

## Algorithmen und Datenstrukturen

Sommersemester 2013

### 1. Übungsblatt

#### Aufgabe 1: Java - Mischen von Zahlenfolgen aus Dateien

Schreiben Sie eine Methode mit der Signatur

```
public static void mischen(Eingabe d1, Eingabe d2, Ausgabe a)
```

welche in den Eingaben `d1` und `d2` jeweils eine aufsteigende `int` - Zahlenfolge erwartet. Die Methode soll die beiden Zahlenfolgen mischen, d.h. eine aufsteigende Zahlenfolge aus den beiden Einzelfolgen bilden und in die Ausgabe `a` schreiben. Verwenden Sie die Klassen `Eingabe` und `Ausgabe` aus `simpleio` und insbesondere die Methode `eof()` aus `Eingabe` zum Erkennen des Endes einer Eingabe. Beachten Sie hierbei das Beispiel *EofAnwendung* in Abschnitt 0.1. Testen Sie Ihre Methode durch eine geeignete `main` - Methode und geeignete Eingabedateien und achten Sie insbesondere darauf, dass leere Eingabedateien abgedeckt sind. (10 Punkte)

#### Aufgabe 2: Java - Generics

Diese Aufgabe basiert auf der in der Vorlesung vorgestellten generischen Implementierung einer sortierten verketteten Liste (vgl. Abschnitt 0.3). Durch diese allgemeine Implementierung ist es möglich diese Liste auch für andere Objekttypen zu verwenden, **ohne** die Implementierung der Liste ändern zu müssen.

- a) Implementieren Sie eine Klasse `BeerBottle` und modellieren Sie darin die Bezeichnung, Brauerei, Alkoholgehalt sowie die Füllmenge. Geben Sie einen Konstruktor an und überschreiben Sie die `toString()`-Methode geeignet. (3 Punkte)
- b) Erweitern Sie `BeerBottle`, so dass unter Verwendung des Interface `MyList` und der Klasse `MyLinkedList` (in **unveränderter** Form(!)) eine Liste von Objekten dieses Typs erzeugt werden kann. Die Sortierung soll anhand der Füllhöhe (aufsteigend) realisiert werden. Sollten zwei Flaschen die selbe Füllmenge besitzen, so soll alphabetisch nach der Bezeichnung sortiert werden. Testen Sie Ihre Implementierung in einer `main`-Methode. (7 Punkte)

### Aufgabe 3: Java - Parameterübergabe mit primitiven Datentypen und Referenzen

Gegeben sei der folgende Java-Code:

```
1 package ueb01Aufg3;
2
3 public class Ueb01Aufg3 {
4
5     public static void main(String args[]){
6         int a = 1;
7         int[] b = new int[2]; b[0]=2 ; b[1] = 3;
8         Integer c = new Integer(5);
9         process(a, b, c);
10        System.out.println(a);
11        System.out.println(b[0] + " " + b[1]);
12        System.out.println(c);
13    }
14
15    public static void process(int x, int[] y, Integer z){
16        //Punkt A
17        z = new Integer(2);
18        x = 6;
19        int[] k = y;
20        k[0] = 7; k[1] = 8;
21        //Punkt B
22    }
23 }
```

Welches Ergebnis liefert die Ausführung der `main`-Methode? Begründen Sie Ihre Antwort durch ein geeignetes Diagramm, ähnlich wie in der Vorlesung in Abschnitt 0.4. Geben Sie dabei vor allem die Belegung der Variablen ( $a, b, c, x, y, z$ ) unmittelbar nach dem Eintritt (*Punkt A*) bzw. vor dem Austritt (*Punkt B*, zusätzlich Variable  $k$ ) der Methode `process(...)` an! (5 Punkte)

#### Hinweise

- Diese Hinweise gelten für dieses und alle folgenden Übungszettel dieser Veranstaltung; auch wenn diese Hinweise nicht noch einmal explizit angegeben sind.
- Rufen Sie regelmäßig die Ihnen von der Universität zugewiesene E-Mail-Adresse ab! Ankündigungen und Hinweise werden neben der Webseite zur Veranstaltung ausschließlich über diese E-Mail-Adressen verbreitet.
- Lösungen für die Übungsaufgaben sind (in der Regel) zu zweit abzugeben. Die erste Seite **muss** oben rechts den Namen und Matrikelnummer der Abgebenden sowie die Gruppennummer enthalten. Lösungen ohne Gruppennummer werden mit 0 Punkten bewertet.
- Personen, die eine Lösung zusammen abgeben, müssen der **gleichen** Gruppe angehören. Lösungen mit mehreren Gruppennummern werden mit 0 Punkten bewertet.
- Die Einteilung der Übungsgruppen erfolgt über die Anmeldung unter

<http://www.informatik-vor-2009.uni-luebeck.de/veranstaltungen/>

Dieser Link ist auch über die Übungsseite der Vorlesung erreichbar. Beachten Sie die Einteilung der Übungsgruppen nach Studiengang.

- Zertifikatskriterium: Für das Zertifikat sind 50% der erreichbaren Punkte bei den Übungsaufgaben, einmaliges Vorrechnen in der eigenen Übungsgruppe und das Bestehen der Klausur am Ende des Semesters erforderlich. Die Benotung des Zertifikats wird ausschließlich durch die Note in der Klausur bestimmt.
- **Ausnahmslos** sind Lösungen zu Programmieraufgaben **zusätzlich** per E-Mail einzureichen. Papierabgaben zu Programmieraufgaben ohne E-Mail-Einreichung bzw. E-Mails ohne Papierabgabe werden nicht gewertet. Senden Sie Ihre Quelldateien hierzu an die E-Mail-Adresse *aud?@ifis.uni-luebeck.de* wobei Sie das *?* durch Ihre Gruppennummer ( $? \in \{1, 2, 3, \dots, 10, 11, 12\}$ ) ersetzen. Fügen Sie dem Betreff Ihrer Abgabe-Mail unbedingt die Gruppennummer und die Nummer des Übungsblattes hinzu.
- Alle Abgaben zu Programmieraufgaben müssen eine ausführbare Klasse und somit eine Methode `public static void main(String[] args)` enthalten, welche die zu implementierenden Funktionen ausgiebig testet. Nichtkompilierbare bzw. nichtausführbare Abgaben führen zu erheblichem Punktabzug.
- Kommentieren Sie Ihren Quellcode ausführlich, damit Ihr Tutor Ihre guten, aber vielleicht nicht ganz korrekt umgesetzten Ideen honorieren kann.

---

**Abgabetermin:** Donnerstag, den 11. April bis 10 Uhr im Institut für Informationssysteme, 2. OG im Informatik-Neubau (Raum 2.071, Küche neben Sekretariat)