

Zeichenkettenverarbeitung:

Eine Zeichenreihe lässt sich als Objekt der Klasse **String** mit der Objektmethode **toCharArray()** in eine Reihung von Zeichen umwandeln.

Ist z.B. `s` ein String, so referenziert nach der Zuweisung **`char[] sarray = s.toCharArray();`** `sarray` die zum String `s` korrespondierende Reihung von Zeichen. Umgekehrt kann durch Aufruf des Konstruktors **`public String (char[] carray)`** aus einer Reihung von Zeichen ein String erzeugt werden.

Verwende nun diese beiden Operationen zur Lösung der folgenden Aufgaben:

a) Geben Sie eine Methode `public static String revert (String s)` an, die als Ergebnis die zu `s` revertierte, also in umgekehrter Reihenfolge geschriebene Zeichenreihe liefert (Bsp. `revert("nebel")` liefert als Ergebnis `"leben"`).

b) Eine Zeichenreihe `s` heißt Palindrom, wenn sie identisch zu ihrer revertierten Zeichenreihe ist (Bsp.: `"otto"`, `"ada"`). Geben Sie eine Java-Methode `public static boolean istPalindrom (String s)` an, die feststellt, ob die als Parameter übergebene Zeichenreihe `s` ein Palindrom ist oder nicht.

Aufgabe 1

Implementieren Sie eine Methode `istCharInString(String s, char c)`, welche überprüft, ob ein `char`-Zeichen `c` in einem String `s` vorkommt und einen entsprechenden Wahrheitswert zurückgibt.

Die Ausgabe könnte so aussehen:

```
Gib einen Text ein: Ein Text zum Üben ...
Gib ein Zeichen ein: u
u ist in "Ein Text zum Üben ..." enthalten.
```

Aufgabe 2

Implementieren Sie eine Methode, die die Anzahl eines gewünschten Zeichens innerhalb eines Strings bestimmt und zurückgibt.

Die Ausgabe könnte so aussehen:

```
Gib einen Text ein: Ein Text zum Üben ...
Gib das zu zählende Zeichen ein: n
u ist in "Ein Text zum Üben ..." 2x enthalten.
```

Aufgabe 3

Implementieren Sie eine Methode, die die letzten drei Zeichen eines eingegebenen Strings groß schreibt. Achten Sie darauf, dass die Zeichenkette auch weniger als drei Zeihen haben kann.

```
"Informatik" => "InformaTIK"  
"Hallo"      => "HaLLo"  
"Hi"         => "HI"
```

Aufgabe 4

Implementieren Sie ein Java-Programm, welches aus drei eingegebenen Zeichen alle möglichen Kombinationen dieser ausgibt.

Aufgabe 5

Implementieren Sie eine Methode, die alle Zeichen „x“ (und „X“) aus einem String entfernt, außer diese stehen an erster oder letzter Stelle.

```
"dexXxtxdx" => "detdx"  
"xabcdXDEfX" => "xabcdefx"
```