

## Übungen Kapitel 7 String

**In den folgenden Übungen geht es darum, sich mit den String Methoden vertraut machen. Also online String API lesen.**

### 7.1 (\*) Stringvergleich

(*compareTo*)

Schreiben Sie eine Java Anwendung, die die Klasse **String** und die Methode **compareTo** zum Vergleich von zwei durch den Bediener eingegebenen Strings benutzt. Geben Sie auf dem Bildschirm aus, ob der erste String größer, gleich oder kleiner ist.

**Lösungshinweis: die `compareTo` Methode existiert bereits in der Klasse `String`. Sie sollen diese nur benutzen!!**

### 7.2 (\*) Stringvergleich für Teilstrings

(*regionMatches*)

Schreiben Sie eine Java Anwendung, die die Klasse **String** und die Methode **regionMatches** zum Vergleich von zwei durch den Bediener eingegebenen Strings benutzt. Dabei soll das Programm die Anzahl der zu vergleichenden Character und den Startindex eingeben. Geben Sie auf dem Bildschirm aus, ob der erste String gleich oder ungleich ist. Ignorieren Sie Gross- und Kleinschreibung.

**Lösungshinweis: die `regionMatches` Methode existiert bereits in der Klasse `String`. Sie sollen diese nur benutzen!!**

### 7.3 (\*) Stringzerlegung

(*Tokenizer, split*)

- Schreiben Sie eine Java Anwendung, bei der eine Zeile Text eingegeben wird und dieser Text durch ein **StringTokenizer** Objekt in einzelne Token zerlegt wird. Geben Sie die Token in umgekehrter Reihenfolge aus.
- Ändern Sie die Anwendung von a. um, in dem Sie Tokens aus der Scanner Klasse direkt einlesen. Verwenden Sie dabei die existierende Methode **split** aus der **String** Klasse.

### 7.4 (\*) Zusatzaufgabe: Zeichenzählen

(*indexOf*)

Schreiben Sie eine Java Anwendung, bei der eine Zeile Text und ein Character eingegeben werden. Sie benutzen die **indexOf** Methode der Klasse **String**, um die Anzahl des Buchstabens im Text zu ermitteln und geben diese auf dem Bildschirm aus.

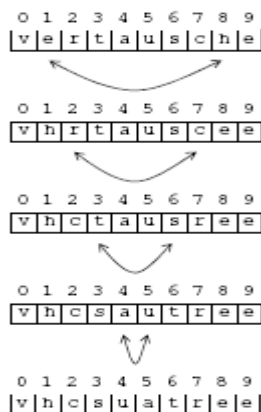
**Lösungshinweis: die `indexOf` Methode existiert bereits in der Klasse `String`. Sie sollen diese nur benutzen!!**

## 7.5 (\*\*) Effizientere Stringmanipulation

(StringBuffer)

Geübte Leser erkennen angeblich ganze Wörter (sozusagen bildlich) auf einmal. Dies wollen wir überprüfen und Text so verändern, dass die inneren Buchstaben in Wörtern vertauscht werden. Schreiben Sie dazu ein Programm, das Text von einem Satz von der Console einliest und die einzelnen Wörter wie folgt verändert:

- Der erste und der letzte Buchstabe bleiben unverändert
- Der 2. Buchstabe wird mit dem vorletzten vertauscht, der 3. Buchstabe mit dem vorvorletzten, usw.



Der veränderte Text soll auf der Console ausgegeben werden. Leerzeichen, Satzzeichen und Sonderzeichen sind dabei unverändert zu übernehmen.

Beispiel:

Originaltext: Da steh ich nun ich armer Thor und bin so klug als wie zuvor.

Veränderter Text: Da seth ich nun ich aemrr Tohr und bin so kulg als wie zovur.

### Lösungshinweise:

- Sie dürfen `StringTokenizer` dazu NICHT verwenden
- Verwenden Sie stattdessen die Klasse `StringBuffer`, um die Wörter zu verändern. Arbeiten Sie dabei mit den folgenden Methoden von `StringBuffer`:
 

```
public StringBuffer append(char c)
public int length()
public char charAt(int index)
public void setCharAt(int index, char ch)
public String toString()
```
- Gehen Sie bei der algorithmischen Lösung wie folgt vor:
  1. Beschreiben Sie zuerst in wenigen Worten die grundsätzliche Idee zu dem Verfahren.
  2. Beschreiben Sie das Verfahren in Prosa.
  3. Implementieren Sie das Verfahren in Java, indem Sie das in Prosa beschriebene Verfahren in Java umsetzen. Verwenden Sie Methoden für klar zu definierte Aufgaben.

### 7.6 (\*)Zusatzaufgabe: Anfangsbuchstabe

(weitere Stringmethode)

Schreiben Sie eine Java Anwendung, bei der eine Zeile Text eingegeben wird und der nur die mit ,b' beginnenden Wörter ausgibt.

### 7.7 (\*)Zusatzaufgabe: Endbuchstabe

(weitere Stringmethode)

Schreiben Sie eine Java Anwendung, bei der eine Zeile Text eingegeben wird und der nur die mit ,er' endenden Wörter ausgibt.

### 7.8 (\*)Zusatzaufgabe: Zufallssätze

Schreiben Sie eine Anwendung, die zufälligerweise Sätze generiert. Dazu verwenden Sie vier Arrays: **article**, **noun**, **verb**, und **preposition**. Aus jedem dieser Arrays nehmen Sie zufallsmässig ein Wort und stellen dann einen Satz daraus zusammen. Die einzelnen Wörter sollen durch blank getrennt sein. Jeder Satz beginnt mit einem Grossbuchstaben und endet mit einem Punkt. Das Programm soll 20 Sätze generieren.

Die einzelnen Arrays sollen mit folgenden Wörtern gefüllt werden:

**article**: „ the“ , „ a“ , „ one“ , „ some“ ,any“ ; **noun**: „ boy“ , „ girl“ , „ dog“ , „ town“ , „ car“ ; **verb**: „ drove“ , „ jumped“ , „ ran“ , „ walked“ , „ skipped“ ; **preposition**: „ to“ , „ from“ , „ over“ , „ under“ , „ on“ .

**Lösungshinweis: Für die zufallsmässige Auswahl eines Wortes verwenden Sie Math.random.**

### 7.9 (\*) Taschenrechner

(Konvertierungsmethoden)

Schreiben Sie einen „ Mini-Taschenrechner“ , bei der der Bediener die Möglichkeit hat zwischen zwei Operationen Addition und Multiplikation zu wählen, die entsprechenden Zahlen eingibt und das Programm das Resultat auf dem Bildschirm ausgibt. Verwenden Sie hierzu die entsprechende Konvertierungsmethode **valueOf** der Klasse **String**.

### 7.10 () Papieraufgabe: Konvertierungen

Wie kann man einen String z.B. „1500“ in den entsprechenden fundamentalen Datentyp (in diesem Fall **int**) konvertieren? Welche Klassen und Operationen verwendet man dazu?

**7.11 (\*) URL Konvertierung***(StringBuffer)*

Schreiben Sie eine Klassenmethode `convertUrl2JavaPackageName`, die eine url als `String s` übergeben bekommt und diese in einen qualifizierten Java Packagenamen umwandelt und diesen als `String` zurückliefert. **Implementieren Sie diese Klassenmethode mit Hilfe eines `StringBuffer`s** und testen Sie Ihre Implementierung.

Für die Konvertierung gelten folgende Regeln:

- Falls `www` und/oder `http://` in der url auftaucht, wird `www` bzw. `http://` weggelassen
- ansonsten wird die url in umgekehrter Reihenfolge konvertiert

Beispiele für die Konvertierung:

- `"www.sun.com "` → `"com.sun"`
- `"http://www.hs-coburg.de "` → `"de.hs-coburg"`
- `"hs-coburg.de "` → `"de.hs-coburg"`
- `"//hs-coburg.de "` → `"de.hs-coburg"`

**Lösungshinweis:**

- **Es reicht aus, wenn wir die Konvertierung nur für die angegebenen Beispiele durchführen. D.h. bspw. keine Konvertierung für Unterseiten `"http://www.hs-coburg.de"` durchführen.**
- **Außerdem gehen wir von einer korrekt geschriebenen url aus. D.h. wir betrachten Fehler in der url Schreibweise nicht.**