
Programmieren in Java

<http://proglang.informatik.uni-freiburg.de/teaching/java/2017/>

subsequence*Teilfolgenordnung auf Strings*

Woche 05 Aufgabe 4/4

Herausgabe: 2017-05-22

Abgabe: 2017-06-02

Achtung: beachten Sie unbedingt die allgemeinen Hinweise zur Abgabe auf der Homepage.

Project **subsequence**Package **subsequence**

Klassen

Main
<code>public static PartialOrdering subsequenceCompare(String s1, String s2)</code>

Achtung: Von jetzt an ist es notwendig, dass Sie ihre Methoden sinnvoll mit JUnit testen, um die Punkte für Codequalität zu bekommen.

Eine Teilfolge eines Strings ist ein String, der entsteht wenn Buchstaben des ursprünglichen Strings weggelassen werden.

<https://de.wikipedia.org/wiki/Teilfolge>

Strings können durch die Teilfolge-Eigenschaft partiell geordnet werden. Die Funktion `subsequenceCompare` berechnet wie die zwei Eingabestrings `s1` und `s2` bezüglich der Teilfolge-Eigenschaft geordnet sind. Das Ergebnis von `subsequenceCompare` ist vom `enum`-Typ `PartialOrdering`:

```
1 public enum PartialOrdering {  
2     LESS,  
3     EQUAL,  
4     GREATER,  
5     INCOMPARABLE  
6 }
```

Ein `enum`-Typ definiert eine abgeschlossene, endliche Menge von Konstanten, die im Programm eine besondere Bedeutung haben. Im Fall von `PartialOrdering` gibt es die folgenden vier Konstanten. Sie sollen angeben, wie zwei Strings zueinander geordnet sind.

- `PartialOrdering.LESS`: `s1` ist Teilfolge von `s2`, aber nicht gleich `s2`
- `PartialOrdering.EQUAL`: `s1` ist gleich `s2` (und somit auch Teilfolge von `s2`).

- `PartialOrdering.GREATER`: `s2` ist Teilfolge von `s1`, aber nicht gleich `s1`.
- `PartialOrdering.INCOMPARABLE`: `s1` ist weder Teilfolge von `s2`, noch ist `s2` Teilfolge von `s1`

Die Funktion sollte auch für relativ große Strings (z.B. Länge 10000) gut funktionieren.

Beispielaufruf 21: `Main.subsequenceCompare("Hello World", "eloWrlD")` ergibt `GREATER` (als `PartialOrdering`)

Beispielaufruf 22: `Main.subsequenceCompare("Hello World", "World Hello")` ergibt `INCOMPARABLE` (als `PartialOrdering`)

Hinweise: Weitere Informationen zu `enum`-Typen finden Sie hier:

<http://docs.oracle.com/javase/tutorial/java/java00/enum.html>.