

Einführung in die Programmierung
WS 2014/15

Übungsblatt 10: Mehr Imperative Programmierung mit Arrays

Besprechung: 07./09./12.01.2015

Ende der Abgabefrist: Dienstag, 23.12.2014 14:00 Uhr.

Hinweise zur Abgabe:

Geben Sie bitte Ihre gesammelten Lösungen zu diesem Übungsblatt in einer Datei `loesung10.zip` unter <https://uniworx.ifi.lmu.de> ab.

Alle Code-Aufgaben müssen als `.java` abgegeben werden. PDF-code wird nicht akzeptiert. Nicht kompilierbare Abgaben werden mit 0 Punkten bewertet.

Aufgabe 10-1 *Algorithmen*

1+2+3 Punkte

- (a) Implementieren Sie eine statische Methode `boolean istQuadratzahl(int zahl)` in Java, die den Wahrheitswert `true` zurückgibt, falls `zahl` eine Quadratzahl ist.
- (b) Implementieren Sie eine statische Methode `int[] fib(int n)` in Java, die die ersten `n` Fibonacci-zahlen in einem Array der Länge `n` zurückgibt. Die Fibonacci-Folge ist eine unendliche Folge von Zahlen, beginnend mit der Zahl 0, bei der die Summe zweier benachbarter Zahlen die unmittelbar folgende Zahl ergibt: 0, 1, 1, 2, 3, 5, 8, 13,...
- (c) Implementieren Sie eine öffentlich sichtbare statische Methode `int[] countElements(int[] werte, int max, int numBuckets)` in Java, die das Integer-Intervall $[0, \text{max}-1]$ in gleich große zusammenhängende Teilintervalle unterteilt und in einem Ergebnisarray des Typs `int[]` die Anzahl der Array-Elemente aus `werte` zurückgibt, die in das entsprechende Teilintervall fallen. Sie können davon ausgehen, dass beide Arrays eine Länge von mindestens 1 haben, dass $\text{max} > 0$ und max durch `numBuckets` teilbar ist. Außerdem können Sie annehmen, dass alle Zahlen in `werte` im Intervall $[0, \text{max}-1]$ liegen.

Beispiel: Gegeben sei ein Array `werte = {0, 9, 2, 3, 4, 5, 7, 8, 1}`, `max = 10` und `numBuckets = 2`. Das Intervall $[0, \text{max}-1]$ wird also in `numBuckets = 2` gleich große Teilintervalle $[0, 4]$ und $[5, 9]$ zerlegt. Ausgegeben wird ein Array mit `numBuckets = 2` Elementen.

1. Feld des Arrays: wie viele Werte kommen in $[0, 4]$ vor? $\rightarrow 5$

2. Feld des Arrays: wie viele Werte kommen in $[5, 9]$ vor? $\rightarrow 4$

Das heißt, als Ergebnis wird `{5, 4}` zurückgegeben.

Aufgabe 10-2 *Matrizen in Java***1+1+1+2 Punkte**

Eine Matrix, die aus Zeilen und Spalten besteht, kann wie folgt veranschaulicht werden:

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{pmatrix}$$

Statt Zahlen können an jeder Stelle auch allgemein Objekte stehen. Als (nicht objekt-orientierte!) Repräsentation in Java wählen wir ein Array von Arrays. Eine Variable, die obige Matrix repräsentiert, können wir dann wie folgt definieren:

```
int[][] matrix = {  
    {1, 2, 3},  
    {4, 5, 6},  
    {7, 8, 9},  
    {10,11,12}  
};
```

Die inneren Arrays repräsentieren also Zeilen. Spalten werden durch die Einträge der inneren Arrays mit gleichem zweitem Index gebildet.

Sie können das Klassengerüst in der Datei `Matrix.java`, die Sie auf der Vorlesungshomepage finden, verwenden.

Außerdem nehmen wir an, dass in einer Matrix nach obigem Schema immer alle Zeilen die gleiche Länge haben.

- (a) Implementieren Sie die Methode `boolean gleich(int[] a1, int[] a2)`, die für zwei beliebige `int`-Arrays entscheidet, ob sie genau die gleichen Werte in der gleichen Reihenfolge enthalten.
- (b) Implementieren Sie die Methode `boolean enthaeltZeile(int[][] matrix, int[] zeile)`, die für eine Matrix nach obigem Schema entscheidet, ob eine gegebene Zeile enthalten ist.
- (c) Implementieren Sie die Methode `boolean enthaeltSpalte(int[][] matrix, int[] spalte)`, die für eine Matrix nach obigem Schema entscheidet, ob eine gegebene Spalte enthalten ist.
- (d) Implementieren Sie die Methode `int[][] transponieren(int[][] matrix)`, die eine Matrix transponiert. Bei der Transponierten einer Matrix werden Spalten und Zeilen vertauscht. Zum Beispiel ist die Transponierte der obigen Matrix gegeben durch:

$$\begin{pmatrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \end{pmatrix}$$

Aufgabe 10-3 *Addition von natürlichen Zahlen in p -adischer Darstellung***0 Punkte**

Die Dokumentation der Klasse `Addition`, die Sie auf der Vorlesungswebseite finden, spezifiziert bereits, was diese Klasse leisten soll. Sie müssen die Datei nur entsprechend ergänzen:

- (a) Ergänzen Sie den Rumpf der Methode `alphabet`, so dass das Array `ALPHABET` für das Alphabet mit den Zeichen `0, ..., 9, A, ..., Z, a, ..., z` geschickt initialisiert wird.
- (b) Für die Addition werden zwei Zahlen in p -adischer Darstellung als Arrays der entsprechenden Zeichen erwartet sowie eine Basis p (in Dezimalschreibweise) der Darstellung. Das Ergebnis soll die Summe der Zahlen wiederum als Array der Zeichen (mit Basis p) ohne führende Nullen sein. Ergänzen Sie die Methode `summe`, so dass Sie diese Aufgabe erfüllt.