

Praktikum zu
**Einführung in die Informatik für
LogWings, WiMas und MedPhys**
Wintersemester 2019/20

Übungsblatt 10
Besprechung:
13.–17.01.2020
(KW 3)

Vorbereitende Aufgaben

Aufgabe 10.1: Wiederholung: Klammern

Geben Sie an, wofür folgende Klammern in Java verwendet werden.

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 im Paket „blatt10“ 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 package blatt10;
2
3 using java.util.Random;
4
5 public class Player {
6     private int score;
7     private String name;
8     private Random d20;
9
10    private Player(String name) {
11        name = name;
12        d20 = new Random();
13        score = 0;
14        public getName() {
15            return name;
16        }
17
18        public int getScore {
19            return score;
20        }
21
22        public String getInfo() {
23            return getName() + " with " + getScore() + "points ";
24        }
25
26        public void addScore(int score); {
27            return null;
28            score += score;
29        }
```

```

30
31     public void int throwDice() {
32         return d20.nextInt(20) + 1;
33     }

```

```

1  package Blatt10;
2
3  public classe Game {
4
5      private Player[] players;
6      private Random d20;
7
8      public Game(double n) {
9          d20 = new Random();
10         players = new Player[n];
11         for (int i = 0, i < n, i++); {
12             players[i] = new Player("Player" + i);
13         }
14     }
15
16     public static int throwDice() {
17         return d20.nextInt(20) + 1;
18     }
19
20     public protected void playRound() {
21         for (int i = 0; j < players.lenght; i++) {
22             if (players[i].throwDice() > throwDice() {
23                 players[i].addScore(10.0);
24             } else {
25                 players[i].addScore(-10.0);
26             }
27         }
28     }
29
30     public Player getWinner[] {
31         Player winner = players(0);
32         for (int i = 1; i < players.lenght; i++) {
33             if (winner.getScore() < player[i].getScore()) {
34                 winner = player[i];
35             }
36         }
37         return winner;
38     }
39 }

```

```

1  package blatt10;
2
3  public class Program {
4
5      public stativ void main(String args) {
6          Game game = new Game(10);
7          for (int i = 0; i < 100; i++)
8              game.playRound();
9          Player winner = game.getWinner();
10         System.out.println("The first winner is " + winner.getInfo());
11     }

```

```

12
13     public static void main(String args) {
14         Game game = new Game(25);
15         for (int i = 0; i < 100; i++)
16             game.playRound();
17         Player winner = game.getWinner();
18         System.out.println("The second winner is " + winner.getInfo());
19     }

```

Aufgabe 10.4: Objektvariablen und -methoden vs. Klassenvariablen und -methoden

In dieser Aufgabe wollen wir uns mit der unterschiedlichen Verwendung von Objekt- und Klassenelementen vertraut machen. Manche Zuweisungen und Methodenaufrufe sind im unteren Programm nicht erlaubt (vgl. dazu die Folien in Kapitel 6). Notieren Sie auf den Linien neben dem Programmtext, ob die jeweilige Zuweisung oder der jeweilige Methodenaufruf erlaubt ist oder nicht.

```

1  class Tester {
2      int var1;
3      static int var2;
4
5      void test1() {
6          var1++;
7          var2--;
8      }
9
10     static void test2() {
11         var1++;
12         var2--;
13     }
14
15     public static void main(String[] args) {
16         var1 = 1;
17         var2 = 1;
18         test1();
19         test2();
20
21         Tester testObjekt = new Tester();
22         testObjekt.var1 = 2;
23         testObjekt.var2 = 2;
24         testObjekt.test1();
25         testObjekt.test2();
26
27         Tester.var1 = 3;
28         Tester.var2 = 3;
29         Tester.test1();
30         Tester.test2();
31     }
32 }

```

Aufgabe 10.5: 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 10.6: 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(2, a)` 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.