

Eine der 3 Programmieraufgaben muss individuell erfolgreich bearbeitet werden.

Aufgabe 1: Inversionszahl

Es soll für ein Feld die Inversionszahl mit Insertionsort berechnet werden. Das Feld enthält Zeichenketten als Elemente und wir gehen von einem besonderen Sortierkriterium (siehe unten) aus, das ebenfalls implementiert werden muss.

Das Programmierprojekt ist bereits entsprechend vorbereitet, sodass Sie nur in eine Datei als Lösung programmieren müssen. Dazu finden Sie in einem Zip-Archiv in OPAL die folgenden Dateien:

- Verzeichnis `adsaufgabe1`
Dies entspricht dem Paketnamen in Java und muss so unverändert übernommen werden.
- Datei `IInversionszahl.java`
Dieses Interface beschreibt die Methoden, die Ihre selbst programmierte Klasse implementieren muss. Hier ist auch die Funktionsweise genau als Kommentar beschrieben. Diese Datei darf **nicht** verändert werden.
- Datei `InversionszahlHTWKLOGIN.java`
Von dieser Datei legen Sie sich eine Kopie an, wobei Sie im Datei- und Klassennamen `HTWKLOGIN` durch Ihren Namen ersetzen. In dieser Datei programmieren Sie dann die gewünschte Lösung. Achten Sie bitte darauf, dass die Klasse das `implements IInversionszahl` enthält und der Klassenname zum Dateinamen angepasst wird. Verzichten Sie zum Zeitpunkt der Abgabe auch in Ihrer Klasse auf Konsolenausgaben per `System.out.print*`. Konkret muss die Klasse die folgenden Methoden bzw. Konstruktoren besitzen:
 - Konstruktor `InversionszahlHTWKLOGIN()`
 - `int berechne(String[] feld)`: Der Algorithmus setzt Insertionsort um und sortiert die Zeichenketten. Dabei wird gezählt, wie oft ein Element im Insertionsortalgorithmus vor ein anderes Element geschoben wird. Das Ergebnis dieser Zählung entspricht dem Rückgabewert. Als Seiteneffekt ist danach das Feld sortiert, was wir hinnehmen, aber nicht überprüfen.
- Datei `Testtreiber.java`
Diese Klasse enthält das Hauptprogramm mit einem beispielhaften Testtreiber. Mit zwei Testfällen werden die beiden zu implementierenden Methoden geprüft. Haben Sie Ihre Implementation mit Ihrem individuellen Namen abgespeichert, müssen Sie den Aufruf in Zeile 4 anpassen. Hier dürfen Sie weitere Testfälle ergänzen.

Bitte lesen Sie die Kommentare in den Dateien aufmerksam, bevor Sie mit der Bearbeitung beginnen!

Wenn Sie davon überzeugt sind, dass Ihr Programm das richtige Ergebnis liefert, laden Sie Ihre Version der Datei `InversionszahlHTWKLOGIN.java` ungepackt in OPAL unter *Prog.Aufg. 1* hoch.

Ich werde die Datei mit meinen erweiterten Testfällen übersetzen und entsprechend auswerten. Ferner prüfe ich auf Quelltext-Kopien (Plagiate). Wenn die Testfälle richtig gelöst werden und keine Kopie vorliegt, haben Sie die erste Programmieraufgabe erfolgreich bestanden. In der Regel erlaube ich einen fehlgeschlagenen Testfall.

Das Sortierkriterium für die Zeichenketten ist dabei wie folgt. Die Zeichenkette $s = s_0s_1s_2 \cdots s_{k-1}$ sei **kleiner** als Zeichenkette $t = t_0t_1t_2 \cdots t_{\ell-1}$, d.h. $s < t$, genau dann wenn:

1. Das erste Vorkommen von a in s am Index i vor dem ersten Vorkommen von a in t am Index j passiert, also $i < j$ gilt. Falls a in einem der Strings nicht vorkommt, sei der entsprechende Index ∞ . — oder —
-

2. Falls in (1) $i = j$ gilt und das letzte Vorkommen von b in s an der Stelle i' dichter am Ende von s steht als das letzte Vorkommen von b in t and der Stelle j' , als $k - i' < \ell - j'$. Hier entspreche das Nichtvorkommen von b dem Index $-\infty$.

Die Zeichenketten s und t seien gleich, d.h. $s = t$, genau dann, wenn in (1) $i = j$ und in (2) $k - i' = \ell - j'$ gilt. In allen anderen Fällen gilt $t < s$.

Beispiele:

- `ijkaubbbbbbbbbbb < bbbba`
- `uiaooooobo < qbao`
- `udhgbaii = poikjabxxx`
- Das Feld `[aaa, bba, aba, aab]` hat die Inversionszahl 5 und im sortierten Zustand die Belegung `[aab, aba, aaa, bba]`.

Hinweise zu Strings in Java:

- Zeichenketten sind in Java unveränderliche Objekte.
- Ihre Länge kann mit der Methode `length()` erfragt werden.
- Die Zeichen im String beginnen beim Index 0. Das Zeichen an der i -ten Stelle kann mit der Methode `charAt(i)` erfragt werden.

Hinweises zu Feldern in Java:

- Die Deklaration von Feldern kann dem Testtreiber entnommen werden.
- Die Länge von Feldern kann über das Attribut `length` (ohne die runden Klammern der Strings) erfragt werden – für ein Feld `a`: `a.length`
- Felder beginnen ebenfalls mit dem Index 0. Auf das i -te Element eines Feldes `a` kann lesend und schreibend mit eckigen Klammern zugegriffen werden `a[i]`.

Hinweise zur Verwendung von Insertionsort:

- Da ich an keiner Stelle überprüfe, inwieweit wirklich Insertionsort implementiert wird, könnten Sie theoretisch die Inversionszahl auch anders ermitteln.
- Allerdings werde ich auch Performance-Tests laufen lassen, wodurch wesentlich ineffizientere Implementationen als eine Referenzimplementation den jeweiligen Test nicht bestehen werden.
- Die Tests gehen davon aus, dass die Strings in jedem Vergleich neu analysiert werden – diesbezüglich ist es also nicht notwendig, zu optimieren und Zwischenberechnungen zu speichern, um sie später wieder zu benutzen.

Hinweise zur Abgabe:

- Ich benötige ausschließlich die Datei, die mit `Inversionszahl` beginnt und mit `.java` aufhört. (Eine `.class`-Datei kann ich nicht bewerten und weitere Testfälle o.ä. benötige ich nicht von Ihnen. Bitte laden Sie die Datei auch direkt und nicht gezippt oder anderweitig komprimiert hoch.)
-

- Verzichten Sie auf jegliche Konsolen-Ausgabe in Ihrer Klasse - meine Testfälle prüfen die Rückgabewerte der Funktionen und die Änderungen im Feld.
- Benennen Sie Ihre abzugebende Datei korrekt mit Ihrem HTWK-Login.
- Vergewissern Sie sich, dass Ihre Abgabe übersetzbar ist – dies kann ggf. durch ein nachträgliches Umbenennen der Datei beeinträchtigt werden.

Jedes Problem, das sich aus der Missachtung einer der Hinweispunkte ergibt, wird als ein fehlgeschlagener Testfall gewertet. (Ich möchte nicht > 50 Java-Projekte erst mühsam übersetzbar machen – dafür fehlt mir die Zeit. . .)

Abgabedatum: **Montag, 6. Mai 2024, 8:00 Uhr**
