

## 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.

Gib einen regulären Ausdruck für diese Sprache an.

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

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

$$L_1 = \{x | x \text{ beinhaltet eine gerade Anzahl von } a\}$$
$$.*(aa)+.* \text{ oder } (a|b)*(aa)(a|b)*$$
$$L_2 = \{x | x \text{ beinhaltet eine ungerade Anzahl von } b\}$$
$$.*b(bb)*.* \text{ oder } (a|b)*b(bb)*(a|b)*$$
$$L_3 = \{x | x \text{ beinhaltet an seinen geradzahligen Positionen ausschließlich } a\}$$
$$((a|b)a)^*$$

- (c) Gib einen regulären Ausdruck 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.

[a-zA-Z0-9]+@[a-zA-Z0-9]+(de|com)

```

3 public class TestRegularExpressionns {
4     // Hier bitte Loeungen der Aufgaben eintragen.
5     static String regexABBA = "(a|b)*abba(a|b)*";
6     static String regexGeradeA = "(b+(aa)*)*";
7     static String regexUngeradeB = ".*b(bb)*.*";
8     static String regexGeradzahligA = "((a|b)a)*";
9     static String regexEMAIL = "[a-zA-Z0-9]+@[a-zA-Z0-9]+\\.\\.(de|com)";
10    // Wenn die Losungen stimmen, geben alle Tests true aus
11
12    public static void testregexABBA() {
13        boolean[] b = new boolean[7];
14        b[0] = "abba".matches(regexABBA);
15        b[1] = !"aba".matches(regexABBA);
16        b[2] = "abbaabbaabba".matches(regexABBA);
17        b[3] = "abababbaaabaabaaba".matches(regexABBA);
18        b[4] = !"ab".matches(regexABBA);
19        b[5] = !"bbb".matches(regexABBA);
20        b[6] = !"".matches(regexABBA);
21        if (b[0] & b[1] & b[2] & b[3] & b[4] & b[5] & b[6]) {
22            System.out.println("Alle ABBA-Tests bestanden. Dein RegEx stimmt!");
23        } else {
24            for (int i = 0; i < b.length; i++) {
25                if (!b[i]) {
26                    System.out.println("Test mit dem Index" + i +
27                        "\u2192 " + " leider nicht bestanden.");
28                }
29            }
30        }
31    }
32}

```

```

27     }
28   }
29 }
30 }
31
32 public static void testregexGeradeA() {
33     boolean[] b = new boolean[7];
34     b[0] = "aa".matches(regexGeradeA);
35     b[1] = "aaa".matches(regexGeradeA);
36     b[2] = "abbaabbaabba".matches(regexGeradeA);
37     b[3] = "abababbabaaabaaaba".matches(regexGeradeA);
38     b[4] = "ab".matches(regexGeradeA);
39     b[5] = "bbb".matches(regexGeradeA);
40     b[6] = "".matches(regexGeradeA);
41     if (b[0] & b[1] & b[2] & b[3] & b[4] & b[5] & b[6]) {
42
43         ↪ System.out.println("Alle GeradeA-Tests bestanden. Dein RegEx stimmt!");
44     } else {
45         for (int i = 0; i < b.length; i++) {
46             if (!b[i]) {
47                 System.out.println("Test mit dem Index" + i +
48                     ↪ " leider nicht bestanden.");
49             }
50         }
51     }
52 }
53
54 public static void testregexUngeradeB() {
55     boolean[] b = new boolean[7];
56     b[0] = "b".matches(regexUngeradeB);
57     b[1] = "bb".matches(regexUngeradeB);
58     b[2] = "abbaabbaabba".matches(regexUngeradeB);
59     b[3] = "abababbabaaabaaaba".matches(regexUngeradeB);
60     b[4] = "ab".matches(regexUngeradeB);
61     b[5] = "bbb".matches(regexUngeradeB);
62     b[6] = "!".matches(regexUngeradeB);
63     if (b[0] & b[1] & b[2] & b[3] & b[4] & b[5] & b[6]) {
64
65         ↪ System.out.println("Alle UngeradeB-Tests bestanden. Dein RegEx stimmt!");
66     } else {
67         for (int i = 0; i < b.length; i++) {
68             if (!b[i]) {
69                 System.out.println("Test mit dem Index" + i +
70                     ↪ " leider nicht bestanden.");
71             }
72         }
73     }
74 }
75
76 public static void testregexGeradzahligA() {
77     boolean[] b = new boolean[7];
78     b[0] = "ab".matches(regexGeradzahligA);
79     b[1] = "b".matches(regexGeradzahligA);
80     b[2] = "babab".matches(regexGeradzahligA);
81     b[3] = "!bababaab".matches(regexGeradzahligA);
82     b[4] = "ba".matches(regexGeradzahligA);
83     b[5] = "aaa".matches(regexGeradzahligA);
84     b[6] = "".matches(regexGeradzahligA);
85     if (b[0] & b[1] & b[2] & b[3] & b[4] & b[5] & b[6]) {
86
87         ↪ System.out.println("Alle GeradzahligeA-Tests bestanden. Dein RegEx stimmt!");
88     }
89 }

```

```

83     } else {
84         for (int i = 0; i < b.length; i++) {
85             if (!b[i]) {
86                 System.out.println("Test mit dem Index" + i +
87                                     ↳ " leider nicht bestanden.");
88             }
89         }
90     }
91
92     public static void testregexEMAIL() {
93         boolean[] b = new boolean[7];
94         b[0] = "3@s.de".matches(regexEMAIL);
95         b[1] = !"@0.de".matches(regexEMAIL);
96         b[2] = "asdf@asdf.com".matches(regexEMAIL);
97         b[3] = !"@.de".matches(regexEMAIL);
98         b[4] = "s@1.com".matches(regexEMAIL);
99         b[5] = !"a@a".matches(regexEMAIL);
100        b[6] = !"".matches(regexEMAIL);
101        if (b[0] & b[1] & b[2] & b[3] & b[4] & b[5] & b[6]) {
102            System.out.println("Alle Email-Tests bestanden. Dein RegEx stimmt!");
103        } else {
104            for (int i = 0; i < b.length; i++) {
105                if (!b[i]) {
106                    System.out.println("Test mit dem Index" + i +
107                                        ↳ " leider nicht bestanden.");
108                }
109            }
110        }
111
112        public static void main(String[] args) {
113            testregexABBA();
114            testregexGeradeA();
115            testregexUngeradeB();
116            testregexGeradzahligA();
117            testregexEMAIL();
118        }
119    }

```