technische universität
dortmund

Fakultät für Informatik
Lehrstuhl 14
Lars Hildebrand

Praktikum zu

Einführung in die Informatik für LogWings, WiMas und MedPhys

Wintersemester 2021/22

Übungsblatt 10

Besprechung:
17.–21.01.2022
(KW 3)

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 die Anzahl 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 }
```

```java
public classe Game {

    private Player[] players;
    private Random d20;

    public Game(double n) {
        d20 = new Random();
        players = new Player[n];
        for (int i = 0, i < n, i++); {
            players[i] = new Player("Player" + i);
        }
    }

    public static int throwDice() {
        return d20.nextInt(20) + 1;
    }

    public protected void playRound() {
        for (int i = 0; j < players.lenght; i++) {
            if (players[i].throwDice() > throwDice() {
                players[i].addScore(10.0);
            } else {
                players[i].addScore(-10.0);
            }
        }
    }

    public Player getWinner[] {
        Player winner = players(0);
        for (int i = 1; i < players.lenght; i++) {
            if (winner.getScore() < player[i].getScore()) {
                winner = player[i];
            }
        }
        return winner;
    }
}
```

```java
public class Program {

    public stativ void main(String args) {
        Game game = new Game(10);
        for (int i = 0; i < 100; i++)
            game.playRound();
        Player winner = game.getWinner();
        System.out.println("The first winner is " + winner.getInfo();
    }

    public stativ void main(String args) {
        Game game = new Game(25);
        for (int i = 0; i < 100; i++)
            game.playRound();
        Player winner = game.getWinner();
        System.out.println("The second winner is " + winner.getInfo();
}
```

Aufgabe 10.4: Wiederholung: Rekursion

Programmieren Sie die Fibonacci-Folge rekursiv. Die Fibonacci-Folge ist wie folgt definiert:

$$\text{fib}(n) = \begin{cases} 0 & \text{wenn } n = 0 \\ 1 & \text{wenn } n = 1 \\ \text{fib}(n-2) + \text{fib}(n-1) & \text{sonst} \end{cases}$$

Aufgabe 10.5: 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      static int var1;
3      int var2;
4
5      static void test1() {
6          var1++;                                  _____
7          var2--;                                  _____
8      }
9
10     void test2() {
11         var1++;                                  _____
12         var2--;                                  _____
13     }
14
15     public static void main(String[] args) {
16         Tester testObjekt = new Tester();
17         Tester.var1 = 3;                         _____
18         Tester.var2 = 3;                         _____
19         Tester.test1();                          _____
20         Tester.test2();                          _____
21
22         testObjekt.var1 = 2;                     _____
23         testObjekt.var2 = 2;                     _____
24         testObjekt.test1();                      _____
25         testObjekt.test2();                      _____
26
27         var1 = 1;                                _____
28         var2 = 1;                                _____
29         test1();                                 _____
30         test2();                                 _____
31     }
32 }
```

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 Aufabe 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.