

PalindromPruefer.java

```

1
2
3
4 /**
5  * &Uuml;berpr&uuml;fung der Eigenschaft, ein Palindrom zu sein.
6  */
7 public class PalindromPruefer {
8
9 // static {
10 //     char c = 65;
11 //     for ( ; c <= 'Z'; c++) {
12 //         System.out.printf("%8s -> %s\n",
13 //             Integer.toBinaryString(lower(c)), ""+lower(c));
14 //     }
15 //     c = 'a';
16 //     for ( ; c <= 'z'; c++) {
17 //         System.out.printf("%8s -> %s\n", Integer.toBinaryString(c),
18 //             ""+c);
19 //     }
20 // }
21 /**
22  * &Uuml;berpr&uuml;fe, ob die &uuml;bergebene Zeichenkette
23  * ein Palindrom ist. Hierbei soll nicht zwischen Gross- und
24  * Kleinbuchstaben unterschieden werden.
25  * @param eingabe Zu testende Zeichenkette.
26  * @return <CODE>true</CODE>, wenn die Zeichenkette ein Palindrom
27  * ist, <CODE>false</CODE> sonst.
28  */
29 public boolean istPalindrom(String eingabe) {
30     // Auf falsche eingaben überprüfen.
31     if(eingabe == null)
32         throw new IllegalArgumentException("Argument darf nicht null
33 sein.");
34
35     // Chararray des Strings abrufen, weil ich die charAt(int )
36     Methode nicht aufrufen will.
37     char[] c_eingabe = eingabe.toCharArray();
38
39     // Das Array auf eine überprüfen indem zuerst die beiden äußeren
40     Zeichen auf gleichheit geprüft werden.
41     // Danach werden die Zeichen eins weiter innen überprüft. Dies
42     wird wiederholt bis
43     // das gesamte Array geprüft ist.
44     // Falls das Array eine ungerade anzahl von Elementen besitzt wird
45     das mittlere Element nicht
46     // geprüft. Dies ist egal, da das Element in der Mitte steht und
47     deswegen die Symetrie nicht stört.
48     for (int i = 0; i < c_eingabe.length/2; i++) {
49         //Beide Zeichen werden zu einem kleinbuchstaben konvertiert
50         und verglichen.
51         // Alternrnativ koennte die Character.toLowerCase() an der
52         Stelle von lower() verwendet werden.
53         if(lower(c_eingabe[i]) != lower(c_eingabe[c_eingabe.length-1 -

```

PalindromPruefer.java

```

i)))
44         return false; // Gibt false zurück sobald ein Fehler
         gefunden wird.
45     }
46     // Falls kein Fehler gefunden wurde wird true zurückgegeben.
47     return true;
48 }
49
50 /**
51  * &Uuml;berpr&uuml;fe, ob die &uuml;bergebene Zeichenkette
52  * ein Satzpalindrom ist. Hierbei soll nicht zwischen Gross- und
53  * Kleinbuchstaben unterschieden werden.
54  * @param eingabe Zu testende Zeichenkette.
55  * @return <CODE>true</CODE>, wenn die Zeichenkette ein Palindrom
56  * ist, <CODE>false</CODE> sonst.
57  */
58 public boolean istSatzPalindrom(String eingabe) {
59     // Auf falsche eingaben überprüfen.
60     if(eingabe == null)
61         throw new IllegalArgumentException("Argument darf nicht null
        sein.");
62
63     // Entf&uuml;rnt alle whitespace-Zeichen und alle Satzzeichen und
        überprüft den String.
64     return istPalindrom(
65         eingabe.replaceAll("\\s|\\p{Punct}", ""));
66
67 }
68
69 /**
70  * Wandelt ein Großbuchstaben in einen Kleinbuchstaben um.
71  * Die Nummer des Zeichens muss aus der ASCII-Tabelle sein,
72  * weil die Konvertierung sonst fehlschlägt.
73  * @param c
74  * @return
75  */
76 public static char lower(char c) {
77
78     // Damit die Zeichen konvertiert werden sollen müssen die Zeichen
        im Beireich der Buchstaben liegen.
79     // Dies wird hier geprüft.
80     if(c < 'A' || c > 'Z' && c < 'a' || c > 'z')
81         return c;
82
83     // Großbuchsaben: 10xxxxx
84     // Kleinbuchstaben: 11xxxxx
85     // Wandelt Zeichen mit or um. Alternativ koennte man auch
86     // eine Subrtaktion oder Addition nutzen.
87     return (char) (c | 0b1100000);
88 }
89
90 }
91

```