

Artemis 6.4.0

Course Overview

ge64baw

Courses > Praktikum: Grundlagen der Programmierung WS22/23 > Exercises > W03H01 - Array Funktionen

W03H01 - Array Funktionen

Hausaufgabe

Easy

Submission due:

9 months ago

Points: 8 of 8

Assessment: automatic

Complaint due:

9 months ago

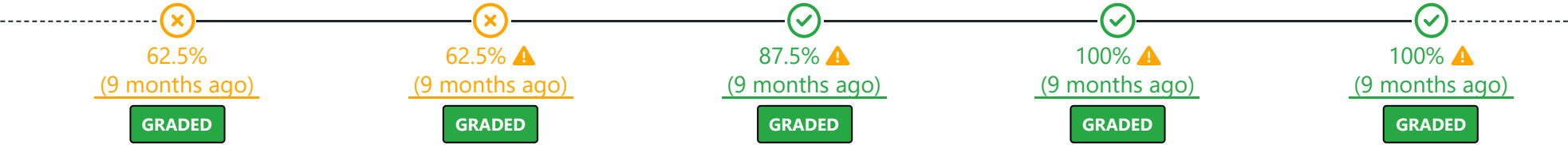
Resume practice in exercise

100%

(9 months ago)

GRADED

Recent results:



Show all results

Tasks:

W03H01 - Array Fun(ktionen)

Die Pinguine Happy und Snappy haben von deiner tollen Leistung in der W03P01 und W03P02 gehört und wollen unbedingt, dass du ihnen hilfst, noch weitere Funktionen zu schreiben.

Implementiere daher die folgenden statischen Methoden in `ArrayFunctions`.

?

Sichere Summe

No results

Die Methode `public static long sumOfSquares(int[] array)` berechnet die Summe der Quadrate der einzelnen Einträge des Arrays `array`, also in anderen Worten das Quadrat der Länge des Vektors `array`. Sie soll den korrekten Wert dieser Summe zurückgeben, insofern dieser noch in einen `long` passt. Wenn dies nicht der Fall ist, soll der `String` `"Overflow!"` auf der Konsole ausgegeben und der Fehlerwert `-1` zurückgegeben werden. Achte also auf Overflows, sowohl von Integern als auch von Longs.

Die Summe des leeren Arrays ist für diese Aufgabe als 0 definiert.

Hinweis: Integer können Werte zwischen jeweils einschließlich -2^{31} und $2^{31} - 1$, Longs Werte zwischen jeweils einschließlich -2^{63} und $2^{63} - 1$ darstellen.

?

Reißverschluss

No results

Die Methode `zip()` hat die Signatur `int[] zip(int[] a, int[] b)`. Der Rückgabewert ist ein Array mit allen Werten aus `a` und `b`. Dabei wechseln sich Werte aus `a` und `b` immer ab, beginnend mit `a`. Mit `a=[1,3]` und `b=[2,4]` ist also das Ergebnis `[1,2,3,4]`. Wenn die Arrays unterschiedliche Länge haben, werden die fehlenden Zahlen des schon vollständig übernommenen Arrays übersprungen und nur noch Zahlen aus dem längeren Array angehängt.

?

Reißverschlüsse

No results

Die Methode `zipMany()` verhält sich ähnlich zu `zip()`, nur hat es n verschiedene Arrays als Eingabe. Die Signatur ist folglich `int[] zipMany(int[][] arrays)`. Auch hier sollen die Zahlen aus den Eingabe-Arrays der Reihe nach abwechselnd in das Ergebnis-Arrays eingefügt werden. Ebenso sollen bereits vollständig bearbeitete Arrays dabei übersprungen werden, bis alle Werte der Eingabe-Arrays verbraucht wurden. Wenn das Eingabe-Array leer ist, wird ein leeres Array zurückgegeben.

Beispiel: mit `arrays=[[1,4],[2,5],[3,6]]` ist das Ergebnis `[1,2,3,4,5,6]`.

?

Filtern

No results

Die Methode `public static int[] filter(int[] array, int min, int max)` entfernt Zahlen aus dem Array, die nicht innerhalb eines angegeben Intervalls sind. Das Intervall wird durch 2 integer angegeben, die jeweils den größten und kleinsten erlaubten Wert inklusiv darstellen. Das zurückgegebene Array ist, wenn alle Zahlen außerhalb des Intervalls herausgefiltert wurden, entsprechend kleiner, die Reihenfolge der erlaubten Zahlen bleibt aber gleich. Wenn der größte erlaubte Wert kleiner als der kleinste erlaubte Wert ist, kann kein Wert dieses Kriterium erfüllen und das Ergebnis ist ein leeres Array.

?

Rotieren

No results

Die Methode `public static void rotate(int[] array, int amount)` soll die Einträge von `array` um `amount` nach hinten rotieren. D.h. alle Einträge sollen um `amount` nach hinten verschoben werden, wobei das hinterste Element beim um 1 Rotieren wieder nach vorne an die erste Stelle kommt. Wenn man also z.B. das Array

[1, 2, 3, 4, 5]

um 2 rotiert, erhält man das Array

```
[4, 5, 1, 2, 3]
```

1, 2 und 3 wurden um 2 nach hinten bewegt, 4 und 5 wären aus den Bounds des Arrays gefallen, landen also wieder ganz vorne.

Rotation um eine negative Zahl entspricht dabei einer Rotation in die andere Richtung (nach links), Rotation um eine Zahl mit Betrag $\geq \text{array.length}$ rotiert es einfach mehrfach komplett. Wenn man z.B.

```
[1, 2, 3, 4, 5]
```

um -1 rotiert, erhält man das Array

```
[2, 3, 4, 5, 1]
```

Wenn man

```
[1, 2, 3, 4, 5]
```

um 6 rotiert, erhält man das Array

```
[5, 1, 2, 3, 4]
```

Für `amount` kann also jeder beliebige `int` übergeben werden, es kommt immer eine valide Rotation heraus.

? Zählen No results

Die Methode `public static int[][] quantities(int[] array)` soll die Anzahl an Vorkommen jeder im Array `array` überhaupt vorkommenden Zahl zählen. Dabei soll die Anzahl `#x` von Vorkommen der Zahl `x` durch das Array `{x, #x}` repräsentiert werden. Die Methode gibt nun ein Array von derartigen Arrays zurück. Für jede in `array` vorkommende Zahl soll ein solcher Eintrag erstellt werden und zwar in der Reihenfolge, in der die Zahlen in `array` jeweils zum ersten Mal vorkommen.

Hier ein Beispiel: Das Array

```
ex1 = [1, 1, 2, 1, 3, 2, 1]
```

enthält 4-mal die 1, 2-mal die 2 und 1-mal die 3, was durch die Arrays `{1, 4}`, `{2, 2}` und `{3, 1}` dargestellt wird. Das erste Vorkommen von der 1 steht vor dem ersten Vorkommen der 2, welches wiederum vor dem ersten Vorkommen der 3 steht. Der Rückgabewert von `quantities(ex1)` ist daher

```
[
  [1, 4],
  [2, 2],
  [3, 1]
]
```

Hier noch ein weiteres Beispiel: Für das Array

```
ex2 = [2, 0, 3, 0, 2, 5, 5, 3, 2, 0, 0, 0]
```

wäre der Rückgabewert von `quantities(ex2)`

```
[
  [2, 3],
  [0, 5],
  [3, 2],
  [5, 2]
]
```

Tests

Du bekommst bei dieser Aufgabe von den PublicTests (den Tests, die direkt nach jedem Push ausgeführt werden), zwar kein Feedback darüber, was du falsch gemacht hast, allerdings teilen sie dir nach jedem Push mit, ob du die jeweilige Teilaufgabe korrekt gelöst hast oder nicht. Daher kannst du dir auch sicher sein, dass du alle Punkte bekommst, wenn du alle PublicTests bestehst. Die HiddenTests sind deckungsgleich mit den PublicTests und geben

dir nach der Deadline ein Feedback, warum sie fehlgeschlagen sind.

[Lösungsvorschlag \(nach der Deadline\)](#)

Exercise details

Release date:	Nov 3, 2022 18:30
Submission due:	Nov 13, 2022 18:00
Complaint due:	Nov 20, 2022 18:00

Every student is allowed to complain once per exercise. In total 1000 complaints are possible in this course. You still have **998** complaints left. 