## Einfacher Umgang mit Arrays

### Propädeutik (sehr einfach, nur für Anfänger!)

**Aufgabe:**

Unten sehen Sie ein kleines Programm, in dem Sie einem Array mit dem Namen „zahlen” sechs Integer Werte zuweisen.

Ändern Sie das Programm so ab, dass aus dem Array „zahlen” ein String Arrays mit der Bezeichnung „termine“ wird, das in der Lage ist vier Werte aufzunehmen und weisen Sie vier Elemente (vom Typ String) zu (beispielsweise Aufstehen, Frühstück, Mittag und Abendessen). Das Programm soll „Termine1“ heißen.

Geben Sie anschließend alle vier Werte aus.

**public** **class** Zahlen1 {

**public** **static** **void** main(String[] args) {

int zahlen[] = new int[6];

zahlen[0] = 4711;

zahlen[1] = 815;

zahlen[2] = 190;

zahlen[3] = 7;

zahlen[4] = 123;

zahlen[5] = 998877;

for (int i = 0; i < 6; i++)

System.out.println("Die Zahlen lauten: " + zahlen[i]);}}

### Array mit Zufallzahlen füllen und ausgeben

**Ziel:**

Schreiben Sie ein kleines Programm "Wuerfel", dass ein Integer Array mit der Bezeichnung "wurf" anlegt, welches in der Lage ist, drei gewürfelte Zufallszahlen zu speichern.

Die Zahlen sollen durch eine Zufallsfunktion ermittelt werden und innerhalb einer FOR-Schleife zugewiesen werden. Auch sollen die gewürfelten Zahlen ausgegeben werden.

**Hinweise:**

Die Zahlen sollen dem Feld zufällig mittels der Funktion Math.random() zugewiesen werden. Diese Funktion liefert eine Pseudo-Zufallszahl x aus dem Bereich 0,0 < x < 1,0 also Werte vom Typ double (Lösungshinweis: Explizite Typumwandlung). Um beispielsweise Zahlen zwischen 1 und 100 zu bekommen müssten Sie schreiben: (Math.random() \* 100 + 1);.

**Ausgabe/Darstellung:**

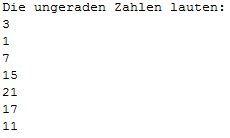
****

**Ziel:**

Sie haben in dem Programm „UZahlen“ ein Array „x“ initialisiert (bereits im Programmcode vorgegeben). Nun wollen Sie die ungeraden Zahlen mittels for-Schleife berechnen und ausgeben lassen. Hinweis: Modulo Funktion!

Vervollständigen Sie das Programm UZahlen, so dass die folgende Anzeige erscheint:

**Ausgabe/Darstellung:**



**Programmcode Teilnehmer:**

**public** **class** UZahlen {

**public** **static** **void** main(String[] args) {

**int** x[] = { 3, 1, 7, 15, 12, 21, 20, 17, 8, 11, 14 };

System.***out***.println("Die ungeraden Zahlen lauten: ");

**int** j = 0;

### Die Methode Arrays.toString

In der Regel werden Arrays mit der Hilfe von [Schleifen](https://javabeginners.de/Schleifen_und_Verzweigungen/index.php) ausgegeben. Wollen Sie sich jedoch auf die Schnelle einen Überblick über die Inhalte eines Arrays ausgeben lassen, so können Sie dies mit Hilfe der statischen Methode **Arrays.toString()** erreichen, da sie Arrays in Strings umwandelt.

**Ziel:**

Deklarieren und initialisieren Sie in dem Programm „ArrayAusgabe01“ ein String Array "namen" mit den folgenden drei Elementen: { "Joachim K.", "Marita E.", "Michael D."}. Lassen Sie sich dieses Array ohne FOR-Schleife als String ausgeben.

Überprüfen Sie auch, was passiert, wenn Sie die Arrays.toString Methode weglassen.

**Ausgabe/Darstellung:**



**Ziel:**

Gegeben sei folgendes Programm:

**public** **class** ArrayAusgabe02 {

**public** **static** **void** main(String[] args) {

**int**[] array = { 77, 19, 23 };

**for** (**int** i = 0; i < array.length; ++i) {

System.***out***.print(array[i] + " ");

}

}

}

Ändern Sie das Programm so ab, dass die Ausgabe in einer Anweisung und ohne for-Schleife erfolgt.

**Ziel:**

Legen Sie in einem Programm „ByteArray“ ein Byte-Array „b“ an und initialisieren Sie es mit folgenden Werten:

69, 73, 78, 32, 84, 69, 83, 84

1. Geben Sie das Array b ohne FOR oder WHILE Schleife in einer Anweisung (als Zahlen) aus.

**Ausgabe/Darstellung:**



Wie lautet die Anweisung?

1. Erzeugen Sie danach mittels „new“-Operator ein String Objekt „t“, dem Sie das Array „b“ übergeben. Testen Sie, was passiert, wenn Sie dieses Array als String ausgeben.

**Ausgabe/Darstellung:**



## Arrays sortieren

### Verwendung der Methode sort()

Möchte man nicht selber einen Sortieralgorithmus schreiben um ein Array zu sortieren, muss zunächst die Klasse java.util.Arrays importiert werden.

Die Methode sort(Object[] ist statisch und muss daher mit dem Klassennamen (also z.B. Arrays.*sort*(namen);) aufgerufen werden. Sie besitzt keinen Rückgabewert, sondern sortiert lediglich das übergebene Array (im Beispiel „namen“).

Die Methode Arrays.sort() ist überladen und kann unterschiedliche Parameter, darunter natürlich auch primitive Datentypen übernehmen.

**Ziel:**

Legen Sie in einem Programm „ArraySortiert“ ein String Array "namen" mit folgenden Elementen: { "Martin", "Susanne", "Cedrik", "Kevin", "Anton", "Doris" } an.

Geben Sie das Array zunächst unsortiert aus und benutzen Sie dazu wieder die Methode Arrays.toString().

Sortieren Sie anschließend das Array mittels der sort()-Methode der Klasse Arrays und lassen Sie sich das sortierte Array anzeigen

**Ausgabe/Darstellung:**

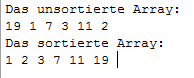


**Ziel:**

In dem Programm „ArraySortiert02“ ist ein Array „a3“ deklariert. Ihre Aufgabe besteht darin, das Array einmal unsortiert und einmal sortiert auszugeben.

Die Ausgabe Ihres Programms muss auch dann noch ohne Änderung des Codes funktionieren, wenn man in das Array „a3“ zwei weitere „Testwerte“ einfügt (Hinweis: FOR-Schleife).

**Ausgabe/Darstellung:**

****

**Programmcode Teilnehmer:**

**import** java.util.Arrays;

**public** **class** ArraySortiert02{

**public** **static** **void** main(String[] args) {

**int**[] a3 = { 19, 1, 7, 3, 11, 2 };

...