



Praktikum zu
**Einführung in die Informatik für
LogWings, WiMas und MedPhys**
Wintersemester 2020/21

Übungsblatt 10
Besprechung:
1.–5.02.2021 (KW 5)

Vorbereitende Aufgaben

Aufgabe 10.1: Wiederholung: Klammern

Quiz

Welche Klammer wird wofür verwendet?

- | | | |
|---|----------|----------|
| i) Kennzeichnung von Blöcken | | |
| a) [...] | b) (...) | c) {...} |
| ii) Parameter-Angabe bei Funktionen | | |
| a) [...] | b) (...) | c) {...} |
| iii) Definition von Arrays und Zugriff auf Arrayelementen | | |
| a) [...] | b) (...) | c) {...} |
| iv) Priorisierung von Berechnungen bzw. Auswertung | | |
| a) [...] | b) (...) | c) {...} |

Aufgabe 10.2: Wiederholung: Funktionsparameter

Geben Sie einen geeigneten Methodenkopf für die folgenden öffentlichen, statischen Funktionen an.

- a) Eine Funktion **average**, die den Durchschnitt eines **double**-Arrays berechnet und zurückgibt.

- b) Eine Funktion **plus**, die zwei reelle Zahlen miteinander addiert und die Summe zurückgibt.

- c) Eine Funktion **countWords**, die die Wörter in einem **String** zählt und zurückgibt.

- d) Eine Funktion **printMaximum**, die das Maximum eines **int**-Arrays mit `System.out.println` auf dem Bildschirm ausgibt.
-

- e) Eine Funktion **times**, die einen Integer **n** und einen Integer **x** entgegen nimmt und ein **n** Elemente langes Array, gefüllt mit dem Wert **x** zurückgibt.
-

Präsenzaufgaben

Aufgabe 10.3: Fehlersuche

In den nachfolgenden Klassen haben sich syntaktische und semantische Fehler eingeschlichen. Korrigieren Sie sämtliche Fehler in den Klassen auf dem Blatt. Der Programmcode soll dabei **nicht** implementiert werden.

```
1 using java.util.Random;
2
3 public class Player {
4     private int score;
5     private String name;
6     private Random d20;
7
8     private Player(String name) {
9         name = name;
10        d20 = new Random();
11        score = 0;
12        public getName() {
13            return name;
14        }
15
16        public int getScore {
17            return score;
18        }
19
20        public String getInfo() {
21            return getName() + " with " + getScore() + "points ";
22        }
23
24        public void addScore(int score); {
25            return null;
26            score += score;
27        }
28
29        public void int throwDice() {
30            return d20.nextInt(20) + 1;
31    }
```

```

1 public classe Game {
2
3     private Player[] players;
4     private Random d20;
5
6     public Game(double n) {
7         d20 = new Random();
8         players = new Player[n];
9         for (int i = 0, i < n, i++); {
10             players[i] = new Player("Player" + i);
11         }
12     }
13
14     public static int throwDice() {
15         return d20.nextInt(20) + 1;
16     }
17
18     public protected void playRound() {
19         for (int i = 0; j < players.lenght; i++) {
20             if (players[i].throwDice() > throwDice() {
21                 players[i].addScore(10.0);
22             } else {
23                 players[i].addScore(-10.0);
24             }
25         }
26     }
27
28     public Player getWinner[] {
29         Player winner = players(0);
30         for (int i = 1; i < players.lenght; i++) {
31             if (winner.getScore() < player[i].getScore()) {
32                 winner = player[i];
33             }
34         }
35         return winner;
36     }
37 }

```

```

1 public class Program {
2
3     public stativ void main(String args) {
4         Game game = new Game(10);
5         for (int i = 0; i < 100; i++)
6             game.playRound();
7         Player winner = game.getWinner();
8         System.out.println("The first winner is " + winner.getInfo());
9     }
10
11     public stativ void main(String args) {
12         Game game = new Game(25);
13         for (int i = 0; i < 100; i++)
14             game.playRound();
15         Player winner = game.getWinner();
16         System.out.println("The second winner is " + winner.getInfo());
17     }

```


Aufgabe 10.6: Klassenvariablen Implementierung

Erweitern Sie die Klassen **Car** und **Vehicle** um je eine private Klassenvariable **carCounter** bzw. **vehicleCounter**, die die Anzahl der erzeugten **Car**- bzw. **Vehicle**-Objekte zählt. Geben Sie anschließend in Ihrer Testklasse die Anzahl der Instanziierungen aus.

Hinweis: Sie benötigen hierzu eine funktionierende Lösung der Aufgaben aus Blatt 9.

Ergänzende Aufgaben

Aufgabe 10.7: Minimum rekursiv

In einer Aufgabe von Blatt 8 haben Sie eine Funktion geschrieben, die das Minimum eines **int**-Arrays findet. Höchstwahrscheinlich haben Sie dies iterativ mit einer **for**-Schleife gelöst. Diese Funktion wollen wir nun rekursiv implementieren. Legen Sie dazu die Klasse **MinRec** an. Verwenden Sie bei der Implementierung *keine* Schleifen!

- a) Schreiben Sie nun eine Funktion **minArray** mit einem **int**-Array und einem Index als Parameter. Die Funktion soll das Minimum des Arrays ab dem Index zurückgeben.

Beispiel: Sei das Array $a = \{30, 10, 50, 20, 40, 60\}$, soll `minArray(a, 2)` den Rückgabewert 20 haben.

- b) Überladen Sie die Funktion **minArray** mit einer Funktion, die nur ein **int**-Array als Parameter hat. Diese soll das Minimum des **ganzen** Arrays zurückgeben. Rufen Sie dazu die soeben geschriebene **minArray**-Funktion auf.