

Quellcode Serie 04 - Programmieren 1

Implementationsaufgabe 1:

Klasse Coin.java

```
src > kap4 > Coin.java > Coin > flip()
1  package kap4;
2  /*
3   Jara Zihlmann(20-117-032)
4   Vithusan Ramalingam (21-105-515)
5   Jan Ellenberger (21-103-643)
6   */
7  public class Coin {
8
9      //Variabeln zum ermitteln ob Kopf oder Zahl
10     //geworfen wird, Number wird nicht verwendet
11     //nur Vollständigkeitshalber aufgeführt
12     //für allfällige Erweiterungen dieses Programms
13     private final int HEAD = 0;
14     private final int NUMBER = 1;
15
16     private int seite;
17
18     //Münze direkt Werfen um einen Anfangswert
19     //der Münze zu kriegen
20     public Coin(){
21         flip();
22     }
23
24     // 50/50 Chance ob 0 oder 1 errechnet wird in random
25     // wie oben definiert 0 = Kopf
26     // 1 = Zahl
27     public void flip(){
28         seite = (int) (Math.random() * 2);
29     }
30
```

Zeile 24, Spalte 58 Leerzeichen: 4 UTF-8 CRLF Java

```
src > kap4 > Coin.java > Coin > toString()
30
31 // boolean Wert ob Kopf oben ist
32 public boolean isHead ()
33 {
34     return (seite == HEAD);
35 }
36 // vergleich der obenliegenden Seite
37 // um zu schauen ob die Münzen gleich sind
38 public boolean equals(Coin coin2){
39
40     if (this.seite == coin2.seite)
41     |   return true;
42     else
43     |   return false;
44 }
45
46 // toString Methode für Ausgabe
47 public String toString(){
48     {
49         String seiteOben;
50
51         seiteOben = (seite == HEAD)? seiteOben = "Kopf": "Zahl";
52
53         return seiteOben;
54     }
55 }
56 }
57
```

Zeile 51, Spalte 70 Leerzeichen: 4 UTF-8 CRLF Java

Klasse CoinRace.java

```
src > kap4 > CoinRace.java > CoinRace > main(String[])
1  package kap4;
2  /*
3   Jara Zihlmann(20-117-032)
4   Vithusan Ramalingam (21-105-515)
5   Jan Ellenberger (21-103-643)
6   */
7  public class CoinRace {
8      public static void main(String[] args) {
9          // Ziel ist es 3 mal hintereinander Kopf zu werfen
10         final int GOAL = 3;
11         //Zähler für die Münzen und Runden
12         int coin1Head=0;
13         int coin2Head=0;
14         int round = 0;
15
16         Coin c1 = new Coin();
17         Coin c2 = new Coin();
18         //toString und equals testen
19         System.out.println(c1.toString());
20         System.out.println(c2.toString());
21         System.out.println(c1.equals(c2));
22         System.out.println("toString und equals() funktionieren");
23
24         //solange das Ziel von 3mal hintereinander Kopf zu erhalten
25         //nicht erfüllt ist, wird die Münze nochmal geworfen
26         while (coin1Head < GOAL && coin2Head < GOAL)
27         {
28             c1.flip();
29             c2.flip();
```

Zeile 53, Spalte 7 Leerzeichen: 3 UTF-8 CRLF Java

```
src > kap4 > CoinRace.java > CoinRace > main(String[])
26     while (coin1Head < GOAL && coin2Head < GOAL)
27     {
28         c1.flip();
29         c2.flip();
30         round ++;
31
32         //Rundenanzahl und Ergebniss des Wurf wird ausgegeben
33         System.out.println ("Runde " + round + ": \t"
34         + c1.toString() + " " + c2.toString());
35
36         //Zähler erhöhen bei aufeinanderfolgenden Würfeln
37         //mit dem Ergebnis Kopf, sonst wird der
38         //Zähler zurückgesetzt
39         coin1Head = (c1.isHead()) ? coin1Head+1 : 0;
40         coin2Head = (c2.isHead()) ? coin2Head+1 : 0;
41     }
42
43     //Wird das Ziel erreicht, muss noch entschieden werden welche Münzen
44     //gewonnen hat
45     if (coin1Head < GOAL)
46         System.out.println ("Münze 2 gewinnt!");
47     else
48         if (coin2Head < GOAL)
49             System.out.println ("Münze 1 gewinnt!");
50         else
51             //Spezialfall dass beide Münzen 3 mal hintereinander Kopf haben
52             System.out.println ("Unentschieden!");
53 }
54 }
55
```

Zeile 51, Spalte 75 Leerzeichen: 3 UTF-8 CRLF Java

Implementationsaufgabe 2:**Klasse PairOfDice:**

```
src > kap4 > PairOfDice.java > PairOfDice > scanPoints()
1  package kap4;
2  /*
3   Jara Zihlmann(20-117-032)
4   Vithusan Ramalingam (21-105-515)
5   Jan Ellenberger (21-103-643)
6   */
7  import java.util.Scanner;
8
9  public class PairOfDice {
10     //2 Dice objekte
11     public static Dice d1 = new Dice();
12     public static Dice d2 = new Dice();
13     public static void main(String[] args) {
14         //alle vier Methoden testen
15         scanPoints();
16         printPoints();
17
18         rollDice();
19         //nochmals ausgeben um zu sehen
20         //ob Würfel geändert wurden
21         printPoints();
22         System.out.println(getTotalPoints());
23     }
24
25     //setzen der Punkte nach Benutzereingabe
26     public static void scanPoints(){
27         Scanner scn = new Scanner(System.in);
28         System.out.println("Geben Sie eine Punktzahl"
29         + "für ihren ersten Würfel ein".
```

Zeile 28, Spalte 55 Leerzeichen: 4 UTF-8 CRLF Java

```
src > kap4 > PairOfDice.java > PairOfDice > getTotalPoints()
28     System.out.println("Geben Sie eine Punktzahl"
29     + "für ihren ersten Würfel ein");
30     int i = scn.nextInt();
31     d1.setPoints(i);
32     System.out.println("Geben Sie eine Punktzahl"
33     + "für ihren zweiten Würfel ein");
34     int j = scn.nextInt();
35     d2.setPoints(j);
36 }
37 //ausgabe der Punkte auf Würfel
38 public static void printPoints(){
39     String printPoints1 = "Würfel 1: ";
40     String printPoints2 = "Würfel 2: ";
41
42     System.out.println(printPoints1 + d1.getPoints());
43     System.out.println(printPoints2 + d2.getPoints());
44 }
45 //Würfel werfen
46 public static void rollDice(){
47     System.out.println("Würfel werden neu geworfen:");
48     d1.roll();
49     d2.roll();
50 }
51 //Punkte auf Würfel zusammenrechnen und zurückgeben
52 public static int getTotalPoints(){
53     int getTotalPoints = d1.getPoints() + d2.getPoints();
54     return getTotalPoints;
55 }
56 }
```

Zeile 53, Spalte 62 Leerzeichen: 4 UTF-8 CRLF Java