

Übung 6.1 (Genetik)

Das Erbgut (Genom) eines Lebewesens wird in der DNA als Abfolge (Sequenz) der Basen Adenin (A), Guanin (G), Cytosin (C) und Thymin (T) gespeichert. In einem Programm lässt sich die Erbinformation als String speichern:

```
tttcaagctgcggtaatcggtagatccccggtacggtaagct
```

In der sogenannten Bioinformatik wird mit solchen Strings gearbeitet. Aufgabe: stelle fest wie oft die Sequenz `cggt` in der obigen Erbinformation vorkommt.

Übung 6.2 (Satz und Wort)

Schreibe eine Klasse die zwei Strings (`satz` und `wort`) als Instanzvariablen enthält (zu setzen über den Konstruktor).

- a) Implementiere eine Methode die ermittelt wie oft `wort` in `satz` vorkommt.

- b) ☺ Implementiere eine Methode die alle passenden Wörter aus dem Satz löscht. Der neue Satz soll der Rückgabewert sein. Achte darauf, dass sich die Instanzvariable `satz` nicht ändert.

Übung 6.3 (String – Palindrom)

Schreibe ein Programm das einen Text umgekehrt ausgibt. Weiters soll das Programm prüfen können ob es sich um ein Palindrom handelt. Ein Palindrom ist ein Text der von vorn und von hinten gelesen gleich bleibt (Bsp. Rentner). Implementiere:

- a) Methode `String reverse(String str)`, die den String umdreht und das Ergebnis zurückgibt.
- b) Methode `boolean isPalindrom(String str)`, die prüft ob es sich bei dem als Parameter übergebenen String um ein Palindrom handelt, dabei soll Groß- und Kleinschreibung ignoriert werden.

Übung 6.4 (String – Leerzeichen)

Ein fix im Programm vorgegebener String soll so in einen zweiten String kopiert werden, dass darin alle Leerzeichen gelöscht sind.

- a) Lösung mit der `charAt` Methode.

- b) ☺ Lösung ohne Verwendung von `charAt`. Verwende stattdessen `indexOf` und `substring` (benötigt weniger Schleifendurchläufe).

Übung 6.5 (Count words)

From <https://wiki.engr.illinois.edu/display/cs125/Sample+Exams>

Write a complete Java program that processes user input lines. Your program should ignore any lines until it reads the line with exactly the text "START" in uppercase with no other preceding or trailing characters. Once START has been read, count the number of subsequent lines that include the character sequence "salad" in

upper- or lower-case or any mixture (for example the line "Ho;MySaLADer!" counts as a matching line). When your program reads a line that is exactly the text "END" in uppercase with no other preceding or trailing characters, output the count of the matching salad lines and exit. Do not read any additional lines. Assume there will always be a valid "START" and a later "END" line.

Übung 6.6 (String Split)

Implementiere die Klasse `StringSplit`. Ein vorgegebener Text soll damit in Teilstrings gleicher Länge zerlegt werden. Der String und die Länge der Teile wird im Konstruktor gesetzt. Die Methode `hasMoreChunks()` liefert die Information ob noch ein Teil zum Lesen da ist. Die Methode `nextChunk()` liefert den nächsten Teil.

Anwendung:

```
StringSplit ss = new StringSplit("abcdefghij",4);
```

```
while (ss.hasMoreChunks()) {  
    System.out.println("\""+ss.nextChunk()+"\"");  
}
```

Ausgabe:

```
"abcd"  
"efgh"  
"ij"
```