

Übungen Kapitel 4

4.1 () Anweisungen

(Array Deklaration und Initialisierung)

Schreiben Sie Java Anweisungen für die folgenden Aufgaben:

- Deklariere Sie eine Konstante Variable `ARRAY_SIZE` und initialisieren diese mit 10. Eine konstante Variable ist eine Variable, deren Wert nach der Initialisierung nicht mehr geändert werden darf. Verwenden Sie hierfür das Schlüsselwort `final`.
- Deklariere Sie ein float Array mit dem Namen `fractions` der Größe `ARRAY_SIZE` und initialisieren jedes Element mit 0.
- Weisen Sie dem vierten Element den Wert 4.5 zu
- Weisen Sie dem letzten Element den Wert 5.3 zu.
- Addieren Sie alle Elemente des Arrays auf und verwenden Sie hierzu eine `for` Schleife. Die Summe speichern wir in einer Variablen `total`.

4.2 () Fehler (Papieraufgabe)

(Array Syntax)

Finden und korrigieren Sie die folgenden Fehler:

- ```
final int ARRAY_SIZE = 10;
 ARRAY_SIZE = 34;
```
- ```
int b[] = new int [10];
    for (int i = 0; i <= b.length; i++)
        b[ i ] = 1;
```
- ```
int a[][] = { {1 , 2 } , {3 , 4 } };
 a[1,1] = 5;
```

### 4.3 (\*) Sortieren (Einfaches Array)

(Einfacher Algorithmus mit Array)

Gegeben ist das Array `a` mit den Werten { 2000, 2, 45, 34, 100000, 345, 56, 78, 99, 756, 999, 0, 666, 2345, 7492, 22 }.

Schreiben Sie einen Java Algorithmus, der das Array `a` aufsteigend sortiert und lassen sich das sortierte Array `a` mit `System.out` ausgeben.

Beispiel:

Data items in original order

```
2000 2 45 34 100000 345 56 78 99 756 999 0 666
2345 7492 22
```

Data items in ascending order

```
0 2 22 34 45 56 78 99 345 666 756 999 2000
2345 7492 100000
```

**Lösungshinweis: Sie dürfen sich entweder einen Algorithmus selber ausdenken oder am einfachsten z.B. den BubbleSort Algorithmus verwenden.**

#### 4.4 (\*) Tabellenkalkulation

(Mehrdimensionales Array)

Implementieren Sie ein vereinfachtes Excel-Programm Tabellenkalkulation, das aus einer beliebigen Anzahl von Zeilen und Spalten besteht. Die Anzahl der Zeilen und Spalten gibt der Bediener auf der Console ein. Außerdem erzeugen Sie sich zufällig Werte zwischen 0 und 9 für jede einzelne Zelle. Das Programm berechnet die Summe aller Zeilen sowie die Summe aller Spalten und gibt die Tabelle mit allen Summen gemäß Beispiel wie folgt aus:

```
>Bitte Zeilenanzahl angeben
9
Bitte Spaltenanzahl angeben
10
8 7 5 6 9 5 2 0 8 2 || 52
2 7 6 3 4 2 4 2 8 1 || 39
7 8 7 6 8 2 3 8 8 9 || 66
6 8 5 4 3 2 1 8 3 9 || 49
6 5 1 2 3 6 4 2 4 3 || 36
3 6 8 1 9 5 7 7 7 3 || 56
5 8 6 2 4 7 2 2 1 0 || 37
4 4 7 7 6 2 6 0 9 3 || 48
2 0 6 9 5 7 5 2 5 5 || 46
=====
43 53 51 40 51 38 34 31 53 35
```

Die Ausgabe darf für eine größere Anzahl von Zeilen und Spalten noch ungeeignet sein.

#### 4.5 (\*) Sätze

(String Array)

Schreiben Sie eine Anwendung, die zufälligerweise Sätze generiert. Dazu verwenden Sie vier Arrays: `article`, `noun`, `verb`, und `preposition`. Aus jedem dieser Arrays nehmen Sie zufallsmässig ein Wort und stellen dann einen Satz daraus zusammen. Die einzelnen Wörter sollen durch blank getrennt sein. Jeder Satz endet mit einem Punkt. Das Programm soll 20 Sätze generieren.

Die einzelnen Arrays sollen mit folgenden Wörtern gefüllt werden:

`article`: " the" , " a" , " one" , " some" , " any"

`noun`: " boy" , " girl" , " dog" , " town" , " car"

`verb`: " drove" , " jumped" , " ran" , " walked" , " skipped"

`preposition`: " to" , " from" , " over" , " under" , " on" .

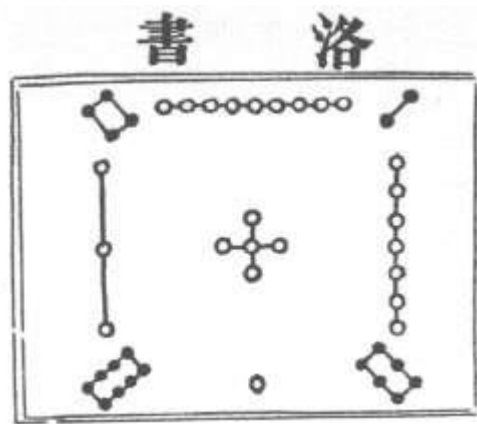
Beispiel:

```
a car ran under one car.
the dog jumped over some boy.
some dog drove from the town.
the car skipped under a boy.
a boy skipped over one dog.
any girl drove on the town.
one car ran to some boy.
a boy walked over a dog.
the dog ran from any boy.
```

**Lösungshinweis: Um ein Wort zufällig auszuwählen, bedient man sich des Tricks, dass man die Strings in entsprechenden arrays ablegt und sich zufällig korrekte Indices durch `Math.random` für diese Arrays erzeugt.**

**4.6 (\*) Magisches Quadrat (Permutationen)** (Algorithmus, mehrdimensionales Array)

Im alten China regierte von 2205 bis 2198 v. Chr. ein gerechter und weiser Herrscher, der Kaiser Yü, der sein Reich mit großer Umsicht verwaltete. Konfuzius berichtet, der Kaiser sei einmal damit beschäftigt gewesen, Dämme zu bauen, um Überschwemmungen des gelben Flusses Einhalt zu tun. Als er in Gedanken versunken war, erschien ihm die göttliche Schildkröte, die den Namen Hi trug. Auf dem Rücken der Schildkröte war eine Figur gezeichnet, die mit Zahlzeichen versehen war:



Die Aufschrift der Schildkröte entsprach dem magischen Quadrat "Lu Sho" der Kantenlänge  $n = 3$ :

|   |   |   |
|---|---|---|
| 4 | 9 | 2 |
| 3 | 5 | 7 |
| 8 | 1 | 6 |

Ein magisches Quadrat ist ein Zahlenfeld mit  $n$  Zeilen und  $n$  Spalten, das mit den natürlichen Zahlen von 1 bis  $n^2$  gefüllt wird - und zwar derart, dass die Zahlensumme in jeder Zeile, Spalte und den Hauptdiagonalen gleich ist.

Ein Algorithmus zur Ermittlung der magischen Quadrate **für ungerade  $n$**  ist der Algorithmus des Mathematikers Claude Gaspar Bachet.

Implementieren Sie ein Programm zur Ermittlung der magischen Quadrate für ungerade  $n$  wie folgt:

1. Lesen Sie den Wert  $n$  von der Console ein, wobei für  $n$  nur gelten darf:  $2 < n < 10$ . Falls der Bediener einen falschen Wert für  $n$  eingibt, wird er nochmals aufgefordert eine neue Zahl einzugeben, solange bis der Wert von  $n$  korrekt ist.

2. Erzeugen Sie ein zweidimensionales Array `quad` mit der Länge `n`.

Ermittlung des magischen Quadrats für `n` nach dem Algorithmus von Bachet:

3. Deklarieren Sie drei Variablen `zeile`, `spalte` und `m` und initialisieren Sie `zeile` mit `n/2`, `spalte` mit `n/2 + 1` sowie `m` mit `0`

( Bedenken Sie: Die Felder werden von 0 ab durchnummeriert.)

4. Führe durch

1. Inkrementieren Sie `m`.
2. Tragen Sie an der aktuellen `zeile` und `spalte` Position `m` ein.
3. Dekrementieren Sie `zeile` und inkrementieren Sie `spalte`.
4. Wenn `zeile` aus dem Zahlenbereich des Feldes herauslaufen würde, setzen Sie `zeile` statt auf `-1` auf den Wert `n-1`
5. Wenn `spalte` aus dem Zahlenbereich des Feldes herauslaufen würde, setzen Sie `spalte` statt auf `n` auf den Wert `0`.
6. Wenn die Position an dieser Stelle bereits mit einer Zahl belegt ist,
  - a. Inkrementieren Sie `spalte` und `zeile`
  - b. Wenn dadurch `spalte` bzw. `zeile` aus dem Zahlenbereich des Feldes herausläuft, setzen Sie `spalte` bzw. `zeile` statt auf `n` auf den Wert `0`.

solange `m < n2`

5. Geben Sie die Werte von `quad` auf der Console aus.

6. Testen und verifizieren Sie Ihr Programm für die Werte `n` gleich `1, 3, 5, 7, 9, 10`.

|                                               |                                               |
|-----------------------------------------------|-----------------------------------------------|
| <b>4.7 (**) Zusatzaufgabe: Projektaufgabe</b> | <i>(Algorithmus, mehrdimensionales Array)</i> |
|-----------------------------------------------|-----------------------------------------------|

Sofern Sie noch nicht mit dem Projekt angefangen haben, beginnen Sie jetzt mit Projektaufgabe Teil 1 (s. moodle).