

# Übungsblatt 3

Adrian Schollmeyer

## Aufgabe 1

- (a)    • 42, Hello!
- Hello!
- (b)    • Zeile 5 zu: fun(a, new String[] { "" });

## Aufgabe 2

Listing 1: AnagramDetector.java

```
1 package me.adrian.ha3;
2
3 import java.util.HashMap;
4
5 public class AnagramDetector
6 {
7     static public boolean check(String lhs, String rhs)
8     {
9         String lhsLc = lhs.toLowerCase();
10        String rhsLc = rhs.toLowerCase();
11
12        if (lhsLc.length() != rhsLc.length())
13            return false;
14
15        HashMap<Character, Integer> lc = new HashMap<>(), rc = new HashMap<>();
16        for (int i = 0; i < lhsLc.length(); i++) {
17            char cl = lhsLc.charAt(i);
18            char cr = rhsLc.charAt(i);
19
20            if (lc.containsKey(cl))
21                lc.put(cl, 1 + lc.get(cl));
22            else
23                lc.put(cl, 1);
```

```
24
25     if (rc.containsKey(cr))
26         rc.put(cr, 1 + rc.get(cr));
27     else
28         rc.put(cr, 1);
29 }
30
31     for (Character c : lc.keySet()) {
32         if (!rc.containsKey(c))
33             return false;
34
35         if (lc.get(c).intValue() != rc.get(c).intValue())
36             return false;
37     }
38
39     return true;
40 }
41
42 static public void main(String[] args)
43 {
44     System.out.println("Eingaben sind" + (check(args[0], args[1]) ? "" : " "
45         keine") + " Anagramme");
46 }
```

## Aufgabe 3

Listing 2: MultiDimSearch.java

```
1 public static void find1(char[] [] a, char[] [] m)
2 {
3     int curX, curY;
4     boolean possibleMatch = false;
5     for (int i = 0; i < a.length; i++) // Zeilen a
6     {
7         for (int j = 0; j < a[i].length; j++) // Spalten a
8         {
9             if (a[i][j] != m[0][0])
10                 continue;
11
12             possibleMatch = true;
13             curX = i;
14             curY = j;
15             try {
16                 for (int k = 0; k < m.length; k++) // Zeilen m
17                 {
18                     for (int l = 0; l < m[k].length; l++) // Spalten m
19                     {
20                         if (possibleMatch && a[i + k][j + l] != m[k][l])
21                             possibleMatch = false;
22
23             }
24         }
25     }
26 }
```

```
22         }
23     }
24 } catch (ArrayIndexOutOfBoundsException e) {
25     possibleMatch = false;
26 }
27
28 if (possibleMatch) {
29     System.out.println("(" + curX + ", " + curY + ")");
30     return;
31 }
32 }
33 System.out.println("(-1, -1)");
34 }
35 }
36
37 public static void find2(char[][] a, char[][] m)
38 {
39     int curX, curY;
40     boolean possibleMatch = false;
41     for (int i = 0; i < a.length; i++) // Zeilen a
42     {
43         for (int j = 0; j < a[i].length; j++) // Spalten a
44         {
45             if (!matchOrWildcard(a[i][j], m[0][0]))
46                 continue;
47
48             possibleMatch = true;
49             curX = i;
50             curY = j;
51             try {
52                 for (int k = 0; k < m.length; k++) // Zeilen m
53                 {
54                     for (int l = 0; l < m[k].length; l++) // Spalten m
55                     {
56                         if (possibleMatch && !matchOrWildcard(a[i + k][j + l],
57                             m[k][l]))
58                             possibleMatch = false;
59                     }
60                 }
61             } catch (ArrayIndexOutOfBoundsException e) {
62                 possibleMatch = false;
63             }
64             if (possibleMatch) {
65                 System.out.println("(" + curX + ", " + curY + ")");
66                 return;
67             }
68         }
69     }
```

```
70     System.out.println("(-1, -1)");
71 }
72
73 public static boolean matchOrWildcard(char lhs, char rhs)
74 {
75     if (rhs == '*')
76         return true;
77     else
78         return lhs == rhs;
79 }
```

**Ausgabe:**

(25, 34)

(-1, -1)

(30, 34)

## Aufgabe 4

```
1: bin( $n, k$ ) =
2: if  $n < k$  then 0
3: else
4:   if  $n = k \vee k = 0$  then 1
5:   else
6:     bin( $n - 1, k - 1$ ) + bin( $n - 1, k$ )
7:   end if
8: end if
```

## Aufgabe 5

```
1: fib( $n$ ) =
2: if  $n = 0 \vee n = 1$  then
3:   1
4: else
5:   h( $n, 1, 1, 1$ )
6: end if
7: h( $n, i, e, f$ ) =
8: if  $i = n$  then
9:    $f$ 
10: else
11:   h( $n, i + 1, f, e + f$ )
12: end if
```