



Programmierung für Mathematik (HS13)

Übung 7

1 Aufgabe: Konstruktoren

1.1 Lernziele

1. Konstruktoren anwenden

1.2 Aufgabenstellung

In den vergangenen Übungen haben Sie getter und setter Methoden verwendet, um den Instanzvariablen Werte zuzuweisen. In dieser Aufgabe geht es darum, dass Sie diese Klassen mit Hilfe von Konstruktoren so erweitern, dass Sie Instanzvariablen gleich bei der Instanziierung Werte zuweisen können.

Erweitern Sie die folgenden Klassen der vergangenen Übungen:

- `MyList` (Übung 5): Diese Klasse soll einen Konstruktoren haben, der ein `int[]` Array erwartet und die `setArray` Methode ersetzt. Schreiben Sie ausserdem einen Default-Konstruktoren, der keine Parameter hat und das Array mit von Ihnen ausgewählten Default-Werten initialisiert.
- `Trapezoid` (Übung 6): Diese Klasse soll einen Konstruktoren haben, der die `double` Werte für die Seite `a` und `c` sowie die Höhe erwartet und dann den Instanzvariablen zuweist. Schreiben Sie ausserdem einen Default-Konstruktoren, der keine Parameter hat und die beiden Seiten sowie die Höhe mit von Ihnen ausgewählten Default-Werten initialisiert.

2 Aufgabe: Repetition: Employee Klasse schreiben

2.1 Lernziele

1. Eine Klasse mit Instanzvariablen, Konstruktoren und Methoden schreiben.
2. Die erstellte Klasse testen.

2.2 Aufgabenstellung

Sie haben die Aufgabe, ein neues Mitarbeiterverwaltungs-System zu entwickeln. Die Funktionen sind zunächst noch recht rudimentär, sollen aber in zukünftigen Versionen erweitert werden.

Schreiben Sie eine neue Klasse `Employee`, die einen Mitarbeiter repräsentiert. Jeder Mitarbeiter hat einen Vor- und Nachnamen, sowie eine Adresse. Die Adresse besteht aus Strasse, Postleitzahl und Ort. Schreiben Sie anschliessend folgende Methoden und Konstruktoren:

- Einen Default-Konstruktor, der allen Instanzvariablen Standardwerte zuweist. Das System soll ausgeben, dass ein neuer Test-Mitarbeiter erstellt worden ist.
- `setFirstName` und `setLastName`: Diese Methoden erwarten einen `string` und initialisieren die entsprechende Instanzvariable sofern die Länge des angegebenen Strings grösser als 0 ist. Andererseits wird eine Fehlermeldung ausgegeben.
- `setStreet`: Ähnlich wie in den beiden Methoden oben, soll die Strasse als Parameter erwartet werden und nur dann der Instanzvariable zugewiesen werden, wenn die Länge der Strasse (`string`-Länge) grösser als 0 ist. Geben Sie eine Fehlermeldung aus, falls dies nicht der Fall ist.
- `setZipAndResidence`: Diese Methode erwartet die Postleitzahl und den Ort als Parameter und fügt diesen den entsprechenden Instanzvariablen hinzu, sofern die Länge des Ortes (`string`-Länge) grösser als 0 und die Postleitzahl zwischen 4 und 6 Zahlen hat. Andererseits wird ebenfalls eine Fehlermeldung ausgegeben.
- Einen Konstruktor, der die Instanzvariablen als Parameter erwartet und diese über die eben erstellten Methoden zuweist.
- Eine Methode `print`, die die Mitarbeiter-Daten auf der Konsole ausgibt.

Schreiben Sie anschliessend einen `TestDriver`, der die Funktionalität testet.

3 Aufgabe: Roll the Dice v2.0

3.1 Lernziele

1. Eine bestehende Klasse erweitern

3.2 Aufgabenstellung

In dieser Aufgabe geht es darum, das RollTheDice Spiel aus der letzten Übung so zu erweitern, dass beliebig viele Spieler mitspielen können. Der `TestDriver` soll nun vom User in der Konsole die Anzahl Spieler erfragen und anschliessend das Spiel mit der eingegebenen Anzahl Spielern starten. Auf der nächsten Seite finden Sie die `RollTheDice`-Klasse, deren leere Methoden Sie wie folgt anpassen sollen:

- `runNextRound`: Diese Methode soll für jeden Spieler einmal den Würfel werfen und das Zwischenresultat in das Array `playerTotalPoints` speichern.
- `checkIfAnyPlayerWon`: Diese Methode kontrolliert für jeden Spieler, ob er bereits die erforderliche Anzahl Punkte (siehe `target`) erreicht hat. Ist dies der Fall, so wird `true` zurückgegeben, was die `while`-Schleife in der methode `runGame` stoppt. Andernfalls wird `false` zurückgegeben und das Spiel fortgesetzt.

```

1 import java.util.Random;
2
3 public class RollTheDice {
4
5     private int target = 30;
6     private int[] playerTotalPoints;
7     private int numberOfPlayers;
8
9     public void initiateGame(int numberOfPlayers) {
10         playerTotalPoints = new int[numberOfPlayers];
11         this.numberOfPlayers = numberOfPlayers;
12     }
13
14     public void runGame(){
15         boolean playerWon = false;
16
17         while (playerWon == false) {
18             runNextRound();
19             playerWon = checkIfAnyPlayerWon();
20         }
21     }
22
23     private void runNextRound() {
24
25
26
27
28
29     }
30
31     private boolean checkIfAnyPlayerWon() {
32
33
34
35
36
37
38     }
39
40     public void reset(){
41         playerTotalPoints = new int[numberOfPlayers];
42     }
43 }

```

Listing 1: unvollständige RollTheDice-Klasse