Aufgabenblatt #2

In Ihrer Übungsumgebung unter Linux am Informatik-Labor finden Sie das Projekt Aufgabenblatt2. Dieses Projekt besteht aus fünf Aufgaben, die Sie erweitern müssen.

Abgabe: Die Abgabe muss bis Montag, den 17.11. um 16:00 erfolgen. Alle Änderungen am Projekt in der Übungsumgebung, die bis zu diesem Zeitpunkt von Ihnen vorgenommen wurden, werden von uns als Abgabe betrachtet. Bitte beachten Sie folgende Punkte:

- Dieses Aufgabenblatt wird benotet.
- Verändern Sie bitte nicht die Ordnerstruktur oder die Dateinamen!
- Ändern Sie nur die Inhalte der Dateien Aufgabe1.java - Aufgabe5.java.
- Die Antworten zu den Zusatzfragen können bei den jeweiligen Aufgaben als Kommentare geschrieben werden.

Aufgabe 1: Rekursion

Implementieren Sie eine rekursive Methode `zahlInHex`, die für eine ganze positive Zahl n die Darstellung im Hexadezimalsystem als String zurückgibt. Z.B. sollte für n = 1234 der String 4D2 erzeugt werden. Sie dürfen keine speziellen Methodenaufrufe aus der Java-API benutzen.

Zusatzfragen:

- Wie kann man das Verhalten dieser Methode (unabhängig von der Rekursion) mit einem Aufruf aus der Java-API erreichen?
- Welche Vorteile ergeben sich aus der Benutzung von Methoden aus der Java-API?

Aufgabe 2: Rekursion

Implementieren Sie eine iterative (`iterativLeo`) und eine rekursive (`rekursivLeo`) Methode, die für eine ganze positive Zahl n die sogenannte Leonardo-Zahl $L(n)$ berechnet. $L(n)$ ist definiert durch:

$$L(n) = \begin{cases} 1 & \text{wenn n} = 0 \\ 1 & \text{wenn n} = 1 \\ L(n-1) + L(n-2) + 1 & \text{wenn n} > 1 \end{cases}$$

Rufen Sie die Methode mit mehreren kleinen Zahlen ($n < 30$) in der `main` Methode auf.

Zusatzfragen:

- Wann würden Sie bei diesem Beispiel die iterative Version bevorzugen?
- Welche Nachteile kann eine rekursive Lösung bei diesem Beispiel haben?

Aufgabe 3: Rekursion

Implementieren Sie eine Methode `printSequenz` für folgende Ausgabe für eine ganze positive Zahl $n > 0$ mit einem rekursiven(!) Verfahren.

Ausgabe für n = 3	Ausgabe für n = 5
1	1
12	12
123	123
12	1234
1	12345
	1234
	123
	12
	1

Vor der Ausgabe der Zahlenpyramide soll keine zusätzliche Leerzeile vorhanden sein, d.h. die Ausgabe beginnt mit 1. Nach der letzten Ausgabe der Zahl 1 soll die Ausgabe in die nächste Zeile springen. Ist $n < 1$, dann soll nichts ausgegeben werden.

Aufgabe 4: Eindimensionales Array

Implementieren Sie folgende Methoden für das gegebene Hauptprogramm:

- `befuelleArray`: Befüllt ein übergebenes int-Array mit Zufallszahlen im Bereich 0 (inklusive) bis N (inklusive). N wird auch als Parameter übergeben. Für die Generierung von Zufallszahlen können Sie `Math.random()` verwenden: `http://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#random()`
- `generiereStatistik`: Berechnet für ein übergebenes int-Array und Parameter N wie oft jede Zahl von 0 bis N in diesem Array vorkommt und gibt jede Anzahl in einer eigenen Zeile aus.

Beispiel:

Array (Zahlen von 0 bis 4): [1, 0, 3, 2, 3, 4, 1, 0, 4, 2, 3, 1, 3, 0, 1]
Ausgabe:
Zahl 0 kommt 3x vor
Zahl 1 kommt 4x vor
Zahl 2 kommt 2x vor
Zahl 3 kommt 4x vor
Zahl 4 kommt 2x vor

Zusatzfragen:

- Welche Werte kann man in einem Array vom Typ `int[]` abspeichern?
- Warum sind bei dieser Aufgabe Arrays hilfreich?
- Wie würden Sie diese Aufgabe ohne Arrays lösen?

Aufgabe 5: Zweidimensionales Array

Implementieren Sie folgende Methoden für das gegebene Hauptprogramm:

- `generiereMatrix`: Generiert für ein gegebenes N eine $N \times N$ Matrix (zweidimensionales int-Array) und gibt diese zurück. Die einzelnen Einträge ergeben sich aus einer laufenden Nummer. Für eine $N \times N$ Matrix gibt es N^2 Nummern von 0 bis $N^2 - 1$ und diese werden zeilenweise vergeben. Beispiel für N = 3:
  0 1 2
  3 4 5
  6 7 8
- `printMatrix`: Gibt eine $N \times N$-Matrix aus (N Zeilen mit jeweils N Spalten)
- `transponiereMatrix`: Transponiert eine $N \times N$-Matrix ohne eine zusätzliche Matrix, d.h. die übergebene Matrix ist nach der Ausführung der Methode in transponierter Form vorhanden.

Zusatzfragen:

- Wie werden zweidimensional Arrays in Java vom Aufbau her realisiert?
- Kann man auch mehrdimensionale (3 oder mehr Dimensionen) Arrays in Java anlegen? Wenn ja, geben Sie ein Beispiel für eine Deklaration an!