

Wintersemester 2024/25
Prof. Dr. J. Rexilius

Abgabetermin: 07.11.2024, 08:00

Aufgabe 3 (9 Punkte)

Beachten Sie die aktuelle Version zu den allgemeinen Abgabehinweisen (Praktikumsordner im ILIAS).

(a) Arrays (3 Punkte)

In dieser Aufgabe sollen 2-dimensionale Arrays auf 1-dimensionale Arrays nach bestimmten Regeln reduziert werden. Gegeben sei dazu ein 2D Array der Größe $N \times N$ mit Integer-Werten $v \geq 0$.

Implementieren Sie eine Klasse `MyArray` mit mindestens folgenden Methoden (nutzen Sie keine Methoden der Klasse `Math` oder aus anderen Klassen wie `Arrays` oder dem `Collection Framework`):

- `reduce(int[][] arr)`
Reduziert das 2D Array `arr` auf ein 1D Array. In jedem Eintrag des 1D Array werden dabei die Anzahl der Einträge in `arr` für einen Werte in einen Eintrag des 1D Array geschrieben. Die Größe des 1D Array ergibt sich als $s = (\max_{arr} - \min_{arr} + 1)$. Ein Element k im 1D Array enthält damit die Summe Einträge im 2D Array mit $arr[i,j]=k$, $arr[i,j] \in [\min_{arr}, \max_{arr}]$. Verteilen Sie auf diese Art alle Zahlen im 2D Array `arr` auf das 1D Array.
- `ausgabe()`
Gibt den Inhalt des 1D Array auf der Konsole aus.

Testen Sie Ihre Implementierung für ein 2D Array der Größe $N=4$ und ein 2D Array der Größe $N=5$.

Beispiel:

2D Array ($arr[i,j] \in [1,7]$)

1, 2, 3
4, 1, 2
7, 5, 4

Ergebnis nach `reduce(...)`: 2, 2, 1, 2, 1, 0, 1

(`arr1D[0]` enthält die Anzahl der Werte k in `arr` für $k=1$, `arr1D[1]` die für $k=2$, etc.)

(b) Vergleich von Arrays (3 Punkte)

Implementieren Sie eine Methode `boolean isEqual(int[] a, int[] b)`, die zwei Arrays miteinander vergleicht. Testen Sie Ihre Methode mindestens mit einem positiven (Ergebnis=`true`) und einem negativen Beispiele (Ergebnis=`false`), d.h. mit Beispielen bei denen das Ergebnis für `isEqual()` die Werte `true` bzw. `false` zurückliefert.

Beispiel:

Input: `a=[1,2,3,4]`, `b=[3,1,6,2]` -> Output: `false`

Input: `a=[3,5,1,3]`, `b=[1,5,3,3]` -> Output: `true`

Wintersemester 2024/25
Prof. Dr. J. Rexilius

Abgabetermin: 07.11.2024, 08:00

Hinweis:

- Zwei Arrays sind genau dann gleich, wenn sie die gleiche Länge haben und alle ihre Komponenten paarweise übereinstimmen. Es muss für jedes Element in einem Array *a* auch ein eigenes Element in einem Array *b* geben, d.h. wenn in Array *a* ein Wert doppelt vorkommt, dann muss dieser in einem gleichen Array *b* auch doppelt vorkommen.
- Die Reihenfolge der Werte in den beiden Arrays soll egal sein. D.h. ein Vergleich von *array1* mit Einträgen *[1, 2, 3]* und *array2* mit Einträgen *[3, 1, 2]* soll *true* zurückliefern.
- Pro Array kann ein Eintrag auch mehrmals vorkommen, z.B. ein Array mit Einträgen *[1, 1, 1]*.
- Nutzen Sie keine Methoden aus einer bereits vorhandenen Java-Klassen. Auch nicht die Methode *sort* aus der Klasse *Arrays*.
- Der Inhalt der Arrays darf in Ihrer Implementierung nicht verändert werden. Auch nicht die Positionen der Werte.

(c) Auf den Kopf gestellt (**3 Punkte**)

Gegeben sei ein binäres Bild mit den zwei Farben Schwarz und Weiß. Die Bildelemente (Pixel) seien dabei in einem zweidimensionalen Array vom Typ *int[][]* als Zahlen 1 für Weiß und 0 für Schwarz abgespeichert. Ziel der Aufgabe ist es, das Bild auf den Kopf zu stellen und die Werte zu invertieren.

Auf den Kopf stellen heißt dabei, dass die erste Zeile im Originalbild zur letzten Zeile im neuen Bild wird, die zweite Zeile zur vorletzten Zeile im neuen Bild usw. Zudem sollen die Werte invertiert werden, d.h. die Farbe Weiß wird zu Schwarz und Schwarz wird zu Weiß.

Implementieren Sie eine Klasse *Bild*. Diese enthält ein zweidimensionales Array als privaten Parameter und hat eine Methode *Bild flipAndInvert()*, in der die geforderte Funktion umgesetzt ist. Zusätzlich hat die Klasse eine Methode *void ausgabe()*, mit der ein Bild auf der Konsole angezeigt wird.

Beispiel:

Originalbild:		Bild nach <i>flipAndInvert()</i> :
1 0 0		1 0 1
0 1 0		1 1 0
0 0 1		1 0 1
0 1 0		0 1 1

Hinweise:

- Dokumentieren Sie Ihren Entwurf der Klasse in einem Klassendiagramm. Nutzen Sie dazu die Notation aus der Vorlesung.
- Es sollen nur Bilder mit den Werten 0 und 1 berücksichtigt werden.
- Testen Sie Ihre Methode mit den unten gezeigten Bildern in der *main*-Methode einer eigenen Testklasse. Nutzen Sie für die Ausgabe der Ergebnisse die Methode *ausgabe*.

Testbild 1:	Testbild 2:
1 1 0 1	0 0 1 1 0
0 1 0 1	0 1 0 1 1
0 1 1 1	1 0 0 1 1
1 0 1 0	0 1 0 1 0