JAVA: VORBEREITUNG BLOCKKURS-KLAUSUR

Übungsblatt 6

Janosch Fuchs
Jürgen Dietel, Abdelrahman Elhabashy, Timon Römer, Daniel Weinholz
MATSE Gruppe
ITC - RWTH Aachen

WS24/25
18.09.2024
Abgabe: —

- Die Übungsblätter sollten alleine bearbeitet werden, aber Hilfe von den Übungsleitern oder Studierenden ist gewünscht.

Übungsaufgabe 6.1

Notieren Sie die Antworten in ein oder mehreren Stichworten oder einem kurzen Satz:

(a) Wie lautet der Funktionskopf der `main`-Funktion?

(b) Wie lautet die Anweisung, um aus einer Schleife zu springen?

(c) Wie wandelt man eine `double`-Variable `d=2.5` in eine `int`-Variable um und welchen Wert besitzt die `int`-Variable anschließend?

(d) Eine Variable, die direkt zu einem Objekt gehört, heißt:

(e) Mit welchen Werten ist ein int-Feld vorinitialisiert?

(f) Welche Werte kann eine `boolean`-Variable annehmen?

(g) Wie kann man in Java eine Textzeile von der Tastatur einlesen?

(h) Der Wertebereich von `int` liegt etwa zwischen ... und ...

(i) Wie lautet die Deklaration einer `int`-Variable mit Namen x?

(j) Was ist der ASCII-Wert des Zeichens `'2'`?

(k) Was ergibt der Ausdruck `'X'-'G'`?

(l) Was ergibt der Ausdruck `(char)('x'-'A')`?

(m) Ein Java-Programm enthält folgende Zeilen:

```
Bruch b = new Bruch(1,2);
b = null;
```

Was passiert anschließend mit dem Bruch-Objekt?

Übungsaufgabe 6.2

(a) Schreiben Sie eine Funktion

```
public static double getMax(double d1, double d2, double d3)
```

die das Maximum der 3 Zahlen d_1, d_2 und d_3 zurückgibt.

(b) Schreiben Sie die Programmzeilen, die den Wert dreier int-Variablen a, b und c zyklisch vertauschen.

Beispiel: Aus $\begin{pmatrix} a=1 \\ b=3 \\ c=5 \end{pmatrix}$ wird $\begin{pmatrix} a=5 \\ b=1 \\ c=3 \end{pmatrix}$.

(c) Schreiben Sie eine Funktion

```
public static int XOR(boolean b1, boolean b2)
```

die die sogenannte XOR-Verknüpfung der beiden Übergabeparameter ermittelt. Ist **genau einer** der beiden Parameter `true`, so gibt sie 1 zurück, sonst 0.

(d) Ein Palindrom ist eine Zeichenkette, die vorwärts wie rückwärts gelesen gleich ist. Die Zeichenreihen `"OTTO"`, `"NEBEN"` oder `"LAGERREGAL"` sind z.B. Palindrome.

Schreiben Sie eine Funktion, die als Parameter die zu prüfende Zeichenkette bekommt und als Ergebnis zurückgibt, ob es sich um ein Palindrom handelt (`true`) oder nicht (`false`).

(e) Schreiben Sie eine Funktion ohne Übergabeparameter, die aus den ganzen Zahlen von 1 bis 49 zufällig 6 Zahlen auswählt. Diese 6 Zahlen sollen in einem Integer-Array gespeichert und dieses Feld als Return-Wert zurückgegeben werden. Die Zahlen können mehrfach ausgewählt werden.

(f) Schreiben Sie eine Funktion

```
public static boolean isRichtig(int n, int[] gewinnzahlen)
```

die `true` zurückgibt, falls n mindestens einmal im Feld `gewinnzahlen` enthalten ist und `false` sonst.

(g) Schreiben Sie eine Funktion

```
public static int[][] getIntArray(int x, int y, int val)
```

die ein Integer-Feld der Größe $x \times y$ zurückgibt, das mit dem Wert `val` gefüllt ist.

(h) Schreiben Sie eine Funktion

```
public static int[][][] erzeugeIndexSummenFeld(int n) {
```

die ein 3-dimensionales Feld der Größe $n \times n \times n$ erzeugt, in der jeder Eintrag die Summe seiner Indizes sein soll, d.h.

$$a[i][j][k] = i + j + k$$

(i) Ein Text kann nach dem folgenden einfachen Verfahren verschlüsselt werden. Jedes Zeichen wird um eine Position im Alphabet weitergeschoben. Aus einem A wird ein B, aus einem B ein C usw. Aus einem Z wird wieder ein A. Andere Zeichen als (Groß- und Klein-)Buchstaben bleiben unverschlüsselt. Schreiben Sie eine Funktion

```
public static String verschluessele(String klartext)
```

die den String `klartext` nach dem oben angegebenen Verfahren verschlüsselt.

(j) Schreiben Sie eine Funktion

```
public static int kleinsteZiffer()
```

Die Funktion liest einen String von der Tastatur ein und gibt die kleinste Ziffer (0-9) zurück, die in dem String enthalten ist. Wenn keine Ziffer enthalten ist, wird -1 zurückgegeben.

(k) Schreiben Sie eine Funktion

```
public static boolean istAlternierend(int x)
```

Die Funktion soll genau dann `true` zurückgeben, wenn die Dezimalziffern der Zahl x abwechselnd gerade und ungerade sind. Für negative Werte von x wird der Betrag von x genommen.
Beispiel:
Die Zahl 3652 ist alternierend, weil die Ziffern 3, 6, 5 und 2 abwechselnd gerade und ungerade sind. Auch 27432 ist alternierend, nicht jedoch 4635. Auch einziffrige Zahlen (1 oder 8) sind alternierend.

Übungsaufgabe 6.3

Schreiben Sie eine Funktion

```
public static void berechnePrimzahlen(int n)
```

Die Funktion soll alle Primzahlen von 2 bis n ausgeben. Eine einfache Methode, um festzustellen, ob eine Zahl n eine Primzahl ist, besteht darin, zu überprüfen, ob $n \mod i \neq 0$ für alle i zwischen 2 und $n - 1$. Sie dürfen auch andere Lösungswege verwenden.

Übungsaufgabe 6.4

Schreiben Sie eine Funktion

```
public static boolean isEinheitsmatrix(int[][] matrix)
```

Die Funktion gibt genau dann `true` zurück, falls die übergebene Matrix eine Einheitsmatrix ist. Dies ist dann der Fall, wenn einerseits die Matrix quadratisch ist und andererseits für die Elemente gilt:

$$a_{i,j} = \begin{cases} 0; & i \neq j \\ 1; & i = j \end{cases}$$

Übungsaufgabe 6.5
Schreiben Sie eine Funktion

```
public static int[] zaehleBuchstaben(String txt)
```

Die Funktion gibt ein Feld von 26 Integerzahlen zurück, die die Häufigkeit der Buchstaben A-Z im gegebenen String enthalten. Der String darf Groß- und Kleinbuchstaben enthalten; zwischen diesen wird jedoch nicht unterschieden. Andere Zeichen als Buchstaben werden nicht gezählt.

Übungsaufgabe 6.6
Schreiben Sie eine Funktion

```
public static void printTripel()
```

Diese Funktion gibt alle Tripel (x, y, z) auf dem Bildschirm aus, für die gilt:

$$x^2 + y^2 = z^2$$

Dies sind die sogenannten Pythagoreischen Tripel. x, y und z sind ganze Zahlen und sollen im Bereich $1 \ldots 99$ liegen. Außerdem soll gelten: $y > x$. Das erste Tripel ist $(3, 4, 5)$, denn $3^2 + 4^2 = 9 + 16 = 25 = 5^2$.

Übungsaufgabe 6.7
Schreiben Sie eine Funktion

```
public static int[] extrahiereZiffern(String s)
```

die aus einer gegebenen Zeichenkette ein Feld aller darin befindlichen Ziffern erzeugt. Alle anderen Zeichen sollen somit missachtet werden. Beispiel: "'1hjp33ts7jd1"' ergibt [1, 3, 3, 7, 1].

Übungsaufgabe 6.8
Schreiben Sie eine Funktion

```
public static int bestimmeHaufigsteZiffer(int i)
```

die von einer übergebenen Zahl die häufigste Ziffer bestimmt und zurück gibt. Achten Sie darauf, dass die Zahl auch negativ sein kann. Falls mehrere Ziffern gleich oft vorkommen, soll die größte dieser Zahlen zurückgegeben werden.

Übungsaufgabe 6.9
Schreiben Sie eine Funktion

```
public static void dreheFeld(int[][] f)
```

Die Funktion erwartet ein quadratisches Feld (diese Bedingung muss nicht überprüft werden). Dieses Feld wird um 90 Grad nach rechts gedreht. Beispiele:

Feld vorher	Feld nachher
1 2 3 4 5 6 7 8 9	7 4 1 8 5 2 9 6 3
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16	13 9 5 1 14 10 6 2 15 11 7 3 16 12 8 4

Übungsaufgabe 6.10

Manche Zahlen mit 3 Ziffern (zwischen 100 und 999) haben eine interessante Eigenschaft: Addiert man die dritten Potenzen der drei Ziffern, kommt die gleiche Zahl wieder heraus. Eine dieser Zahlen ist beispielsweise die 153. Beispiel:

$$1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153$$

Schreiben Sie eine Funktion

```
public static void printZahlen()
```

die alle dreiziffrigen Zahlen mit dieser Eigenschaft auf dem Bildschirm ausgibt.

Übungsaufgabe 6.11

Schreiben Sie eine Klasse `Counter`. Ein Objekt dieser Klasse wird gebraucht, um Dinge zu zählen. Als Anforderung soll die Klasse folgende Methoden enthalten:

- `public void reset()`: Setzt den Zähler auf 0 zurück.
- `public void increment()`: Zählt den Zähler um 1 hoch.
- `public void decrement()`: Zählt den Zähler um 1 herunter. Bei 0, behält der Zähler den Wert 0.
- `public int getCounter()`: Gibt den Wert des Zählers zurück.
- `public void printCounter()`: Gibt den Wert des Zählers auf dem Bildschirm aus.

Übungsaufgabe 6.12

Schreiben Sie eine Klasse `Matrix`, die eine zweidimensionale quadratische Matrix aus double-Elementen enthält. Als Anforderung soll die Klasse folgende Konstruktoren und Methoden erhalten:

- `public Matrix(int n)`: Dieser Konstruktor erzeugt ein Matrix-Objekt mit n Zeilen und n Spalten.
- `public double getElement(int zeile, int spalte)`: Gibt den Wert des Matrixelements an der Position *(zeile/spalte)* zurück. Das linke obere Element wird mit `getElement(0,0)` ausgelesen. Es muss nicht überprüft werden, ob *zeile* und *spalte* korrekte Werte enthalten.
- `public void setElement(int zeile, int spalte, double wert)`: Setzt das Matrixelement *(zeile/spalte)* auf den Wert *wert*. Die Nummerierung der Elemente ist entsprechend der getter-Methode. Es muss nicht überprüft werden, ob *zeile* und *spalte* korrekte Werte enthalten.
- `public int getSize()`: Gibt die Größe der Matrix zurück. Bei einer 3×3-Matrix würde der Wert 3 zurückgegeben. Hinweis: Sie können sicher sein, dass die Matrix quadratisch ist.
- `public void mult(double fakt)`: Multipliziert jedes Matrixelement mit dem Faktor *fakt*.

Übungsaufgabe 6.13

Schreiben Sie eine Klasse `Zylinder`, mit der die geometrische Figur eines Zylinders mit kreisrunder Grundfläche verwaltet werden kann. Damit ist die Figur durch die Höhe h und den Radius der Grundfläche r eindeutig bestimmt.

- `public Zylinder (double h, double r)`: Dieser Konstruktor erzeugt ein Zylinder-Objekt mit Höhe h und Radius r.
- `public Zylinder ()`: Dieser Konstruktor soll Höhe und Radius auf 1.0 setzen.
- Neben den Konstruktoren werden Getter- und Setter-Methoden für die Werte h und r benötigt.
- `public double getVolumen ()`: Berechnet das Volumen nach der Formel $V = \pi \cdot r^2 \cdot h$ und gibt dieses zurück.
- `public double getOberflaeche ()`: Berechnet die Oberfläche nach der Formel $F = 2 \cdot \pi \cdot r^2 + 2 \cdot \pi \cdot r \cdot h$ und gibt diese zurück.
- Die Zeilen

```
Zylinder z = new Zylinder(3,5);
System.out.println(z);
```

sollen folgende Ausgabe erzeugen:

```
Zylinder: Hoehe: 5.0000, Radius: 3.0000
```

Die genaue Formatierung der Zahlen ist unwesentlich. Ergänzen Sie dazu die Methode `toString()`.