

## Referenzen auf Arrays

---

Beispielcode:

```
int[] daten = new int[25];  
for ( int i = 0; i < daten.length; i++ ) {  
    daten[i] = (int)(Math.random() * 10);  
}
```

Analyse:

- Die Variable **daten** ist eine Referenz auf das eigentliche Array
  - Die Referenz verweist auf den Anfang des Arrays
- Für's Leistungsfach:
  - Die Variable **daten** liegt auf dem Stack
  - Das eigentliche Array liegt auf dem Heap

- Wird ein Array als Parameter an eine Methode übergeben, so handelt es sich dabei um eine Referenz auf dieses Array
- Beispiel:

```
void setzeZufall(int[] a) {  
    for ( int i = 0; i < a.length; i++ ) {  
        a[i] = (int)(Math.random()*10);  
    }  
}  
  
void main() {  
    int[] werte = new int[20];  
    setzeZufall(werte);  
}
```

# Aufgabe 1

- Übertrage das Beispielprogramm von der vorherigen Folien in BlueJ
  - Erstelle hierzu zunächst ein neues Projekt und dann eine neue Klasse **Test**, in der beide Methoden implementiert werden
- Erweitere die Klasse um eine Methode zur Ausgabe eines Arrays von **int**-Werten
  - Das Array soll ebenfalls als Parameter übergeben werden
- Lasse innerhalb der Methode **main** das Array **werte** vor und nach dem Aufruf von **setzeZufall** ausgeben

- Es ist auch möglich, Methoden zu schreiben, die ein Array als Ergebnis zurück geben
  - Auch in diesem Fall wird eine Referenz auf das Array zurückgegeben

## Arrays als Rückgabewerte - Beispiel

```
public int[] erstelleArray(int groesse) {  
    int[] array = new int[groesse];  
    return array;  
}  
  
public void main() {  
    int[] meinArray = erstelleArray(10);  
    meinArray[5] = 23;  
}
```

Die Methode `erstelleArray` erzeugt ein neues Array und gibt eine Referenz darauf zurück.

`meinArray` ist eine Referenz auf ein Array mit 10 Elementen.

## Aufgabe 2

- Erweitere den Code von Aufgabe 1 um eine Methode **nachfolger**
  - Die Methode soll ein Array von **int**-Werten als Parameter übergeben bekommen
  - Zu jedem Element soll der Nachfolger bestimmt und in einem neuen Array gespeichert werden
  - Dieses neue Array soll zurückgegeben werden
  - Beispiel: Wird das Array {1,2,3} übergeben, soll das Array {2,3,4} zurückgegeben werden
- Teste auch diese Methode durch einen Aufruf in **main**
- Zusatzaufgabe:
  - Schreibe eine Methode **summe**, die ein Array mit einer geraden Anzahl von **int**-Werten als Parameter erwartet und die Summe von je zwei aufeinanderfolgenden Werten in einem neuen Array speichert und dieses zurück gibt
  - Beispiel: Wird das Array {1,2,3,4} übergeben, soll das Array {3,7} zurückgegeben werden

## Referenzen in Arrays

---



- Verwendet man den Namen einer Klasse als Typangabe eines Arrays, so können in diesem Array Objekte der Klasse gespeichert werden
  - Ein solches Array enthält Referenzen auf die jeweiligen Objekte
- Die Objekte selbst müssen separat erzeugt und den Arrayelementen zugewiesen werden
  - Wird nur das Array erzeugt, haben alle Elemente den Wert **null**

```
Ball[] b = new Ball[10];
```

Erzeugt ein Array mit 10 **Referenzen** auf Objekte vom Typ **Ball**.

```
for ( int i = 0; i < b.length; i++ ) {  
    b[i] = new Ball(10, 20, 5);  
}
```

Erzeugt die eigentlichen **Ball**-Objekte.

```
b[3] = b[8];
```

**b[3]** und **b[8]** veweisen nun auf dasselbe **Ball**-Objekt.

- Modifiziere Deine Implementierung von Pong so, dass statt einem Ball insgesamt 10 Bälle erzeugt werden
  - Verwende ein Array zur Speicherung der Bälle
  - Alle Bälle sollen sich bewegen und vom Rand abprallen