
Übung Powershell

Array Erweiterung

Lernziele:

- Umgang mit Arrays
- Zweidimensionale Arrays

1 Befüllen von zwei Arrays

Befüllen Sie **zwei Arrays gleichzeitig** in einer Schleife. Beide Arrays haben Länge 10. Das eine Array wird mit 1, 2, 3, ... befüllt, das zweite Array mit -10, -9, -8,

2 Arrays filtern

Gegeben sei Array a mit ganzen positiven und negativen Zahlen. Geben Sie alle negativen Elemente auf der Konsole aus.

3 Array rückwärts

Gegeben ist ein Array:

```
int[] foo = { 1, 2, 3, 4, 5, 6, 7 };
```

Erstellen Sie einen neuen Array bar, wo die Elemente in umgekehrter Reihenfolge abgelegt sind. Wenn Sie bar ausgeben, sehen Sie also:

```
[0] 7  
[1] 6  
[2] 5  
[3] 4  
....
```

4 Elemente verschieben

Gegeben ist ein Array:

```
int[] foo = { 10, 20, 30, 40, 50 };
```

Verschieben Sie alle Elemente ab Index 1 um eine Stelle nach rechts. Als Ergebnis sollte hinter das folgende in foo stehen:

```
[0] 10  
[1] 10  
[2] 20  
[3] 30  
[4] 40
```

5 Boolsche Operationen

Abstrakt geht es um die Frage, wie Sie bei einer Menge von N Wahrheitswerten herausfinden, ob mindestens ein Wert **true** ist (entspricht ODER) bzw. ob alle Werte **true** sind (entspricht UND). Die Werte liegen in einem Array vor.

Vorbereitung

Zur Vorbereitung der Aufgabe erzeugen Sie einen Array "infiziert" mit zehn Elementen. Jeder Eintrag soll mit einer Wahrscheinlichkeit von 5% true sein.

```
boolean[] infiziert = new boolean[10];  
  
for (int i = 0; i < infiziert.length; i++) {  
    // Ihr Code  
}  
  
println(infiziert); // Kontrolle
```

Tipp

Verwenden Sie `random(100)`. Sie erhalten einen zufälligen Wert zwischen 0 und 100. Wie können Sie diesen Wert nutzen, um nur in 5% aller Fälle eine Variable auf **true** zu setzen?

Teil 1

Die erste Aufgabe ist jetzt festzustellen, ob **mindestens eine Person** infiziert ist von den 10. Sie sollen dies mit Hilfe einer Schleife lösen, damit Ihre Lösung auch für sehr große Arrays funktioniert. Geben Sie also "mindestens eine/r infiziert" oder "keiner infiziert" aus als auch den kompletten Array aus, um zu sehen, ob Ihr Programm korrekt arbeitet.

Teil 2

Die zweite Aufgabe ist festzustellen, ob **alle** infiziert sind. Um dies zu testen, setzen Sie die Wahrscheinlichkeit, dass eine/r infiziert ist, auf 95% (siehe Vorbereitung oben).

Geben Sie hier "alle infiziert" oder "nicht alle infiziert" aus.

6 Personen filtern

Sie haben eine Anzahl Personen in drei Arrays repräsentiert:

```
String[] namen = {  
    "Max Schmidt", "Lisa Marx", "Robin Meier",  
    "Lara Huber", "Anna Groß", "Heribert Lehmann",  
    "Harry Potter", "Obi-wan Kenobi", "Julia Kron"  
};  
int[] alter = {  
    33, 28, 12,  
    22, 23, 64,  
    16, 102, 18  
};  
boolean[] weiblich = {  
    false, true, false,  
    true, true, false,  
    false, false, true  
};
```

Listen Sie mit Hilfe einer Schleife nur die Frauen auf und zwar wie folgt:

```
Frau Lisa Marx  
Frau Lara Huber  
Frau Anna Groß  
Frau Julia Kron
```

Jetzt sollen Sie zählen! Verwenden Sie eine Schleife und (mehrere) Zählervariablen, um folgende Personengruppen zu zählen:

- Alle Frauen
- Alle Personen unter 40
- Alle Männer über 30

Geben Sie die drei Zahlen auf der Konsole aus.

7 Array auseinander nehmen

Gegeben ist folgender Array:

```
int[][] a = {  
    {4, 0, 2},  
    {3, 2, 3},  
    {1, 2, 3}  
};
```

Erzeugen Sie **drei einfache Arrays** b, c und d, welche jeweils eine Zeile von a enthalten. Array b enthält also { 4, 0, 2 }. Sie sollen das natürlich so programmieren, dass die Inhalte aus a genommen werden.

Testen Sie Ihr Programm mit:

```
println(b);  
println();  
println(c);  
println();  
println(d);
```

Sie sollten sehen:

```
[0] 4  
[1] 0  
[2] 2
```

```
[0] 3  
[1] 2  
[2] 3
```

```
[0] 1  
[1] 2  
[2] 3
```

8 Array befüllen

Gegeben ist ein leeres Array:

```
int[][] a = new int[3][5];
```

Schreiben Sie zunächst den Array mit Hilfe von Schleifen auf die Konsole:

```
0 0 0 0
0 0 0 0
0 0 0 0
```

Versuchen Sie als nächstes, den Array wie folgt zu befüllen. Versuchen Sie, das möglichst ökonomisch zu tun.

```
1 0 0 0
0 1 0 0
0 0 1 0
```

Tipp

Sie benötigen nur eine Schleife (und nicht zwei verschachtelte).

Versuchen Sie als nächstes, den Array wie folgt zu befüllen:

```
0 1 2 3 4
0 1 2 3 4
0 1 2 3 4
```

Tipp

Hier benötigen Sie zwei verschachtelte Schleifen.

Anschließend probieren Sie folgende Befüllung:

```
0 0 0 0
1 1 1 1
2 2 2 2
```

Zu guter letzt probieren Sie das hier:

```
0 1 2 3 4
1 2 3 4 5
2 3 4 5 6
```

9 Artikel

Sie schreiben für einen kleinen Laden ein Verwaltungsprogramm. Die Artikel in dem Laden stehen in folgender Tabelle:

```
int[][] tabelle = {  
    {1001, 5, 500, 0},  
    {2001, 23, 1200, 0},  
    {3001, 2, 85, 0}  
};
```

In der ersten Spalte steht die Artikelnummer, in der zweiten Spalte die Anzahl der verfügbaren Artikel und in der dritten Spalte der Stückpreis. In der vierten Spalte soll der Gesamtpreis stehen (tut er aber derzeit nicht).

Sie haben zwei Aufgaben:

1. Erzeugen Sie folgenden Ausgabe:

```
5x Artikel Nr. 1001 à 500 EUR = 0  
23x Artikel Nr. 2001 à 1200 EUR = 0  
2x Artikel Nr. 3001 à 85 EUR = 0
```

2. Berechnen Sie dann für jeden Artikel den Gesamtwert aller Artikel dieses Typs und schreiben Sie ihn in die Tabelle zurück (vierte Spalte). Bei Ausgabe sollten Sie sehen:

```
5x Artikel Nr. 1001 à 500 EUR = 2500  
23x Artikel Nr. 2001 à 1200 EUR = 27600  
2x Artikel Nr. 3001 à 85 EUR = 170
```

Aufgaben wurden entnommen von:

Quellenangabe:

<https://michaelkipp.de/processing/08%20arrays.html>

Die Aufgaben wurden für das Programmieren in Java entwickelt, eignen sich aber auch sehr gut für Powershell!

Historie

Dokument erstellt

R. Müller

07.12.2020