

# Array – Spielereien

Vorwort: Folgende Übungen können alle in einer Main-Methode geschrieben werden. Das Programm muss nicht „hübsch“ sein. Es wird unter anderem einige Redundanzen geben, die wir in diesem Programm aber ohne weiteres akzeptieren. Ein Struktogramm ist nicht notwendig. Es ist ebenfalls okay, wenn Sie `System.out.println()` verwenden um „eine Zeile nach unten“ in der Konsole zu gehen.

Arbeiten Sie einfach Anweisung für Anweisung im Editor nach, um ein Grundgefühl für Arrays und deren Methoden zu bekommen. Für diese kleine Übung arbeiten wir in diesem Fall mit der Konsolenausgabe. Die Konsolenausgabe soll allerdings „beschreibend“ sein. Packen Sie nicht nur die nackten Werte in die Ausgabe!

Benutzen Sie die Java For Each (oder enhanced For – Loop) bis Aufgabe 12 nicht!

1. Deklarieren und Initialisieren Sie ein Array **arr1** mit **5 int-Elementen**. Verwenden Sie nicht die Array-Literal Initialisierung. Denken Sie sich Elemente aus, benutzen Sie nicht die `Arrays.fill` Methode.
2. Greifen Sie auf Element 3 und Element 5 von **arr1** zu und geben Sie beide Elemente mit einer Konsolenausgabe aus.
3. Geben Sie Element 4 von **arr1** aus. Verändern Sie nun Element 4 in eine neue Zahl. Geben Sie Element 4 erneut nach der Änderung aus.
4. Deklarieren und initialisieren Sie ein Array-Literal mit der Variable **arr2**. Das Array-Literal soll Integer Werte enthalten. Denken Sie sich die 5 Elemente aus-
5. Geben Sie **arr2** mit einer For-Schleife auf der Konsole aus.
6. Überschreiben Sie jedes Element von **arr2** mit einer 0. Geben Sie das **arr2** nach dem Überschreiben erneut aus, um Ihr Ergebnis zu prüfen. Nutzen Sie zum Überschreiben ebenfalls eine for-Methode! Geben Sie das Array wie bei Aufgabe 5 aus.
7. Deklarieren und initialisieren Sie ein-Array mit der Variable **arr3**. Die Array-Variable soll 5 Integer Elemente speichern können. Benutzen Sie kein Array-Literal! Füllen Sie das Array mit der **fill-Methode** der Array Klasse. Mit welchen Zahlen Sie das Array füllen ist Ihnen überlassen. Geben Sie das Array danach wie bekannt aus.
8. Erzeugen Sie ein neues, unsortiertes Array-Literal mit der Variable **arr4**. Sortieren Sie es mit der **sort-Methode** der Klasse `Arrays`. Array ausgeben und Ergebnis prüfen!
9. Erzeugen Sie ein neues String Array-Literal mit der Variable **arr5** und Suchen Sie mit der **binarySearch-Methode** ein Element Ihrer Wahl. Achtung! Das Array muss vorher sortiert worden sein! Nutzen Sie hierfür die `sort-Methode`. Speichern Sie den Rückgabewert. Prüfen Sie mit einer Bedingung, ob der Wert gefunden wurde oder nicht. *Die `binarySearch-Methode` gibt ein negativen Rückgabewert, sofern das Element nicht gefunden wurde. Ansonsten liefert die Methode den Index des gefundenen Elements.*

10. Erzeugen Sie das gleiche Array (mit den gleichen Elementen an gleicher Stelle) wie in Aufgabe 9 nur mit der Variable **arr6**. Prüfen Sie beide Arrays auf Gleichheit mit der **equals-Methode** der Arrays Klasse. Der Rückgabewert der equals-Methode ist ein Boolean. Prüfen Sie das Ergebnis mit einer if-Bedingung auf Wahrheit und geben Sie das Ergebnis auf der Konsole aus.
- Kommt bei Ihnen true heraus? Falls nicht, sortieren Sie Ihre Liste vorher und prüfen das Ergebnis erneut. Geben Sie das Ergebnis wieder auf der Konsole aus.
11. Last but not least: Geben Sie arr1 und arr5 beide separat als Java **For-Each bzw. Enhanced For Loop** aus.