

## Übungsblatt 7: Objektorientierte Programmierung

Ausgabe: 05.12.2019

Abgabe: 12.12.2019

**Aufgabe 1 (5+6 Punkte)** Verwenden Sie den folgenden Freigabecode für *FileManagement* und *Parser*: 5d0kgh9c3g69-1fef

Ihre Aufgabe ist es zwei Klassen *FileManagement* und *Parser* zu erstellen. Für den ersten Teil ist es explizit notwendig, dass Sie selbst recherchieren, wie Sie die Aufgabe lösen können! Beachten Sie auch, dass Sie selbst eigene Testfälle erstellen dürfen und allein die als Hilfestellung vorgegebenen Testfälle keine Garantie für ein korrektes Programm sein können.

**FileManagement** Diese Klasse soll die folgenden beiden Methoden beinhalten:

**readFromFile** erhält als Parameter einen String, der einen Dateinamen beinhaltet, und gibt den Inhalt der Datei als Zeichenkette (String) zurück.

**writeToFile** erhält als Parameter zwei Strings. Der erste String beinhaltet den Text, welcher in die Datei geschrieben werden soll. Der Dateiname ist dabei durch den zweiten String gegeben.

**Parser** Diese Klasse soll Brüche als Strings dargestellt in *double*-Werte und umgekehrt umwandeln können. Implementieren Sie dazu die folgenden beiden Methoden:

**parseToDouble** erhält als Parameter einen String, welche die Brüche durch jeweils ; getrennt beinhaltet. Ein Bruch ist immer durch einen ganzzahligen Zähler, gefolgt von einem Schrägstrich und einem ganzzahligen Nenner dargestellt. ( $-1/2$ ;  $3/4$ ;  $5/-2$  würde den Brüchen  $-\frac{1}{2}$ ,  $\frac{3}{4}$  und  $-\frac{5}{2}$  und folglich den Gleitkommazahlen  $-0.5$ ,  $0.75$  und  $-2.5$  entsprechen). Der Rückgabewert ist ein Feld von *double*-Werten.

**parseToString** Erhält ein Feld von *double*-Werten und soll diese in gekürzte Brüche umwandeln und als String im obigen Format darstellen ( $-0.5$ ,  $0.75$  und  $-2.5$  würden somit als  $-1/2$ ;  $3/4$ ;  $-5/2$  dargestellt werden). Um Eindeutigkeit zu gewährleisten, soll bei negativen Brüchen das Minuszeichen immer beim Zähler sein. Der Rückgabewert ist der String.

**Aufgabe 2 (6 Punkte)** Verwenden Sie den folgenden Freigabecode für *Linearcode* für 5 Ziffern und 1 Prüfwert: 5d0kgha2ec0n-5710

Schreiben Sie ein Programm, welches als Kommandozeilenparameter einen String übergeben bekommt, welcher einen Zielcode<sup>1</sup> darstellt. Dieser besteht nur aus Leerzeichen ( ) und senkrechten Strichen (|). Jeweils 5 Zeichen gehören zu einem Block und jeder Block codiert eine Ziffer. Somit gibt es insgesamt 6 Blöcke, wobei der letzte Block die Prüfwert beinhaltet. Die Blöcke selbst sind durch einen senkrechten Strich voneinander getrennt und nach dem letzten Block (der Prüfwert) kommt ebenfalls ein senkrechter Strich. Ihre Aufgabe ist es die Ziffern zu ermitteln und anhand der Prüfwert diese auf Korrektheit zu überprüfen. Wenn die Ziffern nicht korrekt sind, soll *Falscher Code* ausgegeben werden, ansonsten sollen die fünf Ziffern (ohne Prüfwert) in umgekehrter Reihenfolge hintereinander weg (d. h. in einer Zeile) ausgegeben werden.

<sup>1</sup><https://de.wikipedia.org/wiki/Zielcode>