

Einstufungstest K3 WS2021 14. Oktober 2021

Einschränkungen

Sie dürfen keine zusätzlichen Hilfsmethoden oder globalen Variablen verwenden.

Die vorgegebenen Methodenköpfe dürfen nicht erweitert oder geändert werden.

Für die Implementierung der rekursiven Methode dürfen keine Schleifen verwendet werden.

Sie dürfen Strings nicht per Referenz vergleichen.

Sie dürfen nicht die Methoden `Klone` und `System.arraycopy` verwenden

Verwenden Sie nicht folgende Klassen `StringBuilder`, `StringBuffer` oder `Objects`

Sie dürfen nur folgende(n) Methode(n) aus der Klasse `Arrays` verwenden: `deepToString`, `toString`

Sie dürfen nur folgende Methode(n) aus der Klasse `String` verwenden: `charAt`, `length`, `substring`, `isEmpty`

Deklarieren und initialisieren Sie in `main` folgende(n) Variable(n):

```
int[] test1 = {1, -5, 0, 6};
int[][] test2 = {{1},{2},{3}};
int[][] test3 = {{1, 0, 1}, {}, {5, 5, 6, 5}, {}};
int[][] test4 = {{7, 7, 5, 7, 3}, {2, -4}, {10, 20, 30}, {-1, -2, -3, -4, -5, -6, -7}};
int[][] test5 = {{}, {}}, test6 = {{}};
```

Implementieren Sie folgende Methoden:

- **`int[][] generate(int[] inputArray)`** erzeugt aus `inputArray` ein neues zweidimensionales Array und retourniert dieses. Die erste Zeile mit Index 0 ist mit `inputArray` identisch, d.h. es gilt `(inputArray == generate(inputArray)[0]) == true`. Die zweite Zeile enthält die gleichen Einträge wie `inputArray`, jedoch gilt `(inputArray == generate(inputArray)[1]) == false`. Alle weiteren Zeilen enthalten ebenso die gleichen Einträge, werden jedoch mit jeder Zeile um einen Eintrag kürzer, bis zur letzten Zeile, die genau einen Eintrag hat.

Vorbedingung(en): `inputArray != null` und `inputArray.length > 0`.

Wird die Methode z.B. mit `test1` aufgerufen, entsteht folgendes Array (achten Sie auf obige Bedingungen):

1	-5	0	6
1	-5	0	6
1	-5	0	
1	-5		
1			

- **`void change(int[][] inputArray)`** modifiziert `inputArray` wie folgt: die erste Zeile wird zur letzten Spalte, die zweite Zeile zur vorletzten Spalte, usw. (Siehe Beispiel). Die Anzahl der Zeilen des Arrays bleibt unverändert. Spalten, die nach der Modifikation länger werden als die Anzahl der Zeilen in `inputArray`, werden abgeschnitten. Wenn sie kürzer sind, werden sie nach dem letzten Eintrag mit dem Wert 0 gefüllt.

Hinweis: Nach dem Aufruf von `change` hat `inputArray` so viele Spalten wie Zeilen (d.h. es ist quadratisch).

Vorbedingung(en): `inputArray != null` und `inputArray[i] != null` für alle gültigen `i`.

Wird die Methode z.B. mit `inputArray=test4` aufgerufen, entsteht folgendes Array:

-1	10	2	7
-2	20	-4	7
-3	30	0	5
-4	0	0	7

- **Boolean `isEqual(String s1, String s2)`** gibt `true` zurück, falls gilt `s1.equals(s2) == true`, andernfalls `false`. Sind beide Parameter `s1` und `s2` gleich `null`, wird `true` zurückgeliefert, wenn genau ein Parameter `null` ist, wird `false` zurückgeliefert.

Diese Methode muss rekursiv implementiert werden.

Deklarieren Sie auch neue Arrays, die für die Tests benötigt werden.

Testen Sie alle Methoden und deren Seiteneffekte in `main` mit zumindest folgenden Aufrufen und weiteren Aufrufen (z.B. mit `deepToString`) für die Ausgaben. Erzeugen Sie diese Ausgaben nur in `main`, nicht in den implementierten Methoden.

<code>int[][] result1 = generate(test1)</code>	<code>[[1, -5, 0, 6], [1, -5, 0, 6], [1, -5, 0], [1, -5], [1]]</code>
<code>result1[0] == test1</code>	<code>true</code>
<code>result1[0] == result1[1]</code>	<code>false</code>
<code>result1[0] == result1[2]</code>	<code>false</code>
<code>result1[1] == result1[2]</code>	<code>false</code>
<code>change(test2)</code>	<code>[[3, 2, 1], [0, 0, 0], [0, 0, 0]]</code>
<code>change(test3)</code>	<code>[[0, 5, 0, 1], [0, 5, 0, 0], [0, 6, 0, 1], [0, 5, 0, 0]]</code>
<code>change(test4)</code>	<code>[[-1, 10, 2, 7], [-2, 20, -4, 7], [-3, 30, 0, 5], [-4, 0, 0, 7]]</code>
<code>change(test5)</code>	<code>[[0, 0], [0, 0]]</code>
<code>change(test6)</code>	<code>[[0]]</code>
<code>isEqual("blah", "blahlblahl")</code>	<code>false</code>
<code>isEqual("blah", "blah")</code>	<code>true</code>
<code>isEqual("", "")</code>	<code>true</code>
<code>isEqual(null, null)</code>	<code>true</code>