

Übung 05: Arrays

Abgabetermin: 18. 11. 2008

Name: _____

Matrikelnummer: _____

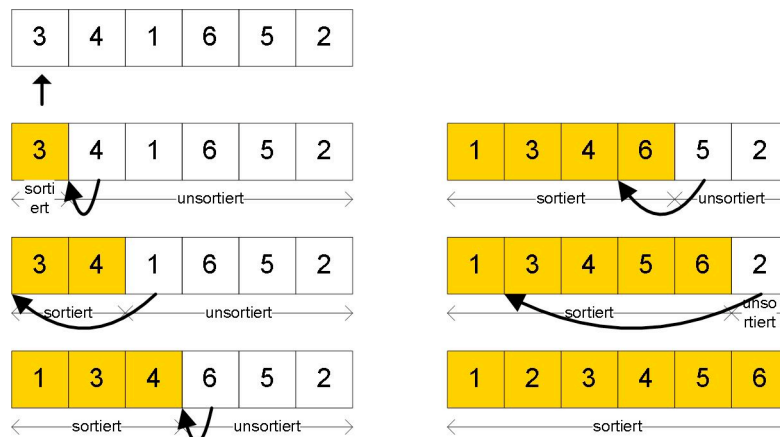
Gruppe: G1 (Prähofer) G2 (Wolfinger) G3 (Wolfinger)

Aufgabe	Punkte	gelöst	abzugeben schriftlich	abzugeben elektronisch	Korr.	Pkte
Aufgabe 05.1	12	<input type="checkbox"/>	Prosabeschreibung Ablaufdiagramm Java-Programm Testplan Testergebnisse	Java-Programm	<input type="checkbox"/>	
Aufgabe 05.2	12	<input type="checkbox"/>	Prosabeschreibung Java-Programm Testplan Testergebnisse	Java-Programm	<input type="checkbox"/>	

Aufgabe 05.1: Sortieren von Zahlen in einem Array (Einfügesortieren)

Um ein Array *a* aufsteigend zu sortieren, kann man wie folgt vorgehen:

Für jede Stelle im Array verschiebt man das Element solange nach vorne, bis es an der richtigen Stelle landet. Dabei müssen die Elemente hinter dem neu eingefügten Element nach hinten verschoben werden. Folgendes Beispiel verdeutlicht die Vorgangsweise:



Schreiben Sie dann ein Programm, das eine Reihe von Folgen von ganzen Zahlen von einer Datei einliest, die Folgen jeweils in einem Array speichert, die Arrays aufsteigend sortiert und die Ergebnisse in eine neue Datei ausgibt. Dabei seien die Zahlen in folgendem Format in der Datei gespeichert:

- Die erste Zahl bestimmt die Länge *n* der Folge; es folgt ein Doppelpunkt.
- Es folgen die *n* Zahlen in einer Zeile durch Leerzeichen getrennt.

Die Ausgabedatei soll das gleiche Format haben.

Beispiel

Eingabedatei:

```
6: 3 4 1 6 5 2
5: 4 7 2 8 7
7: 3 8 12 5 12 4 2
3: 1 1 1
```

Ausgabedatei:

```
6: 1 2 3 4 5 6
5: 2 4 7 7 8
7: 2 3 4 5 8 12 12
3: 1 1 1
```

Schreiben Sie eine Methode `void sort(int[] a)`, die das Array `a` nach dem Einfügeverfahren sortiert sowie Methoden für das Einlesen und für die Ausgabe der Arrays.

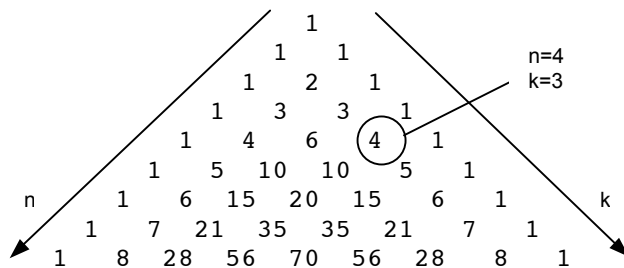
Hinweis: Wie bei In kann man mit `Out.open("dateiname")` bzw. mit `Out.close()` eine Datei für die Ausgabe öffnen bzw. wieder schließen.

Abzugeben sind:

- Die Beschreibung des Programms in Prosa
- Die Methode zum Sortieren (!) als Ablaufdiagramm
- Java-Programm
- Testfälle (d.h. eingelesenen Werte) und die Ergebnisse.

Aufgabe 05.2: Pascal'sches Dreieck

Das Pascal'sche Dreieck enthält die Binomialkoeffizienten $\binom{n}{k}$ für alle ganzen Zahlen n und k , wobei n und k zwischen 0 und einer Obergrenze m liegen. Ein Beispiel für $m=8$:



Die Werte an den Rändern ($k=0$; $k=n$) sind 1. Alle anderen ergeben sich aus der Summe der beiden darüber liegenden Zahlen:

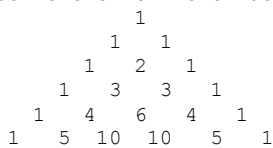
$$\binom{n+1}{k} = \binom{n}{k-1} + \binom{n}{k}$$

Schreiben Sie ein Programm das ein Pascal'sches Dreieck berechnet und auf der Konsole ausgibt.

- a) Implementieren Sie eine Methode `int[][] createTriangle(int m)`, die das Dreieck bis zu einem gegebenen m berechnet und als zweidimensionales Array mit $m+1$ Zeilen speichert. Die Elementzahl ist abhängig von der Zeilennummer - die n -te Zeile hat $n+1$ Elemente.
- b) Implementieren Sie eine Methode `void printTriangle(int[][] tri)` die das Pascal'sche Dreieck auf der Konsole ausgibt.

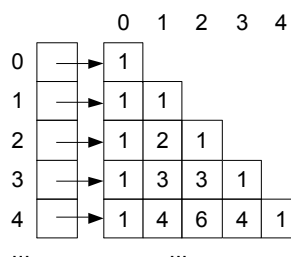
Beispieldialog

Geben Sie einen Wert fuer m ein: 5



Hinweise:

Die folgende Abbildung zeigt die Organisation des 2-dimensionalen Arrays.



Abzugeben ist:

- Die Beschreibung des Programms in Prosa
- Java-Programm
- Testfälle und die Ergebnisse.