

Ausgabe: 16.12.2019
Abgabetermin: Montag, 06.01.2020, 11:00
Besprechung: 09.01.2020

Bitte lesen Sie die folgenden Informationen zum Übungsablauf **sorgfältig** durch.

Grundsätzlich – wenn nicht anders angegeben – sind die Lösungen zu den Übungen zu Einführung in die Programmierung jeden **Montag bis spätestens 11:00 Uhr** an André Karge per E-Mail zu schicken.

Schreiben Sie bitte im Betreff Ihrer E-Mail Ihre **Teamnummer** sowie die Nummer des Übungsblattes. In der E-Mail schreiben Sie bitte zusätzlich Ihren **Namen** und **Matrikelnummer**. Die Lösungen für Sie bitte als Java Dateien als Anlage hinzu. Es werden **keine** kompilierten Dateien, wie *.class oder *.jar angenommen.

Übungen müssen von **minimal zwei** und **maximal drei** Studierenden aus derselben Übungsgruppe in einem festen Team bearbeitet werden (Ausnahmen nur auf Anfrage beim Übungsleiter). Pro Team soll die Lösung nur einmal abgegeben werden. Aufgaben sollen **im Team gelöst** und nicht nur vom Team abgegeben werden. Sie müssen mindestens **50%** dieser Punkte für eine Zulassung zur Prüfung erreichen. Das **Abschreiben** identischer Lösungen wird jeweils mit 0 Punkten bewertet.

Bei Fragen oder Unklarheiten wenden Sie sich bitte **vor der Abgabe** des Übungsblattes an den Übungsleiter (per E-Mail oder persönlich). Es soll nie jemand sagen müssen: „Wir haben die Aufgabe nicht verstanden und konnten sie daher nicht bearbeiten.“

Weitere Informationen, wie aktuelle Ankündigungen, finden Sie online (<https://www.uni-weimar.de/de/medien/professuren/intelligente-softwaresysteme/lehre/>) unter Einführung in die Programmierung

Aufgabe 1 Generics (4+5 Punkte)

- (a) Erstellen Sie eine Klasse `Pair<A, B>`, die zwei Objekte generischen Typs speichern kann. Versehen Sie diese mit folgenden Methoden:
- (i) Einem Konstruktor, mit dem beide Attribute gesetzt werden können.
 - (ii) Jeweils einen Getter für die beiden gespeicherten Attribute.
 - (iii) Jeweils einen Setter für die beiden gespeicherten Attribute.
 - (iv) Eine `toString()` Methode, die die `toString()` Methode der beiden Attribute aufruft und klammert.
- (b) Schreiben Sie ein Programm, das einen Textstring analysiert.

Schreiben Sie dazu eine Methode, die als Parameter einen `String` erhält und als Rückgabe ein `Pair<Character, Integer>` liefert, in dem sowohl das am häufigsten im String vorkommende char, als auch die Zahl der Vorkommnisse gespeichert wird. Leerzeichen sollen hierbei nicht beachtet werden. Enthält der String keine Zeichen oder wird eine `null`-Referenz übergeben, so soll `null` zurück gegeben werden. Gibt es mehr als ein Zeichen, das maximal oft vorkommt, so kann ein beliebiges maximal oft vorkommendes Zeichen zurück gegeben werden.

Geben Sie mit Hilfe dieser Methode fest programmierte Strings aus. Testen Sie ihr Programm mit folgenden Strings:

- “Many had seen it as clinching proof that the whole of known creation had finally gone bananas.”
- “ ”
- “ignorance is bliss”

Hinweis: Sie können hierzu die bereits bekannte `String`-Methode `toCharArray()` benutzen.

Aufgabe 2 Stack (6 Punkte)

Geben Sie eine Klasse `IntStack` in JAVA an, die einen Stack zum Ablegen von `int`-Werten implementiert. Dieser soll die folgenden Methoden besitzen:

- `public IntStack()` - Lege ein neues Objekt vom Typ `IntStack` an.
- `public void push(int value)` - Lege den Wert `value` oben auf dem Stack ab.
- `public int peek()` - Liefere den obersten Wert auf dem Stack zurück ohne ihn vom Stack zu entfernen.

- `public int poll()` - Liefere den obersten Wert auf dem Stack zurück und entferne ihn vom Stack.
- `public boolean isEmpty()` - Gib `true` zurück, falls der Stack leer ist, ansonsten `false`.

Die Klasse `IntStack` soll dabei auf Basis einer Listenstruktur arbeiten, die wie in der JAVA-API gegeben ist:

```
public class ArrayList<T> {

    // Erzeuge eine neue leere Liste.
    public ArrayList() {...}

    // Entferne den an der i-ten Position gespeicherten Wert aus der Liste und gib ihn zurück. Eventuell
    // nachfolgende Listenpositionen werden nach links verschoben.
    public T remove(int i) {...}

    // Belege die i-te Position der Liste mit dem Wert value. Ein eventuell bereits vorhandener Wert an
    // der Position i und eventuell nachfolgende Listenpositionen werden nach rechts verschoben.
    public void add(int i, T value) {...}

    // Liefere den Wert an der i-ten Stelle der Liste zurück.
    public T get(int i) {...}

}
```

Implementieren Sie Ihre Klasse `IntStack` derart, dass die gespeicherten Elemente in einer oben beschriebenen `ArrayList` abgelegt werden. Elemente sollen dabei am Ende der Liste hinzugefügt bzw. vom Ende der Liste entfernt werden. Die interne Arbeitsweise der Klasse `IntStack` soll nach außen verborgen bleiben.

Hinweise:

- Verwenden Sie aus der Klasse `ArrayList` ausschließlich die angegebenen Methoden. Sie dürfen die Klasse `ArrayList` nicht verändern bzw. um Methoden ergänzen. Die angegebene Klasse `ArrayList` und deren Methoden brauchen Sie nicht zu implementieren, sie können sie als gegeben betrachten.
- Die Zählung der Listenpositionen in der `ArrayList` beginnt bei 0.

Beispiel: Eine `ArrayList` sei belegt mit den Werten `[5, 7, 1]`. Nach Aufruf der Methode `add(3, 27)` enthält sie die Werte `[5, 7, 1, 27]`. Nach einem anschließenden Aufruf der Methode `remove(3)` enthält sie wieder die Werte `[5, 7, 1]`.

Aufgabe 3 Ringpuffer (5 Punkte)

Erweitern Sie die Ringpuffer Klasse `ArrayQueue` aus dem Vorlesungsskript um folgende Methoden: (Wichtig: beim Ringpuffer werden die Elemente am Anfang des Puffers entnommen (`first`) und am Ende des Puffers (`end`) eingefügt.)

- `public int peek()` gibt das erste Element aus dem Ringpuffer zurück ohne es zu entfernen.
- `public int poll()` löscht das erste Element aus dem Ringpuffer und gibt es zurück.
- `public int size()` gibt die Anzahl der momentan im Ringpuffer gespeicherten Werte zurück.
- `public String toString()` Der Ringpuffer soll den Inhalt des Puffers als String zurückgeben. Der String beginnt mit dem dem ältesten Element und endet mit dem zuletzt eingefügten Element.

Testen Sie Ihre Implementierung ausführlich und geben Sie auch das Testprogramm ab.