Gleichheit.

