



## Übungsblatt 5

Programmierung und Softwareentwicklung (WS 17/18)

Abgabe: Freitag, 08.12.2017, 23:55 Uhr — Besprechung: ab Montag, 11.12.2017

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.<sup>1</sup> 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 Stringverarbeitung [Punkte: 10]

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

- (a) (5 Punkte) Implementieren Sie die Methode `readTextAndCount`, welche einen Text aus einer Textdatei einliest und für jedes Wort auf der Konsole ausgibt, wie oft das Wort im eingelesenen Text vorkommt. Nutzen Sie eine `HashMap` zum Speichern der Worte und der jeweiligen Anzahl.
- (b) (5 Punkte) Ebenso wie Informatiker erfinden auch Kinder gerne spezielle „Sprachen“, die sich lustig anhören und deren Verständnis nur Eingeweihten zugänglich ist. Das im englischsprachigen Raum bekannte „Pig-Latin“ ähnelt dabei der hierzulande bekannten „ $\pi$ -Sprache“, der wir diese Aufgabe widmen wollen. Die „ $\pi$ -Sprache“ funktioniert so, dass an alle zusammenhängenden Folgen von Vokalen innerhalb eines Wortes ein „pi“ angehängt wird. Aus dem Satz „Meine Mutter isst gerne Fisch“ wird so beispielsweise „Meipinepi Mupittepir ipisst gepirnepi Fipisch“. Nach einiger Übung lernt man problemlos, die „ $\pi$ -Sprache“ fließend zu sprechen und zu verstehen. Implementieren Sie die folgenden beiden Methoden um die „ $\pi$ -Sprache“ verwenden zu können:
  - Die Methode `isVowel` soll genau dann den Wert `true` zurückgeben, wenn das übergebene Zeichen `c`, vom Datentyp `char`, ein Vokal ist, ansonsten soll `false` zurückgegeben werden. Achtung! In dieser Aufgabe werden Umlaute wie Vokale behandelt. Die Liste aller Vokale ist somit: „a“, „e“, „i“, „o“, „u“, „ä“, „ö“, „ü“, „A“, „E“, „I“, „O“, „U“, „Ä“, „Ö“ und „Ü“.
  - Die Methode `toPigLatin` bekommt einen `String` übergeben und soll diesen in die „ $\pi$ -Sprache“ übersetzen. Verwenden Sie hierfür die Methode `isVowel`.

### Aufgabe 2 Zahlensysteme I [Punkte: 11]

- (a) (3 Punkte) Wandeln Sie die folgenden in Binärdarstellung gegebenen Zahlen in Dezimalzahlen um. Überprüfen Sie Ihre Ergebnisse durch eine Rückkonvertierung. Ihr Rechenweg muss ersichtlich sein!

1)  $(11010110)_2$

2)  $(1011011101)_2$

3)  $(110010100111)_2$
- (b) (4 Punkte) Geben Sie für die folgenden Dezimalzahlen eine Repräsentation im Binärsystem an. Überprüfen Sie Ihre Ergebnisse durch eine Rückkonvertierung. Ihr Rechenweg muss ersichtlich sein! Wie viele Bytes benötigt man mindestens, um diese Zahlen binär codiert zu speichern?

1)  $(569)_{10}$

2)  $(6874)_{10}$

3)  $(2568)_{10}$
- (c) (4 Punkte) Wandeln Sie folgende Hexadezimalzahlen sowohl in das Dezimal- als auch in das Binärsystem um. Überprüfen Sie Ihr Ergebnis jeweils durch Rückkonvertierung. Ihr Rechenweg

---

<sup>1</sup>[https://ilias3.uni-stuttgart.de/goto\\_Uni\\_Stuttgart\\_fold\\_1318001.html](https://ilias3.uni-stuttgart.de/goto_Uni_Stuttgart_fold_1318001.html)

muss ersichtlich sein! Wie viele Bytes benötigt man mindestens, um diese Zahlen binär codiert zu speichern?

1)  $(36)_{16}$

2)  $(5B8)_{16}$

3)  $(D5A61)_{16}$

### Aufgabe 3 Impl Chiffrieren und Dechiffrieren *[Punkte: 8]*

Das Chiffrieren und Dechiffrieren von Textnachrichten hat eine lange Tradition. Eine der ältesten bekannten Chiffrierungsmethoden geht auf den römischen Feldherr Gaius Julius Caesar zurück. Daher spricht man auch von der Caesar-Verschlüsselung. Sie ist eine einfache, symmetrische Verschlüsselung, d.h. es wird immer derselbe Schlüssel für das Verschlüsseln und das Entschlüsseln verwendet. Bei der Verschlüsselung wird jeder Buchstabe des Klartexts auf einen anderen Buchstaben abgebildet. Dabei gibt der Schlüssel an, um wie viele Stellen der Buchstabe im Alphabet verschoben wird. Zum Beispiel wird aus „Hello World“ beim Schlüssel drei „Khoor Zruog“.

Implementieren Sie die Methode `cipher`, welche einen Text mittels Caesar Verschlüsselung chiffriert und alle Möglichkeiten der Chiffrierung in einer Liste zurück gibt. Die Methode soll nur ASCII<sup>2</sup> Werte zwischen 65 und 90 sowie zwischen 97 und 122 beachten. Alle anderen Werte sollen unverändert bleiben. Groß- und Kleinschreibung sollen durch die Kodierung nicht verändert werden.

### Aufgabe 4 Impl IceTeaMediaPlayer: Verwaltung von Objekten *[Punkte: 11]*

Laden Sie das aktuelle `IceTeaMediaPlayer` Projekt aus dem ILIAS herunter, in welchem sich das Codegerüst der Klasse `Player` befindet. Vervollständigen Sie die Klasse `Player`, sodass diese alle Songs und Playlists verwalten kann. Beachten Sie, dass die Methoden `opened`, `progress` und `setController` bei dieser Aufgabe nicht implementiert werden müssen. Ansonsten müssen alle Methoden der `Player`-Klasse implementiert werden. Beachten Sie bei der Implementierung die Kommentare der Methoden und implementieren Sie deren Funktionalitäten. Führen Sie für die Abgabe bitte auf dem vorgegeben `IceTeaMediaPlayer` Projekt ein Refactoring durch (in Eclipse: Rechtsklick auf Projektname - Refactor - Rename), sodass die Abgabe Ihre Nachnamen enthält.

---

<sup>2</sup><http://www.asciitable.com/>