

Übungsblatt 5 **Ausgabe: Mi, 24.11.99** **Abgabe: Di, 30.11.99, 18 Uhr**

Aufgabe 1: Schrittweise Verfeinerung [4 Punkte]

Zu Semesterbeginn erstellen Sie sich üblicherweise einen Stundenplan, in dem Sie alle für Sie relevanten Veranstaltungen eintragen. Formulieren Sie dazu einen Algorithmus durch schrittweise Verfeinerung (in 2-3 Schritten) unter Angabe aller Randbedingungen, von denen Sie ausgehen (Studiengang, Semesterzahl etc.).

Aufgabe 2: Algorithmenentwurf und Java-Programmierung [8 Punkte]

Entwickeln Sie ein Verfahren, das eine beliebige Integerzahl einliest und die Ziffern aufsteigend sortiert ausgibt.

Beispiel: Eingabe 739811
Ausgabe 113789

- (a) [4 Punkte] Formulieren Sie (auf Papier) einen geeigneten Algorithmus in Pseudocode und erstellen Sie ein Flußdiagramm.
- (b) [4 Punkte] Implementieren Sie den Algorithmus als Java-Application und geben Sie das Programm über abox ab.

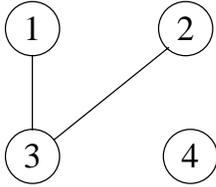
Anmerkung: Benutzen Sie den Klassennamen `Zahlensortierung`! Das Programm muß als Kommandozeilenparameter die Integerzahl erwarten und als Ergebnis die sortierten Ziffern ausgeben. Hinweise zur Ein-/Ausgabe in Java finden Sie auf den Webseiten¹ der Übung. Hier ein Beispiel für einen Kommandozeilenaufruf und der Ergebnisausgabe des von Ihnen zu entwickelnden Programms:

```
> java Zahlensortierung 421571
112457
```

Aufgabe 3: Algorithmenentwurf und Java-Programmierung [8 Punkte]

Mehrere Städte sind durch ein Straßennetz miteinander verbunden (siehe Abbildung auf der nächsten Seite). Zur Repräsentation dieses Sachverhalts im Computer wird eine Nachbarschaftsmatrix M verwendet, aus der sich erkennen läßt, ob zwei benachbarte Städte i und j eine direkte Verbindung haben ($M[i][j] = 1$) oder nicht ($M[i][j] = 0$). Diese Verbindungen sind immer bidirektional, d.h. es gibt in unserem Modell keine Einbahnstraßen.

¹<http://www.informatik.uni-mannheim.de/informatik/pi4/stud/veranstaltungen/ws199900/pi1/ueb/blatt4/io-beispiel.html>



M	1	2	3	4
1	0	0	1	0
2	0	0	1	0
3	1	1	0	0
4	0	0	0	0

Entwerfen Sie einen Algorithmus, mit dem sich überprüfen läßt, ob zwei Städte i und j , $i \neq j$, durch das Straßennetz verbunden sind.

- (a) [4 Punkte] Formulieren Sie (auf Papier) einen geeigneten Algorithmus in Pseudocode und erstellen Sie ein Flußdiagramm.
- (b) [4 Punkte] Implementieren Sie den Algorithmus als Java-Application und geben Sie das Programm über abox ab.

Anmerkung: Benutzen Sie den Klassennamen `Strassennetz`! Das Programm muß folgende Kommandozeilenparameter erwarten: (1) den Namen einer Datei, in der sich die Anzahl n der Städte sowie eine $n \times n$ -Matrix der obigen Form befindet (siehe Testdatei `netztest.txt`²), (2) die Nummer der Startstadt und (3) die Nummer der Zielstadt. Das Programm muß als Ergebnis „ja“ ausgeben, falls die beiden Städte durch das Straßennetz verbunden werden, andernfalls soll „nein“ ausgegeben werden. Hier ein Beispiel für einen Kommandozeilenaufruf und der Ergebnisausgabe des von Ihnen zu entwickelnden Programms:

```
> java Strassennetz netztest.txt 1 2
ja
```

²<http://www.informatik.uni-mannheim.de/informatik/pi4/stud/veranstaltungen/ws199900/pi1/ueb/blatt5/netztest.txt>