

```

package zeichenketten;

// Die wichtigsten Zeichenkettenmethoden

public class Zeichenketten
{
    public static void main(String[] args)
    {
        // Zeichenkette erzeugen

        String s1 = "Das ist ein Beispiel";

        String s2 = new String("Das ist auch ein beispiel");

        // Laenge einer Zeichenkette ermitteln

        int a = s1.length();

        // Einzeleines Zeichen aus einer Zeichenkette extrahieren (Kopie!)

        char c = s1.charAt(2); // Zeichen ab Indexposition 2 aus dem String
                                // s1 extrahieren. Der Index laeuft von 0 bis
                                // s1.length()-1

        // Teilstring aus einer Zeichenkette extrahieren

        String s3 = s1.substring(4,7); // Extrahiert den Substring ab Index-
                                        // position 4 bis Indexposition 6 (!)
                                        // aus dem String s1

        String s4 = " Das ist einfach ";
        String s5 = s4.trim(); // trim() liefert den String, der entsteht,
                                // wenn auf beiden Seiten des Strings s4
                                // die Leerzeichen entfernt werden. Als
                                // Leerzeichen gelten hier alle Zeichen
                                // mit einem ASCII-Code <= 32.

        // Vergleichen von Zeichenketten

        boolean f = s1.equals(s2); // Vergleicht die Zeichenkette s1 mit
                                    // der Zeichenkette s2. Sind die Zeichen-
                                    // ketten gleich, liefert die Methode
                                    // equals true, sonst false.

        int b = s1.compareTo(s2); // Vergleicht lexikalisch die Zeichenkette
                                    // s1 mit der Zeichenkette s2.
                                    // b > 0 : s1 ist groesser als s2
                                    // b < 0 : s1 ist kleiner als s2
                                    // b == 0: s1 und s2 sind gleichinde

        // Suchen in Zeichenketten
    }
}

```

```
int d = s1.indexOf("ein"); // Sucht im String s1 nach einem Teil-
// string "ein" und gibt den Index, ab
// dem "ein" das erste Mal vorkommt,
// zurueck
```

```
int e = s1.indexOf("ein",5); // Sucht nach Teilstring "ein" im String
// s1 erst ab Indexposition 5
```

```
// Ersetzen von Zeichenketten
```

```
String s6 = s1.toLowerCase(); // s6 ist der String, der entsteht, wenn
// alle Buchstaben von s1 in Kleinbuch-
// staben verwandelt werden. Besitz der
// String s1 keine umwandelbaren zeichen,
// liefert toLower() den Originalstring
```

```
String s7 = s1.toUpperCase(); //... das Gegenteil von toLower(), Kon-
// vertierung in Grossbuchstaben
```

```
String s8 = "eins und eins ist zwei, aber nicht immer";
```

```
String s9 = s8.replace("eins", "drei"); // Ersetzt jedes Vorkommen von
// "eins" im String s8 durch
// "drei" . ERgebnis steht im
// String s9
```

```
System.out.println(s9);
```

```
}
}
```