

INLOOP Softwaretechnologie, String Processing - Leetspeek

HENRY HAUSTEIN

vollständiger Code

Datei Leet.java

```
1  class Leet {
2      public static String toLeet(String normal) {
3          String l1 = normal.replace("elite", "1337");
4          String l2 = l1.replace("cool", "k3w1");
5          String l3 = l2.replace("!", "!!!11");
6          String l4 = l3.replace("ck", "xx");
7          String l5 = l4.replace("ers", "0rz");
8          String l6 = l5.replace("er", "0rz");
9          String l7 = l6.replace("en", "n");
10         String l8 = l7.replace("e", "3");
11         String l9 = l8.replace("t", "7");
12         String l10 = l9.replace("o", "0");
13         String l11 = l10.replace("a", "@");
14
15         return l11;
16     }
17
18     public static String[] allToLeet(String[] normals) {
19         String[] leets = new String[normals.length];
20         for (int i = 0; i < normals.length; i++) {
21             leets[i] = toLeet(normals[i]);
22         }
23
24         return leets;
25     }
26 }
```

Erklärung

Wir führen immer eine Ersetzung durch und speichern das Zwischenergebnis in einer neuen Variable. Sicherlich kann man das auch direkter machen, in etwa so:

```
1  String leet = normal.replace("elite", "1337").replace("cool", "k3w1") ...;
```

In der Funktion `allToLeet()` erstellen wir uns zuerst ein Array der passenden Länge, was wir dann auch später zurückgeben wollen. In einer for-Schleife laufen wir dann durch das Array `normals` durch und füllen die entsprechenden Stellen in unserem `leets`-Array.