

Übung 10: Strings und Arrays

Aufgabe 1: Fehlersuche

Im folgenden (ziemlich sinnlosen) Programm sind Fehler enthalten. Das Programm liest einen Namen von der Kommandozeile. Falls „Duck“ eingegeben wird, so wird der Benutzer mit „Hallo Donald“ begrüßt. Anschließend gibt das Programm den 2. und 3. Buchstaben der Eingabe als Großbuchstaben aus.

Überlegen Sie sich ohne¹ Computer, was falsch ist und korrigieren Sie das Programm! Ggfs. können sie das Programm später in IntelliJ prüfen.

```
public class StringOops {
    public static void main(String[] args) {
        Scanner console = new Scanner(System.in);
        System.out.print("Type your name: ");
        String name = console.next();
        process(name);
    }

    public static void process(string "name") {
        if (name == Duck) {
            System.out.println("Hello Donald!");
        }
        toUpperCase(name);
        name.substring(1, 3);
        System.out.println(name + " has " + name.length + " letters");
    }
}
```

Aufgabe 2: Zerlegen und Zusammensetzen von Zeichenketten

In dieser und der nächsten Aufgabe 3 werden einige statische Methoden für Zeichenketten in der Klasse `StringUtil` implementiert.

- a) Gegeben seien zwei Strings `s1` und `s2` sowie 2 Arrays `a1` und `a2`. Welche Java-Anweisung vergleicht `s1` und `s2` bzw. `a1` und `a2` auf Gleichheit? Welche Anweisung vergleicht auf Identität?

Equals

==

- b) `public static String[] string2StringArray(String s)`
Implementieren Sie diese Methode, die den String `s` in seine Einzelwörter zerlegt und die Einzelwörter dann alphabetisch sortiert in ein Array von Strings einträgt. Bsp.: Die Eingabe „bald sind ferien“ liefert das String-Array („bald“, „ferien“, „sind“).

Hinweise:

- Die Java-Klasse `Arrays` liefert eine einfache Möglichkeit ein Array von Strings zu sortieren.
<https://docs.oracle.com/javase/8/docs/api/java/util/Arrays.html>
- Informieren Sie sich über die Methode `split` der Klasse `String` und über reguläre Ausdrücke:
<https://docs.oracle.com/javase/8/docs/api/java/lang/String.html>
<https://docs.oracle.com/javase/8/docs/api/java/util/regex/Pattern.html>
- Als Trennzeichen für Einzelwörter fungieren **1 oder mehrere aufeinanderfolgende** White-spaces.

- c) `public static String stringArray2String(String[] arr)`
Fügt die Elemente eines String-Arrays zu einem langen String (mit Trennzeichen `' '`) zusammen. Verwenden Sie hierzu die Klasse `StringBuilder`.
Hinweis: Verwenden Sie nicht `Arrays.toString()` oder die Methode `join` der Klasse `String`.

¹ In der Klausur gibt es leider auch kein IntelliJ.

- d) Implementieren Sie eine JUnit4 Testklasse `StringUtilTest`, die Folgendes testet:
- „kurt fährt auto“ soll gemäß b) korrekt in das Array [„auto“, „fährt“, „kurt“] zerlegt werden.
 - „kurt fährt auto“ soll trotz der vielen Whitespaces in das gleiche Array zerlegt werden.
 - Das Array [„kurt“, „fährt“, „Auto“] ergibt beim Zusammensetzen gemäß c): „kurt fährt Auto“.
- Die Methode `assertArrayEquals()` kann nützlich sein.

Aufgabe 3: Anagramme

Der Begriff Anagramm bezeichnet eine Buchstabenfolge, die aus einer anderen Buchstabenfolge allein durch Umstellung (Permutation) der einzelnen Buchstaben gebildet wird. Beispielsweise sind „Bier“ und „Brei“ ein Anagramm. Ebenso sind „Fr. Inge C. Sonst, Rheine“ und „Schornsteinfegerin“ Anagramme. Wir betrachten im Folgenden nur Buchstaben und ignorieren Sonderzeichen, Ziffern, Leerzeichen sowie Groß- und Kleinschreibung.

- a) `public static boolean areAnagrams(String string1, String string2)`
Implementieren Sie diese Methode, die prüft ob zwei Strings Anagramme sind. Hinweise:
- Sie können mit `Character.isLetter(char c)` auf einen Buchstaben prüfen.
 - Nützliche Funktionen finden Sie ggfs. auch in der Klasse `Arrays` und `StringUtil`.
- b) Erweitern Sie die JUnit4 Testklasse `StringUtilTest` von Aufgabe 2 mit folgenden Tests:
- „Debit Card“ und „Bad Credit“ sind Anagramme.
 - „derbe Hotline“ und „Bohlen, Dieter“ sind Anagramme.
 - „Haus“ und „Haus“ sind keine Anagramme, da es sich um die gleichen Wörter handelt.
 - „joy“ und „enjoy“ sind ebenfalls keine Anagramme.