

SEW I - Übung: Anagramm-Prüfer

Ziel

Implementierung eines Programms zur Überprüfung von Anagrammen unter Zuhilfenahme von Methoden.

Lernziele

- Arbeiten mit Methoden.
- Implementieren von Sortier-Algorithmen.
- Arbeiten mit *Strings* und *chars*.

Abgaberrichtlinien

- Ihre implementierte Lösung als `.java`-Datei. **Vergessen Sie dabei nicht auf Kommentare und Kommentarkopf!**
- Achten Sie auf **saubere Variablenbenennung** und **Nutzung von Konstanten** wenn sinnvoll!

Aufgabe

Hintergrund

Als Anagramm wird eine Buchstabenfolge bezeichnet, die allein durch Umstellung von Buchstaben einer anderen Buchstabenfolge gebildet wird. Beispiele:

- *LAMPE* \iff *AMPEL*
- *ARBEITSAMT* \iff *BISAMRATTE*
- *THE EYES* \iff *THEY SEE*

Implementierung

Implementieren Sie ein Programm zur Überprüfung von Anagrammen. Programmieren (und testen!) Sie dafür die folgenden Methoden und orientieren Sie sich an den Ausgabebeispielen unterhalb:

- `readSentence` fordert den/die BenutzerIn auf einen Satz einzugeben, liest diesen ein und gibt ihn als `String` zurück. Die Eingabe soll bei leeren Sätzen mit einer Fehlermeldung wiederholt werden.
- `convertSentenceToSymbols` wandelt einen übergebenen `String` in ein `Array` aus `chars` um und gibt dieses zurück.
- `printSymbols` gibt ein übergebenes `char-Array` mit Beistrichen getrennt aus. Nach dem letzten Zeichen soll kein Beistrich mehr ausgegeben werden!
- `sortSymbols` sortiert ein übergebenes `char-Array` mit einem selbst implementierten, frei wählbaren Sortier-Algorithmus. Sie brauchen hier keinen Rückgabewert, da direkt auf dem übergebenen `Array` gearbeitet wird.

- `areSymbolsEqual` erhält zwei `char`-Arrays und vergleicht diese auf Gleichheit (gleiche Länge und gleiche Zeichen an den gleichen Positionen). Das Ergebnis der Überprüfung wird als `boolean` zurückgegeben.

Beispiele

Beispiel 1

```
=====
ANAGRAM CHECKER
=====
Please enter a sentence: leonding
First sentence is "leonding".
Please enter a sentence:
Sentence must not be empty!
Please enter a sentence: neingold
Second sentence is "neingold".
=====
Symbols in the first sentence: l, e, o, n, d, i, n, g
Symbols in the second sentence: n, e, i, n, g, o, l, d
=====
Sorted symbols in the first sentence: d, e, g, i, l, n, n, o
Sorted symbols in the second sentence: d, e, g, i, l, n, n, o
=====
The sentences are anagrams: true
```

Beispiel 2

```
=====
ANAGRAM CHECKER
=====
Please enter a sentence: leo
First sentence is "leo".
Please enter a sentence: leoo
Second sentence is "leoo".
=====
Symbols in the first sentence: l, e, o
Symbols in the second sentence: l, e, o, o
=====
Sorted symbols in the first sentence: e, l, o
Sorted symbols in the second sentence: e, l, o, o
=====
The sentences are anagrams: false
```

Beispiel 3

```
=====
```

ANAGRAM CHECKER

=====

Please enter a sentence: linz

First sentence is "linz".

Please enter a sentence: lodz

Second sentence is "lodz".

=====

Symbols in the first sentence: l, i, n, z

Symbols in the second sentence: l, o, d, z

=====

Sorted symbols in the first sentence: i, l, n, z

Sorted symbols in the second sentence: d, l, o, z

=====

The sentences are anagrams: false