

Übung 05: Arrays

Abgabetermin: 17. 11. 2009

Name: _____

Matrikelnummer: _____

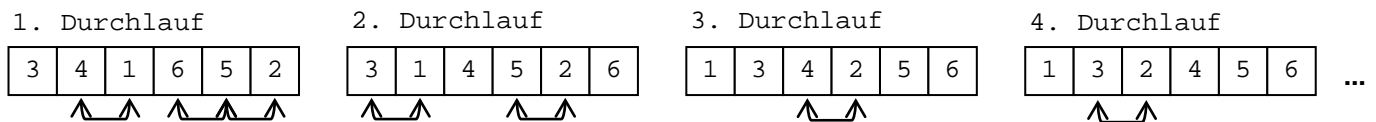
Gruppe: G1 (Prähofer) G2 (Wolfinger) G3 (Wolfinger) G4 (Jahn)

Aufgabe	Pkte	Abzugeben	gelöst	Erreichte Pkte	Kommentar
Aufgabe 05.1	12	Lösungsidee Java-Programm Testfälle und Ergebnisse	<input type="checkbox"/>		
Aufgabe 05.2	12	Lösungsidee Java-Programm Testfälle und Ergebnisse	<input type="checkbox"/>		

Aufgabe 5.1: Sortieren von Zahlen in einem Array (Austauschsortieren)

Um ein Array `a` aufsteigend zu sortieren, kann man wie folgt vorgehen: Wenn zwei benachbarte Elemente in der falschen Sortierreihenfolge stehen, vertauscht man sie. Wenn im gesamten Array keine Vertauschungen mehr möglich sind, ist das Array sortiert. Schreiben Sie eine Methode `sort(a)`, die das Array `a` nach diesem Verfahren sortiert.

Bsp.: Folgendes Beispiel verdeutlicht die Vertauschungen



Schreiben Sie dann 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
```

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

Abzugeben ist:

- Die Lösungsidee
- Das Java-Programm
- Testfälle: eingelesene Werte und Ergebnisse

Aufgabe 5.2: Spiel "Simple Battleship"

In "Simple Battleship" sucht der Spieler auf einem quadratischem Spielbrett nach versteckten U-Booten. Der Spieler schießt Wasserbomben durch Eingabe von x,y-Koordinaten auf das Spielbrett bis er alle U-Boote versenkt hat. Trifft eine Wasserbombe, zerstört die dabei entstehende Druckwelle auch alle unmittelbar benachbarten U-Boote. Entwickeln Sie dieses Spiel als JAVA-Programm.

Verwenden Sie als Hilfestellung das bestehende Programmgerüst von der Übungs-Webseite. Die Methode `createPlayingField(fieldSize, numShips)` erstellt ein Integer-Array für ein `fieldSize x fieldSize` großes Spielbrett auf dem `numShips` U-Boote zufällig angeordnet sind. Die Werte für die Parameter `fieldSize` und `numShips` soll der Spieler beim Start eingeben. Um die Algorithmen zu vereinfachen, wird das Array für das Spielbrett in jeder Dimension um 2 Reihen/Spalten größer als der eingegebene Wert erzeugt. Durch diesen Rand - in der Abbildung dargestellt durch dicke Linie - kann man für jedes Feld im Spiel auch auf alle Nachbarfelder zugreifen, ohne überprüfen zu müssen, ob diese innerhalb der Arraygrenzen liegen. Im Integer-Array hat jedes Feld einen Wert zwischen 0 und 3 - die Legende beschreibt die Bedeutung dieser Werte.

	0	1	2	3	4	
0	0	0	0	0	0	0 ... Wasser
1	0	?	0	1	0	1 ... U-Boot
2	0	1	0	2	0	2 ... Fehlschuss
3	0	3	0	1	0	3 ... Treffer
4	0	0	0	0	0	

Implementieren Sie wie folgt:

a) den Spielablauf im Hauptprogramm. Fragen Sie den Spieler auf welche Koordinaten er feuern möchte. Danach prüfen Sie mit dem Rückgabewert der `fire` Methode wie viele U-Boote der Schuss auf diese Koordinaten versenkt hat. Wiederholen Sie diesen Ablauf bis alle U-Boote versenkt sind.

b) die `fire` Methode:

```
int fire(int[][] playingField, int row, int col) { ... }
```

Prüfen Sie ob sich an den Koordinaten `row, col` ein U-Boot befindet. Bei einem Treffer setzen Sie den Feldwert auf 3 (= getroffenes U-Boot), sonst auf 2 (= Fehlschuss). Wurde ein U-Boot getroffen, prüfen Sie auch die umliegenden Felder, ob angrenzenden U-Boote versenkt werden. Zählen Sie wie viele U-Boote dieser Schuss versenkt hat und geben Sie diese Anzahl an das Hauptprogramm zurück.

c) die `printPlayingField` Methode:

```
void printPlayingField(int[][] playingField) { ... }
```

Geben Sie das aktuelle Spielfeld aus. Berücksichtigen Sie dabei nur jenen Bereich des Arrays der innerhalb des Randes (=die tatsächlichen Felder) liegt. Geben Sie für Treffer 'X', für Fehlschüsse 'O' und sonst '~' aus.

Hinweis:

Verwenden Sie die im Programmgerüst definierten Konstanten `WATER, SHIP, MISS, HIT` beim Abfragen und Setzen der Werte für die Felder am Spielbrett.

Beispiel:

Der Bildschirmdialog des Programms könnte wie folgt aussehen:

```
C:\GdP\05>java SimpleBattleShip

Bitte die Groesse des Spielfeldes eingeben: 4
Bitte die Anzahl der U-Boote eingeben: 4

Runde 1: Noch 4 U-Boote uebrig.
```

	A	B	C	D
1	~	~	~	~
2	~	~	~	~
3	~	~	~	~
4	~	~	~	~

Auf welches Feld moechten sie schiessen?
[1-4][A-D] oder [X] fuer beenden: 2B

Leider daneben.

Runde 2: Noch 4 U-Boote uebrig.

```
  A B C D
1 ~ ~ ~ ~
2 ~ O ~ ~
3 ~ ~ ~ ~
4 ~ ~ ~ ~
```

Auf welches Feld moechten sie schiessen?
[1-4][A-D] oder [X] fuer beenden: 1C

TREFFER!
TREFFER!

Runde 3: Noch 2 U-Boote uebrig.

```
  A B C D
1 ~ O X O
2 ~ O O X
3 ~ ~ ~ ~
4 ~ ~ ~ ~
```

Auf welches Feld moechten sie schiessen?
[1-4][A-D] oder [X] fuer beenden: X

Abzugeben ist:

- Die Lösungsidee
- Das Java-Programm
- Testfälle: eingelesene Werte und Ergebnisse