

Übungen zu regulären Ausdrücken

- (a) Gegeben ist eine Sprache $L \subset \Sigma^*$ mit $\Sigma = \{a, b\}$. Zu der Sprache L gehören alle Wörter, die die Zeichenfolge abba beinhalten.

Geben Sie einen regulären Ausdruck für diese Sprache an („klassischer“ regulärer Ausdruck).

$(a|b)^*abba(a|b)^*$

```
5 static String regexABBA = "(a|b)*abba(a|b)*";
```

- (b) Gebe möglichst einfache reguläre Ausdrücke für die folgenden Sprachen $L_x \subset \Sigma^*$ mit $\Sigma = \{a, b\}$ und $x \in \{1, 2, 3\}$ („klassischer“ regulärer Ausdruck).

$L_1 = \{x | x \text{ beinhaltet eine gerade Anzahl von } a\}$

$b^*(ab^*ab^*)^*$

```
6 static String regexGeradeA = "b*(ab*ab*)*"; // Epsilon aa aaaa
  ↳ abba bbba aabb bbb
```

$L_2 = \{x | x \text{ beinhaltet eine ungerade Anzahl von } b\}$

$a^*ba^*(ba^*ba^*)^*$

```
7 static String regexUngeradeB = "a*ba*(ba*ba*)*"; // Epsilon b
  ↳ bbb abababa
```

$L_3 = \{x | x \text{ beinhaltet an seinen geradzahigen Positionen ausschließlich } a\}$

$((a|b)a)^*(a^*|b)$

```
8 static String regexGeradzahligA = "((a|b)a)*(a*|b)"; // aa ba
  ↳ aab b
```

- (c) Geben Sie einen regulären Ausdruck an, der eine syntaktisch gültige E-Mail-Adresse erkennt. (mindestens 1 Zeichen (Groß-/Kleinbuchstabe oder Zahl) vor dem @; mindestens 1 Zeichen (Groß-/Kleinbuchstabe oder Zahl) nach dem @; alle E-Mail-Adressen sollen auf .de oder .com enden.

```
9 static String regexEMAIL = "\\w+@\\w+\\.\\.(de|com)";
```

```
3 public class TestRegularExpressions {
4     // Hier bitte Lösungen der Aufgaben eintragen.
5     static String regexABBA = "(a|b)*abba(a|b)*";
6     static String regexGeradeA = "b*(ab*ab*)*"; // Epsilon aa aaaa abba bbba
7     static String regexUngeradeB = "a*ba*(ba*ba*)*"; // Epsilon b bbb abababa
8     static String regexGeradzahligA = "((a|b)a)*(a*|b)"; // aa ba aab b
9     static String regexEMAIL = "\\w+@\\w+\\.\\.(de|com)";
10    // Wenn die Lösungen stimmen, geben alle Tests true aus
11
12    // Alternativen:
13    // static String regexGeradzahligA = "((a|b)a)*((a|b)|)"; // aa ba aab b
```

```

14
15 public static void main(String[] args) {
16     testregexABBA();
17     testregexGeradeA();
18     testregexUngeradeB();
19     testregexGeradzahligA();
20     testregexEMAIL();
21 }
22
23 public static void testregexABBA() {
24     boolean[] b = new boolean[7];
25     b[0] = "abba".matches(regexABBA);
26     b[1] = !"aba".matches(regexABBA);
27     b[2] = "abbaabbaabba".matches(regexABBA);
28     b[3] = "abababbaaabaabaaba".matches(regexABBA);
29     b[4] = !"ab".matches(regexABBA);
30     b[5] = !"bbb".matches(regexABBA);
31     b[6] = !"".matches(regexABBA);
32     if (b[0] & b[1] & b[2] & b[3] & b[4] & b[5] & b[6]) {
33         System.out.println("Alle ABBA-Tests bestanden. Dein RegEx stimmt!");
34     } else {
35         for (int i = 0; i < b.length; i++) {
36             if (!b[i]) {
37                 System.out.println("Test mit dem Index" + i +
38                     ↪ " leider nicht bestanden.");
39             }
40         }
41     }
42
43 public static void testregexGeradeA() {
44     boolean[] b = new boolean[7];
45     b[0] = "aa".matches(regexGeradeA);
46     b[1] = !"aaa".matches(regexGeradeA);
47     b[2] = "abbaabbaabba".matches(regexGeradeA);
48     b[3] = !"abababbaaabaabaaba".matches(regexGeradeA);
49     b[4] = !"ab".matches(regexGeradeA);
50     b[5] = "bbb".matches(regexGeradeA);
51     b[6] = "".matches(regexGeradeA);
52     if (b[0] & b[1] & b[2] & b[3] & b[4] & b[5] & b[6]) {
53         ↪ System.out.println("Alle GeradeA-Tests bestanden. Dein RegEx stimmt!");
54     } else {
55         for (int i = 0; i < b.length; i++) {
56             if (!b[i]) {
57                 System.out.println("Test mit dem Index" + i +
58                     ↪ " leider nicht bestanden.");
59             }
60         }
61     }
62
63 public static void testregexUngeradeB() {
64     boolean[] b = new boolean[7];
65     b[0] = "b".matches(regexUngeradeB);
66     b[1] = !"bb".matches(regexUngeradeB);
67     b[2] = !"abbaabbaabba".matches(regexUngeradeB);
68     b[3] = "abababbaaabaabaaba".matches(regexUngeradeB);
69     b[4] = "ab".matches(regexUngeradeB);
70     b[5] = "bbb".matches(regexUngeradeB);
71     b[6] = !"".matches(regexUngeradeB);

```

```

72     if (b[0] & b[1] & b[2] & b[3] & b[4] & b[5] & b[6]) {
73         ↪ System.out.println("Alle UngeradeB-Tests bestanden. Dein RegEx stimmt!");
74     } else {
75         for (int i = 0; i < b.length; i++) {
76             if (!b[i]) {
77                 System.out.println("Test mit dem Index" + i +
78                     ↪ " leider nicht bestanden.");
79             }
80         }
81     }
82 }
83
84 public static void testregexGeradzahligA() {
85     boolean[] b = new boolean[7];
86     b[0] = !"ab".matches(regexGeradzahligA);
87     b[1] = "b".matches(regexGeradzahligA);
88     b[2] = "babab".matches(regexGeradzahligA);
89     b[3] = !"bababaab".matches(regexGeradzahligA);
90     b[4] = "ba".matches(regexGeradzahligA);
91     b[5] = "aaa".matches(regexGeradzahligA);
92     b[6] = "".matches(regexGeradzahligA);
93     if (b[0] & b[1] & b[2] & b[3] & b[4] & b[5] & b[6]) {
94         ↪ System.out.println("Alle GeradzahligeA-Tests bestanden. Dein RegEx stimmt!");
95     } else {
96         for (int i = 0; i < b.length; i++) {
97             if (!b[i]) {
98                 System.out.println("Test mit dem Index" + i +
99                     ↪ " leider nicht bestanden.");
100             }
101         }
102     }
103 }
104
105 public static void testregexEMAIL() {
106     boolean[] b = new boolean[7];
107     b[0] = "3@s.de".matches(regexEMAIL);
108     b[1] = !"@0.de".matches(regexEMAIL);
109     b[2] = "asdf@asdf.com".matches(regexEMAIL);
110     b[3] = !"@.de".matches(regexEMAIL);
111     b[4] = "s@1.com".matches(regexEMAIL);
112     b[5] = !"a@a".matches(regexEMAIL);
113     b[6] = !"".matches(regexEMAIL);
114     if (b[0] & b[1] & b[2] & b[3] & b[4] & b[5] & b[6]) {
115         System.out.println("Alle Email-Tests bestanden. Dein RegEx stimmt!");
116     } else {
117         for (int i = 0; i < b.length; i++) {
118             if (!b[i]) {
119                 System.out.println("Test mit dem Index" + i +
120                     ↪ " leider nicht bestanden.");
121             }
122         }
123     }
124 }

```