

Hausaufgabe 5

Abgabe: 03.12.2023 (23:59 Uhr)

Abgabe:

Für dieses Hausaufgabenblatt ist **eine** `StringHomework.java` Datei zu erstellen: Schreiben Sie alle Methoden aus den Aufgaben in diese Klasse. Schreiben Sie alle Ihre Testfälle in die `main` Methode der Klasse, so dass beim Ausführen alle Programmteile der Teilaufgaben hintereinander durchlaufen. Das Programm muss für sich lauffähig sein und arbeitet mit Konsolenausgaben. Wenn Sie mögen, nehmen Sie den Quellcode aus der SU oder dem Live-Coding als Basis.

Tipp: Schauen Sie in der Dokumentation von `String` nach, welche Methoden Ihnen helfen könnten:

<https://elearning.hs-fulda.de/ai/mod/lti/view.php?id=41088> bzw.

<https://docs.oracle.com/en/java/javase/16/docs/api/java.base/java/lang/String.html#method.summary>

Aufgabe 0: Palindrome (Lernziel: Nutzung von Methoden der Klasse `String`, Umgang mit Strings)

Schreiben Sie eine Methode `public static boolean isPalindrome(String s)`, die berechnet, ob das übergebene Argument ein Palindrom ist oder nicht. Ein Palindrom ist ein `String`, der vorwärts wie rückwärts gelesen gleich ist. Unterschiede der Groß-/Kleinschreibung sollen dabei ignoriert werden, so dass „Abba“ ein Palindrom ist, ebenso wie „Legovogel“. Testen Sie mit Aufrufen aus der `main` Methode insbesondere auch Fälle, wie `null`, den leeren `String`, un/gerade Anzahl Zeichen, sowie Strings der Länge 1.

Tipp: Mit der `String`-Methode `toLowerCase()`¹ erhalten Sie einen neuen `String` zurück, in welchem alle Zeichen in Kleinbuchstaben umgewandelt wurden.

Aufgabe 1: Anagramme (Lernziel: String-Manipulationen, Besonderheiten im Umgang mit String-Methoden wie `replace`, `indexOf`, `charAt` erfahren)

Ein Anagramm einer Zeichenkette ist eine Zeichenkette mit exakt gleich vielen, identischen Zeichen in gegebenenfalls anderer Reihenfolge. Beispiel: *betrug* und *geburt* sind Anagramme voneinander. Außerdem sollen Leerzeichen ignoriert werden, so dass auch *tom marvolo riddle* und *i am lord voldemort* Anagramme voneinander sind. Klein-/Großschreibung soll dabei diesmal berücksichtigt werden.

Schreiben Sie eine Methode `public static boolean checkForAnagram(String s1, String s2)`, die prüft, ob die beiden Argumente Anagramme voneinander sind. Testen Sie in der `main`-Methode insbesondere auch Randfälle (wie leere Strings). Gehen Sie dabei wie folgt vor (und nicht anders):

- Entfernen Sie zunächst alle Leerzeichen.
- Falls `s1` und `s2` dieselbe Länge haben, suchen und entfernen Sie nacheinander jedes Zeichen von `s1` aus `s2`
- *[Nun fehlt nur noch die Überlegung, welcher Endzustand eintritt, falls `s1` und `s2` Anagramme sind.]*

Knobelaufgabe: Plus Operator mit Strings

Die Deklaration `String text = 'a' + 'b';` ergibt einen Compilerfehler.

Die Deklaration `String text2 = "" + 'a' + 'b';` hingegen funktioniert.

Warum?

¹ [https://docs.oracle.com/en/java/javase/16/docs/api/java.base/java/lang/String.html#toLowerCase\(\)](https://docs.oracle.com/en/java/javase/16/docs/api/java.base/java/lang/String.html#toLowerCase())