

Aufgabenblatt 06

Bearbeitungsende: 20.11.2022

Aufgabe 1 [Programmierung]

Schreiben Sie eine Klasse `Zeichenkette` mit folgenden Klassenmethoden:

- (a) `anzahlBuchstaben` nimmt eine Zeichenkette als Argument an und gibt zurück, wieviele Buchstaben diese enthält.
Z.B. ergibt ein Aufruf mit dem Argument `"Hallo, Welt!"` den Zahlwert 9.
- (b) `istZiffernfolge` nimmt eine Zeichenkette als Argument an und gibt einen Wahrheitswert zurück, ob sie ausschließlich Ziffern enthält. Es sind keine White Spaces zugelassen, auch nicht am Anfang oder Ende. Eine leere Zeichenkette ist eine gültige (eben leere) Ziffernfolge.
Z.B. ergibt ein Aufruf mit dem Argument `"Hallo, Welt!"` den Wert `false`, ein Aufruf mit dem Argument `"451"` den Wert `true`.
- (c) `istName` nimmt eine Zeichenkette als Argument an und gibt einen Wahrheitswert zurück, ob sie ausschließlich Buchstaben oder Bindestriche ('-') enthält. Ein Bindestrich darf aber nicht als erstes oder letztes Zeichen auftreten.
Es sind keine White Spaces zugelassen, auch nicht am Anfang oder Ende. Eine leere Zeichenkette ist ein gültiger (eben leerer) Name.
Z.B. ergibt ein Aufruf mit dem Argument `"Lisa-"` oder `"Johann Wolfgang"` den Wert `false` (letzteres sind zwei Namen). Ein Aufruf mit dem Argument `"Brad-Arnold"` oder `"ey"` ergibt den Wert `true`.
- (d) `vokaleGross` nimmt eine Zeichenkette als Argument an und gibt sie als neue Zeichenkette zurück, in der alle Vokale (a, e, i, o, u) groß geschrieben sind, alle anderen Buchstaben klein.
Z.B. ergibt ein Aufruf mit dem Argument `"Hallo, Welt!"` die Zeichenkette `"hAllo, wElT!"`.

Hinweis: Verwenden Sie die Methoden der Klassen `String` und `Character`, die in der Vorlesung vorgestellt wurden - andere brauchen Sie nicht.

Aufgabe 2 [Programmierung]

Schreiben Sie eine Klasse `Zeichenkette` mit folgenden Klassenmethoden:

- (a) `rueckwaerts` nimmt eine Zeichenkette als Argument an und gibt diese in umgekehrter Reihenfolge als neue Zeichenkette zurück.
Z.B. ergibt ein Aufruf mit dem Argument `"Hallo, Welt!"` die Zeichenkette `"!tleW ,ollaH"`.

- (b) **einfuegung** nimmt als Argumente eine Zeichenkette **s**, eine ganzzahlige Position **p** in **s** sowie eine Zeichenkette **t** an. Der Wert von **p** kann auch die Länge von **s** sein und so deren Ende beschreiben.

Die Methode fügt **t** vor der Position **p** in **s** ein und gibt die so entstandene Zeichenkette als Ergebnis zurück.

Z.B. ergibt ein Aufruf mit den Argumenten "Brot" (als **s**), 2 (als **p**) und "Kaese" (als **t**) das Ergebnis "BrKaeseot". Hat **p** den Wert 4, ist das Ergebnis "BrotKaese". Für 0 ergibt sich "KaeseBrot".

Hinweis: Verwenden Sie die Methoden der Klassen **String** und **Character**, die in der Vorlesung vorgestellt wurden – andere brauchen Sie nicht.

Aufgabe 3 [Programmierung – nicht bewertet]

Schreiben Sie eine Klasse **ZeichenketteTest** mit einer Klassenmethode **test**, die keine Argumente annimmt, und die einen Wahrheitswert zurückgibt. Sie testet gegebene Implementierungen der Methode **verkuerze** der Klasse **Zeichenkette** aus der folgenden Aufgabe auf diesem Blatt. Die Testmethode soll genau dann **false** zurückgeben, wenn für eine Implementierung ein Fehler gefunden wird.

Hinweis: Halten Sie sich an das Beispiel für Unit-Tests aus der Vorlesung. Die Klasse sollte aber keine **main**-Methode enthalten. Sie finden in LEA zu dieser Aufgabe eine entsprechende Vorlage, die sich einfach um weitere Testfälle zu ergänzen lässt. Überlegen Sie, welche unterschiedlichen Fälle von Aufrufen vorkommen können. Sehen Sie jeweils einen repräsentativen Aufruf vor und prüfen Sie, ob das Ergebnis der erwartete Wert ist.

Testen Sie die Klasse mit fehlerhaften Implementierungen von **Zeichenkette**. Werden diese erkannt?

Laden Sie (nur) die Datei **ZeichenketteTest.java** in den Praktomat. Dort wird sie verwendet, um verschiedene richtige und falsche Implementierungen von **Zeichenkette** zu testen. Sie sollte alle Fälle korrekt identifizieren.

Hinweis: Wenn der Praktomat einen Fehler in Ihrem Test moniert, bedeutet das:

- Wird von Ihrem Test **false** geliefert, aber **true** erwartet, wurde eine korrekte Lösung als falsch beurteilt. In Ihren Tests ist also wenigstens ein Fall fehlerhaft umgesetzt.
- Wird von Ihrem Test **true** geliefert, aber **false** erwartet, wurde eine falsche Lösung als korrekt beurteilt. Sie haben also wenigstens einen Fall nicht abgedeckt und müssen überlegen, welcher das ist und welche Testfälle entsprechend hinzuzufügen sind.

Aufgabe 4 [Programmierung]

Schreiben Sie eine Klasse **Zeichenkette** mit einer Klassenmethode **verkuerze**. Die Methode soll eine Zeichenkette **s** und eine nichtnegative (was nicht zu prüfen ist) ganze Zahl **n** annehmen.

Die Methode soll eine Zeichenkette zurückgeben, die sich aus folgenden Teilen zusammensetzt:

- die ersten **n** Zeichen von **s**
- die Zeichenfolge "[...]" (symbolisiert „Auslassung“)
- die letzten **n** Zeichen von **s**

(Falls `n` größer ist als die Länge der Zeichenkette, sind nur so viele Zeichen zu berücksichtigen wie vorhanden.)

Wenn aber die Zeichenkette (so kurz ist, dass sie) sich durch die Änderung nicht verkürzen würde, soll das Resultat die unveränderte Zeichenkette sein.

Schreiben Sie ein Testprogramm `ZeichenketteTest`, laden Sie dieses aber nicht in den Praktomat.

Beispiele:

- `verkuerze("Abrakadabra",2)` liefert `"Ab[...]ra"` (in genau dieser Form, mit drei Punkten).
- `verkuerze("Abrakadabra",3)` liefert `"Abrakadabra"`.

Hinweis: Verwenden Sie geeignete Methoden der Klasse `String`.

Der von Ihnen zu schreibende Code sollte i.W. aus (wenigen!) Aufrufen solcher Methoden und aus Fallunterscheidungen bestehen.

Aufgabe 5 [Programmierung – nicht einzureichen]

Statten Sie Ihre Lösung zu B05A1 mit Kommentaren im `Javadoc`-Format aus. Generieren Sie daraus ein Benutzerhandbuch mit der entsprechenden API-Spezifikation.

Hinweis: Sie müssen für diese Aufgabe die Aufgabe nicht korrekt gelöst haben; es reicht sonst auch aus, wenn Sie die Methoden als „Dummies“ implementieren, die Unsinnswerte produzieren.

Lösungen zu mit [Programmierung] markierten Aufgaben sind im **Praktomat** einzureichen.

Lösungen zu mit [Programmierung – nicht bewertet] markierten Aufgaben können ebenfalls im **Praktomat** eingereicht werden, werden jedoch nicht bewertet.

Allgemeine **Fragen** zu den Aufgaben können Sie im **LEA-Forum** „Übungsaufgaben“ stellen.

Hilfe bei der Lösung der Aufgaben erhalten Sie in den **Übungen** und in der **Studierwerkstatt** .