



Übungsblatt 4

Programmierung und Softwareentwicklung (WS 17/18)

Abgabe: Freitag, 01.12.2017, 23:55 Uhr — Besprechung: ab Montag, 04.12.2017

Bitte lösen Sie die Übungsaufgaben in **Gruppen von 3 Studenten** und wählen EINEN Studenten aus, welcher die Lösung in ILIAS als **Gruppenabgabe** (unter Angabe aller Gruppenmitglieder) einstellt. Bitte erstellen Sie dazu einen **Header**, welcher die Namen der Studenten, die Matrikelnummern und die E-Mail-Adressen enthält.

Die Aufgaben, bei denen Quellcode abzugeben ist, sind mit Impl gekennzeichnet. Bitte beachten Sie die Hinweise zu den Implementierungsaufgaben, die in ILIAS verfügbar sind.¹ Achten Sie besonders darauf, dass Sie zu jeder Klasse und Methode JavaDoc-Kommentare erstellen.

Dieses Übungsblatt beinhaltet 4 Aufgaben mit einer Gesamtzahl von 40 Punkten.

Aufgabe 1 Impl Eingaben und Ausnahmen [Punkte: 7]

Laden Sie sich das im ILIAS hochgeladene Java-Projekt herunter und vervollständigen Sie die Klasse `IOConsole`.

- (a) (3 Punkte) In der Klasse `IOConsole` befinden sich verschiedene unvollständige Methoden zur Berechnung des Flächeninhalts (A) und des Umfangs (U) für die geometrischen Grundformen Kreis, rechtwinkliges Dreieck und Rechteck. Vervollständigen Sie die Methoden unter Berücksichtigung folgender Formeln:

- Kreis mit Radius r : $A_k = r^2 * \pi$, $U_k = 2 * r * \pi$
- Rechteck mit Kantenlängen a und b : $A_r = a * b$, $U_r = 2 * a + 2 * b$
- Rechtwinkliges Dreieck mit den Kantenlängen der Katheten a und b : $A_d = a * \frac{b}{2}$, $U_d = a + b + \sqrt{a^2 + b^2}$

Wird beim Aufruf einer der Methoden ein ungültiger Parameter übergeben (z.B. ein negativer Radius), so soll eine `IllegalArgumentException` geworfen werden.

- (b) (4 Punkte) Implementieren Sie die Methode `printMenu`. Die Methode soll ein einfaches textbasiertes Menü auf der Eingabekonsolle realisieren, mit welchem der Benutzer die zuvor implementierten Methoden verwenden kann. Zu Beginn soll das Menü alle Grundform auflisten (Kreis, rechtwinkliges Dreieck, Rechteck). Der Nutzer soll dann mit Hilfe der Konsoleneingabe auswählen können, für welche geometrische Grundform er die Fläche und den Radius berechnen möchte. Nachdem der Benutzer eine Grundform ausgewählt hat, soll er nach den für die Berechnung notwendigen Werten gefragt werden, die er dann ebenfalls per Konsoleneingabe eingeben kann. Anschließend sollen Fläche und Radius berechnet und ausgegeben werden.

Aufgabe 2 Impl Stringmanipulationen [Punkte: 12]

Vervollständigen Sie die Klasse `StringFunctions`, welche die folgenden Methoden enthält:

- (a) (5 Punkte) Implementieren Sie die Methode `numberOfDiphthongs`, welche einen `String` übergeben bekommt und anschließend die Anzahl der Diphthongs des übergebenen `Strings` ermitteln und zurückgeben soll.

Unter einem Diphthong verstehen wir in dieser Aufgabe das Aufeinandertreffen von zwei Vokalen bzw. von Umlaut und Vokal in den folgenden Kombinationen: „ai“, „au“, „äu“, „ei“, „eu“ und „ui“. So besitzt das Wort „ietei“ beispielsweise nur einen Diphthong, während das Wort „auai“ zwei Diphthongs besitzt.

¹https://ilias3.uni-stuttgart.de/goto_Uni_Stuttgart_fold_1318001.html

- (b) (5 Punkte) Implementieren Sie die Methode `isPalindrome`, welche einen `String` übergeben bekommt und anschließend überprüft, ob der `String` ein Palindrom ist, oder nicht. Palindrome sind Wörter, die sowohl vorwärts, als auch rückwärts gelesen denselben Ausdruck ergeben. Bekannte Palindrome sind beispielsweise „Lagerregal“, „Rentner“ oder „Dad“.
- (c) (2 Punkte) Die `main`-Methode soll nun die Methoden aus den ersten beiden Teilaufgaben verwenden und über ein einfaches textbasiertes Menü benutzbar machen. Hierbei soll ein kleines Spiel implementiert werden. Zuerst gibt der Spieler einen `String` auf der Konsole ein. Wenn der `String` ein Palindrom ist, dann darf der Spieler weiter spielen und einen weiteren `String` auf der Konsole eingeben. Enthält dieser `String` mehr als 3 Diphthongs, so hat der Spieler gewonnen, ansonsten hat er verloren.

Aufgabe 3 Impl Java Collections I [Punkte: 11]

Im Folgenden sollen Sie einen Post-Simulator realisieren. Dabei spielen drei Komponenten eine wichtige Rolle: Briefe (`letter`), Postsäcke (`postbag`) und Briefkästen (`postbox`). Ziel dieser Aufgabe ist es, die Basisfunktionalität für den Simulator zu implementieren. Vervollständigen Sie hierfür die Klassen `Letter`, `PostBag` und `PostBox`.

- (a) (2 Punkte) Implementieren Sie die Klasse `Letter`, sodass diese die Funktionalitäten der im Quellcode beschriebenen Kommentare besitzt.
- (b) (2 Punkte) Implementieren Sie die Klasse `PostBag`, sodass diese die Funktionalitäten der im Quellcode beschriebenen Kommentare besitzt.
- (c) (3 Punkte) Fügen Sie der Klasse `PostBox` eine Datenstruktur hinzu, welche verwendet werden kann, um alle Briefe zu speichern. Die Datenstruktur soll im Konstruktor der Klasse `PostBox` initialisiert werden. Implementieren Sie außerdem die Methode `addLetters`, welche beliebig viele Briefe als Parameter übergeben bekommt und alle Briefe der Datenstruktur (und somit dem Briefkasten) hinzufügt.
- (d) (2 Punkte) Implementieren Sie die Methode `printAll` der Klasse `PostBox`, welche sowohl die Senderinformationen, als auch die Empfängerinformationen aller Briefe, die sich im Briefkasten befinden, auf der Konsole ausgibt.
- (e) (2 Punkte) Implementieren Sie die Methode `emptyToPostBag` der Klasse `PostBox`, welche alle Briefe aus dem Briefkasten dem übergebenen Postsack hinzufügt. Sobald alle Briefe dem Postsack hinzugefügt sind, soll der Briefkasten geleert werden.

Aufgabe 4 Impl Dateien und die Serialisierung von Objekten [Punkte: 10]

Erweitern Sie die Klasse `PostBox` und implementieren Sie darin die Methoden `saveLettersToFile` und `loadLettersFromFile`. Beachten Sie, dass Sie eventuell auch die Klasse `Letter` erweitern müssen, um deren Objekte serialisieren zu können. Nutzen Sie die Methode `mainMenu` um die Funktionalitäten aus Aufgabe 3 und 4 zu testen.

- (a) (5 Punkte) Die Methode `saveLettersToFile` soll alle im Briefkasten enthaltenen Briefe in einer Binärdatei speichern.
- (b) (5 Punkte) Die Methode `loadLettersFromFile` soll eine Binärdatei einlesen und alle darin gespeicherten Briefe dem Briefkasten hinzufügen.