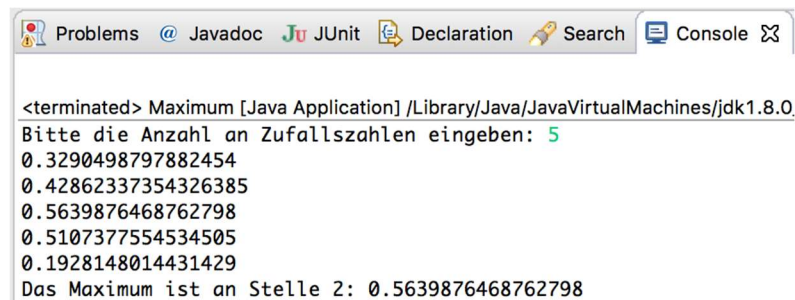


## Übungsblatt 4 – Methoden, Felder und Kontrollstrukturen

### Aufgabe 1 – Finde das Maximum

Erstellen Sie eine Klasse `Maximum` und fügen Sie der Klasse eine Methode `max` hinzu, welche als Argument ein `double`-Array entgegennimmt, und den Array-Index zurückgibt, an dem die größte Zahl steht.

Schreiben Sie auch eine `main` Methode, um Ihre `max` Methode zu testen. Lesen Sie dazu von der Konsole ein, wie viele Zufallszahlen generiert werden sollen, und legen Sie entsprechend ein Array mit Zufallszahlen an. Verwenden Sie dann die `max` Methode, um sowohl den Index als auch den Wert des Maximums zu finden.



```
<terminated> Maximum [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0.  
Bitte die Anzahl an Zufallszahlen eingeben: 5  
0.3290498797882454  
0.42862337354326385  
0.5639876468762798  
0.5107377554534505  
0.1928148014431429  
Das Maximum ist an Stelle 2: 0.5639876468762798
```

#### Hinweise:

- Sie benötigen eine `main()` und eine `max()` Methode
- `main()`:
  - Scanner zum Einlesen der Array Größe
  - Array erzeugen in der vom Nutzer eingegebenen Größe
  - Befüllen des Arrays mit Zufallswerten mittels `Math.random()`
  - Aufruf von `max()` wobei das Array übergeben wird
  - Ausgabe des Arrays und des Index, an dem sich der Max-Wert befindet.
- `max()`:
  - nehmen Sie an dass der erste Array Wert an Index 0 Ihr Maximum ist und speichern Sie die Werte zwischen
  - Mittels einer For-Schleife iterieren Sie über das Array und vergleichen jede Array Position mit dem zwischengespeicherten Wert.
  - Finden Sie einen höheren Wert, speichern Sie diesen und den Index in den von Ihnen definierten Variablen zwischen und verwenden Sie den neuen Max Wert in der nächsten Schleifen Iteration.
  - `Max()` liefert den Index als Rückgabeparameter
  - Verwenden Sie die Methode `Math.random ( )` um eine Zufallszahl zu generieren.

## Aufgabe 2 - Schnittmenge

Erstellen Sie eine Klasse `Schnittmenge` und fügen Sie ihr eine Methode `schnittmenge` hinzu, welche als Argumente zwei `int`-Arrays entgegennimmt, und ein `int`-Array zurückgibt, welches die Zahlen enthält, welche in beiden Arrays vorkommen (Mengenlehre: Schnittmenge).

Schreiben Sie weiterhin eine `main` Methode, in der Sie die `schnittmenge` Methode mit einigen (festgelegten) Beispielen testen:

- `{0, 1, 2, 3, 4, 5}` und `{3, 4, 5, 6, 7, 8}`  $\rightarrow$  `{3, 4, 5}` (`set1a`, `set1b`)
- `{0, 1, 2, 3}` und `{4, 5, 6}`  $\rightarrow$  `{}` (`set2a`, `set2b`)
- `{0, 1, 2}`, `{0, 1, 2}`  $\rightarrow$  `{0, 1, 2}` (`set3a`, `set3b`)

### Hinweise:

- Sie benötigen eine `main()` und eine `max()` Methode
- `main()`:
  - In dieser Aufgabe brauchen Sie keine Werte von der Tastatur einzulesen, ein „statischer“ Test genügt.
  - Sie erzeugen sich 6 Arrays wie oben angegeben.
  - Sie rufen hier die Methode `schnittmenge()` auf und übergeben die Arrays, welche Sie auf eine Schnittmenge prüfen wollen.
  - Für die Ausgabe können Sie folgende printouts verwenden:

```
System.out.println(Arrays.toString(schnittmenge(set1a, set1b)));  
System.out.println(Arrays.toString(schnittmenge(set2a, set2b)));  
System.out.println(Arrays.toString(schnittmenge(set3a, set3b)));
```

- `schnittmenge()`:
  - Da die Größe von Arrays nach Erstellung unveränderlich ist, müssen Sie die Schnittmenge zweimal bilden: Einmal um die Größe zu finden (und das Rückgabearray anzulegen), und einmal um die Werte einzufügen.
  - 1. For-Schleife: finde die Anzahl an Schnittmengenelemente
  - Erzeuge das Array in der Größe, die in der 1.Schleife ermittelt wurde
  - 2. For-Schleife: befülle das erzeugte Array mit den Schnittmengenelementen.

## Aufgabe 3 - Wechselgeld

Erstellen Sie eine neue Klasse `Wechselgeld` und fügen Sie der Klasse eine Methode `wechselgeld` hinzu, welche als Argumente einen Preis (oder Sollwert) sowie den vom Kunden bezahlten Betrag entgegennimmt, und auf der Konsole ausgibt welches Wechselgeld (100, 50, 20, 10, 50 Euro-Scheine, 2, 1, 0.50, 0.20, 0.10, 0.05, 0.02, 0.01 Euro-Münzen) der Kunde erhält. Der Rückgabewert der Methode soll `false` sein, wenn der Kunde zu wenig bezahlt hat, ansonsten `true`.

Schreiben Sie auch eine `main` Methode, um Ihre `wechselgeld` Methode zu testen. Lesen Sie dazu sowohl den Preis als auch den gezahlten Betrag von der Tastatur ein, und geben Sie auf der Konsole aus, ob genug bezahlt wurde („Vielen Dank“) oder nicht („Das reicht leider nicht“).



```
Problems @ Javadoc JUnit Declaration Search Console
<terminated> Wechselgeld [Java Application] /Library/Java/JavaVirtualMachines/jdk1.
Bitte Preis eingeben: 2.60
Bitte Betrag eingeben: 3.00
Wechselgeld:
2x 20ct
Vielen Dank.
```

## Hinweise:

- Sie benötigen eine `main()` und eine `wechselgeld()` Methode
- `main()` :
  - Scanner, um vom Nutzer den Preis und den Betrag einzulesen; Achtung: Nutzer gibt double ein, es ist hilfreich dies als CENT Beträge abzuspeichern und als `int` in die Methode `wechselgeld()` zu übergeben.
  - Aufruf der Funktion `wechselgeld(preis, betrag)`
  - Printout wie oben gegeben
    - Verwenden Sie den Rückgabewert der `wechselgeld` Methode.
- `wechselgeld()` :
  - Da es „höchstens“ um Centbeträge geht, empfiehlt es sich die Wechselgeldstückelung in ganzen Cent (`int`) zu berechnen.
  - Berechnung des Rückgabewertes
  - For-Schleife über das Centbeträge Array, um zu ermitteln, was das Rückgabegeld ist.