

Aufgabenblatt 7

Lesen Sie Kapitel 7 zu Ende. Falls Sie eine ältere Auflage des Lehrbuchs verwenden, müssen Sie sich die Abschnitte 7.3 (Iterator-Form der for-Anweisung) und 7.4 (Methoden mit variabler Parameteranzahl) kopieren. Bei Bedarf können Sie sich zusätzlich Anhang A des Lehrbuchs ansehen, um weitere Informationen zu den Klassen In und Out zu erhalten. Lösen Sie dann die folgenden Aufgaben unter Beachtung der Anforderungen an den Programmierstil. Beachten Sie, dass schlechter Programmierstil zu einem Punktabzug von bis zu 40% pro Aufgabe führen kann. Sie dürfen bei allen Aufgaben ohne Test davon ausgehen, dass der Nutzer stets sinnvolle Werte eingibt.

a) `SmallGame.java` Wir betrachten folgendes Würfelspiel für zwei Mitspieler:

Ziel des Spiels ist es, eine höhere dreistellige Zahl als der Gegner zu generieren. Die Spieler sind abwechselnd mit Würfeln an der Reihe und entscheiden sich jeweils, ob die gewürfelte Zahl auf Position 0 (Einer), Position 1 (Zehner) oder Position 2 (Hunderter) platziert werden soll. Im Anschluss wird der Gewinner festgestellt.

Schreiben Sie ein Java-Programm, das für die Spieler würfelt und zusätzlich die Rolle des Schiedsrichters übernimmt. Spieler 1 darf stets beginnen. Beispiel für Spielablauf:

```
Spieler 1 wuerfelt: 2 -> wohin? (Einer 0, Zehner 1, Hunderter 2): 2
Spieler 2 wuerfelt: 3 -> wohin? (Einer 0, Zehner 1, Hunderter 2): 1
Spieler 1 wuerfelt: 4 -> wohin? (Einer 0, Zehner 1, Hunderter 2): 0
Spieler 2 wuerfelt: 6 -> wohin? (Einer 0, Zehner 1, Hunderter 2): 2
Spieler 1 wuerfelt: 3 -> wohin? (Einer 0, Zehner 1, Hunderter 2): 1
Spieler 2 wuerfelt: 4 -> wohin? (Einer 0, Zehner 1, Hunderter 2): 0
Wert Spieler 1: 234
Wert Spieler 2: 634
Spieler 2 hat gewonnen
```

(12 Punkte)

b) `GeneralGame.java`: Erweitern Sie Ihr Programm aus Aufgabe a), so dass es

- von beliebig vielen Mitspielern gespielt werden kann und
- für beliebig lange Zahlen funktioniert.

Die Anzahl der Mitspieler und die Länge der Zahlen sollen zu Spielbeginn vom Nutzer abgefragt werden. Es müssen nur die Ergebnisse ausgegeben werden, der Sieger muss nicht ermittelt werden. Beispiel:

```
Anzahl Mitspieler: 3
Anzahl Ziffern pro Zahl: 4
Spieler 1 wuerfelt: 1 -> wohin? (Einer 0, Zehner 1 etc.): 0
Spieler 2 wuerfelt: 3 -> wohin? (Einer 0, Zehner 1 etc.): 2
Spieler 3 wuerfelt: 4 -> wohin? (Einer 0, Zehner 1 etc.): 2
Spieler 1 wuerfelt: 1 -> wohin? (Einer 0, Zehner 1 etc.): 1
Spieler 2 wuerfelt: 4 -> wohin? (Einer 0, Zehner 1 etc.): 1
Spieler 3 wuerfelt: 3 -> wohin? (Einer 0, Zehner 1 etc.): 1
Spieler 1 wuerfelt: 2 -> wohin? (Einer 0, Zehner 1 etc.): 2
Spieler 2 wuerfelt: 4 -> wohin? (Einer 0, Zehner 1 etc.): 0
```

```

Spieler 3 wuerfelt: 6 -> wohin? (Einer 0, Zehner 1 etc.): 3
Spieler 1 wuerfelt: 2 -> wohin? (Einer 0, Zehner 1 etc.): 3
Spieler 2 wuerfelt: 5 -> wohin? (Einer 0, Zehner 1 etc.): 3
Spieler 3 wuerfelt: 2 -> wohin? (Einer 0, Zehner 1 etc.): 0
Ergebnisse:
Spieler 1: 2211
Spieler 2: 5344
Spieler 3: 6432

```

(10 Punkte)

- c) FileToArray.java Die Elemente eines dreidimensionalen Arrays seien in Dateien a.txt, a0.txt bis a<z-1>.txt gegeben. Dabei enthält die Datei a.txt die Länge x, Breite y und Höhe z des Arrays. Die Dateien a0.txt bis a<z-1>.txt enthalten jeweils eine "Schicht" des Arrays, also alle Einträge, deren Index in z-Richtung der Nummer im Dateinamen entspricht. Sie können davon ausgehen, dass $z \leq 5$ gilt, es gibt also nur eine begrenzte Zahl von Dateinamen. Innerhalb einer Datei stehen die Einträge der x-Richtung neben- und die der y-Richtung untereinander. Beispiel:

```

a.txt:
5
3
2

a0.txt:
1 4 7 8 5
0 2 3 1 8
5 5 8 1 1

a1.txt:
6 3 6 7 8
2 7 1 1 9
1 0 0 2 1

```

Schreiben Sie ein Java-Programm, welches

1. die Daten aus den Dateien in ein geeignetes dreidimensionales Array überträgt
2. den Inhalt des Arrays am Bildschirm ausgibt. Verwenden Sie hierzu die Iterator-Form der for-Anweisung.
3. den Inhalt einer "Schnittfläche" am Bildschirm ausgibt. Fragen Sie dazu den Nutzer nach einer x-Koordinate. Unter "Schnittfläche" ist das Rechteck mit allen Arrayelementen zu verstehen, deren Index in x-Richtung gleich der gewünschten x-Koordinate ist.

Beispiel (zu obiger Datei a.txt):

```

1 4 7 8 5
0 2 3 1 8
5 5 8 1 1

6 3 6 7 8
2 7 1 1 9
1 0 0 2 1

```

x-Koordinate = 1	bzw. für x-Koordinate = 4
4 3	5 8
2 7	8 9
5 0	1 1

(10 Punkte)

- d) Für die folgende Aufgabe nehmen wir vereinfachend an, dass der Garbage Collector nicht mehr benötigte Speicherbereiche immer sofort freigibt (in der Realität erfolgt die Freigabe erst mit gewisser Verzögerung). Geben Sie im folgenden Programm für die mit `/* i */` gekennzeichneten Zeitpunkte der Programmausführung an, welche Arrays (ohne Berücksichtigung von args) sich momentan auf dem Heap befinden und über welchen bzw. über welche Namen sie jeweils angesprochen werden können.

Beispiel: Zum Zeitpunkt 1 befinden sich auf dem Heap:

- ein Array der Länge 2 (ansprechbar als b)
- ein Array der Länge 3 (ansprechbar als c)
- ein Array der Länge 4 (ansprechbar als d)

```
class Garbage {

    static int[] a;
    static int[] b = new int[2];

    public static void main(String[] args) {
        int[] c = new int[3];
        int[] d = new int[4];
        /* 1 */
        meth(c);
        /* 5 */
    }

    public static void meth(int[] e) {
        int[] f = new int[5];
        /* 2 */
        int[] g = e;
        /* 3 */
        a = new int[6];
        a[3] = 1;
        b = e;
        /* 4 */
    }

}
```

(8 Punkte)

Abgabetermin: Die Lösungen sind bis spätestens Donnerstag, den 7.12.2017 um 8:00 Uhr (strikt!) über das elektronische Abgabesystem einzureichen. Nachträglich eingereichte Lösungen zählen als nicht abgegeben.