

## Wiederholung

---

- Es darf genau dann mehrere Methoden mit gleichem Namen geben, wenn sie alle eine unterschiedliche Signatur haben
- In diesem Fall spricht man vom **Überladen** einer Methode bzw. einer **überladenen** Methode

# Eindimensionale Arrays

---

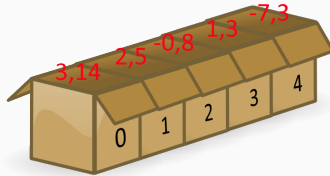
## Aufgabe 1

- Schreibe eine Methode `berechneMittelwert()`, die **zwei** Parameter vom Typ `float` erwartet und den Mittelwert (Durchschnitt) dieser Werte berechnet und zurückgibt.
- Schreibe eine Methode `berechneMittelwert()`, die **drei** Parameter vom Typ `float` erwartet den Mittelwert (Durchschnitt) dieser Werte berechnet und zurückgibt.

- Für wenige Werte funktioniert der Ansatz von Aufgabe 1 noch (halbwegs) gut
- Was tut man, wenn man den Mittelwert von vielen Werten (bspw. Klausurdurchschnitt) berechnen will?
- Problem:
  - Uns fehlt eine Möglichkeit, (beliebig) viele Werte zu speichern und mit ihnen zu arbeiten
- Lösung:
  - Arrays (manchmal auch Reihung genannt)

# Eindimensionale Arrays

- Ein eindimensionales Array ist ein zusammenhängender Speicherbereich für mehrere Werte des gleichen Typs
  - Die Werte sind durchnummeriert, beginnenden mit 0



**Abbildung 1:** Array mit 5 Elementen vom Typ **float**, Urheber: ks, Box-Clipart von [openclipart.org](https://openclipart.org/), User mcol

Definition:

```
float[] boxes = new float[5];
```

Zugriff auf Elemente:

```
boxes[0] = 3.14;
```

```
boxes[4] = -7.3;
```

Definition:

```
float[] boxes = new float[5];
```

Zugriff auf Elemente:

```
boxes[0] = 3.14;
```

```
boxes[4] = -7.3;
```

**float[] boxes** Definiert die Variable **boxes**, die ein Array vom Typ **float** bezeichnet.

**new float[5]** Erzeugt das eigentliche Array mit Platz für 5 Werte vom Typ **float**.

**boxes[0]** Zugriff auf das Element an Position 0 im Array **boxes**.



## Durchlaufen eines Arrays

- Oft will man ein Array komplett vom ersten bis zum letzten Element durchlaufen
  - Dies geht gut mittels einer **for**-Schleife
  - In `arrayname.length` ist die Anzahl der Elemente des Arrays gespeichert
- Beispiel:

```
float[] boxes = new float[500];  
for ( int i = 0; i < boxes.length; i++ ) {  
    System.out.println(boxes[i]);  
}
```

# Aufgabe 1

- Schreibe ein Programm, das ein Array mit 20 Elementen vom Typ `double` erzeugt und jedem Element den Wert `1.23` zuweist.
  - Lasse alle Elemente des Arrays anschließend ausgeben
- Verändere da Programm so, dass jedem Element des Arrays ein zufälliger Wert anstelle von `1.23` zugewiesen wird.
  - Verwende die Methode `random(float max)` zum Erzeugen von Zufallszahlen zwischen 0 (einschließlich) und `max` (ausschließlich)
- Verändere das Programm so, dass das Element an Position 0 den Wert `20.0` zugewiesen bekommt, das nächste den Wert `19.0` usw.

# Arrays als Parameter

Man kann Methoden schreiben, die ein Array als Parameter erwarten:

Definition:

```
void machWas(double[] werte) {  
    werte[0] = 1.234;  
}
```

Aufruf:

```
float[] meinArray = new float[20];  
// irgendwelche Daten ins Array schreiben  
// ...  
machWas(meinArray);
```

Die Methode `machWas()` arbeitet direkt auf dem übergebenen Array, nach der Ausführung hat also `meinArray[0]` den Wert 1.234!

## Aufgabe 2

- Schreibe ein Programm, das ein Array **noten** mit 13 Elementen vom Typ **int** erzeugt und jedem Element eine Zufallszahl zwischen 0 und 15 zuweist<sup>1</sup>.
  - **random(float max)** erzeugt eine Zufallszahl vom Typ **float**, die nicht direkt in einem Array vom Typ **int** gespeichert werden kann
  - Daher ist eine **Typumwandlung** (Typecast) notwendig:  
`noten[...] = (int)random(...)`
- Schreibe eine Methode **berechneMittelwert()**, der man ein Array als Parameter übergeben kann und die den Mittelwert aller Werte im Array berechnet und zurückgibt.

---

<sup>1</sup>Welche Verwendung könnte Herr K. für dieses Programm nur haben?

- Schreibe eine Methode `findeMaximum()`, die ein Array vom Typ `int` als Parameter erwartet und den größten im Array vorhandenen Wert ermittelt und zurückgibt.
- Teste diese Methode mit einem geeigneten Array von mindestens 20 Zufallszahlen.
  - Lasse sowohl das komplette Array als auch das gefundene Maximum ausgeben.