

Zahlumwandlung.java

```

1 /**
2  * Umrechnung einer Zahl im Bereich [1..3000] in das r mische
   Zahlssystem.
3  */
4 public class Zahlumwandlung {
5     private char[] symbole = new char[] {
6         'I',
7         'V',
8         'X',
9         'L',
10        'C',
11        'D',
12        'M'
13    };
14
15
16    /**
17     * Wandel eine Natuerliche Zahl aus dem Bereich 1 bis 3000 in eine
   Roemische Zahl um.
18     * @param n Zahl die umgewandelt werden soll.
19     * @return Zahl n in der Roemischen Schreibweise.
20     */
21    public String wandleZahlUm(int n) {
22        //Bereich ueberpruefen. (1 ... 3000)
23        if(n < 1 || n > 3000) {
24            throw new IllegalArgumentException(String.format("Die Nummer
   n=%d liegt nicht im Interval [1, 3000] liegen. ", n));
25        }
26        //System.out.println(n);
27        String nummerDec = n + "";
28        String nummer = "";
29        for (int i = 1, indexSymbols = 0; i <= nummerDec.length(); i++) {
30            // Wert der i-er Stelle von n berechnen.
31            int val = nummerDec.charAt(nummerDec.length() - i) - 48; //48
   = ascii von '0'
32
33            String numberSymbols = "";
34            //Weil gleiche Zeichen fuer die Zahl nicht mehr als 3 mal
   nebeneinander stehen duerfen muessen die Faelle val > 3 && < 9,
35            // val == 9 und val <= 3 unterschieden werden.
36            //
37            if(val > 3 && val < 9) {
38                int anzahlPrefix = (5-val); // Die anzahl der zu
   Subtrahierenden Zeichen wird berechnet.
39
40                //String fuer die i-er Stelle zusammensetzen.
41                numberSymbols = mul(symbole[indexSymbols], anzahlPrefix)
   + //Prefix (Zeichen die abgezogen werden muessen.
42                    symbole[indexSymbols+1] +
43                    //Zeichen der hoeheren einheit.
44                    mul(symbole[indexSymbols], val-5); //
   Aufuellen der fehlenden Werte um den Wert

```

Zahlumwandlung.java

```

    // der i-ten Stelle zu erreichen.
45
46     }
47     //Weil fuer den Wert  $9 \cdot 10^i$  wieder vier Werte der
    kleinsten Einheit verwendet werden muessen muss der String
48     //mit dem naechst hoeheren Zehner gebildet werden.
49     else if(val == 9) {
50         numberSymbols = symbole[indexSymbols]
    +""+symbole[indexSymbols+2];
51     }else {
52         //Wert kann mit  $val \cdot 10^{(i-1)}$  gebildet
    werden.
53         numberSymbols = mul(symbole[indexSymbols],val);
54     }
55
56     //Setzt das naechst Zeichen fuer die Basiseinheit auf den
    naechsten Zehner.
57     indexSymbols += 2;
58     //System.out.printf("%d | %d -> %d: %s\n",indexSymbols, i, val,
    numberSymbols);
59     nummer = numberSymbols+nummer;
60 }
61
62
63
64     return nummer;
65 }
66
67 /**
68  * Wiederholt ein Zeichen c n mal.
69  * @param s Zeichen das wiederholt werden soll.
70  * @param number Anzahl der Wiederholungen.
71  * @return
72  */
73 public String mul(char c, int number) {
74     String retVal = ""; //Damit der String nicht mit null startet.
75     for(int i = 0; i < number; i++) {
76         retVal+=c;
77     }
78     return retVal;
79 }
80
81 // public String mul(String s, int number) {
82 //     String retVal = ""; //Damit der String nicht mit null startet.
83 //     for(int i = 0; i < number; i++) {
84 //         retVal+=s;
85 //     }
86 //     return retVal;
87 // }
88 }
89
```