

Arrays

A dark blue diagonal gradient bar that starts from the bottom-left corner and extends towards the top-right corner, covering the lower half of the slide.

Arrays

-FATHER-



Een **array** is een **verzameling van elementen** van **hetzelfde datatype** waarbij ieder element voorzien is van een **index** die de **plaats** van het **element** in de verzameling aanduidt.

0	Element 1
1	Element 2
2	Element 3
3	Element 4
4	Element 5

Creatie Arrays

Een array is een **object**.

→ Wordt **gemaakt** met de **new operator**

◆ `DataType arrayname[] = new DataType[length];`

◆ `int[] numbers = new int [10];`

→ Arrays hebben een **vaste grootte** die na initialisatie niet meer gewijzigd kan worden.

◆ De grootte wordt at runtime bepaald: dit mag dus een variabele zijn in plaats van een literal

→ **Alle elementen** in een array moeten van **hetzelfde type** zijn

Array

Na het creëren van de array worden de waarden van de elementen **automatisch geïnitieerd** met de **respectievelijke default waarden**

- 0
- false
- null

[illegible]

Creatie & Initialisatie – Array

Java beschikt over een manier om arrays tegelijk te creëren en in te vullen

- **`int numbers[] = {1,3,6,8,45,2};`**

Als de initialisatie niet op dezelfde regel gebeurt kunnen we volgende notatie gebruiken

- **`int numbers[];`**
- **`numbers = new int[] {1,3,6,8,45,2};`**

naam	[0]	[1]	[2]	[3]	[4]	[5]
numbers	1	3	6	8	45	2

Gebruik arrays

We kunnen specifieke elementen van een array gebruiken door de **naam van de array** te nemen gevolgd door vierkante haken met daartussen het nummer van de **index**.

```
numbers [2]
```

```
int sum = numbers[2] + numbers[3];
```

Eigenschappen array

Arrays hebben een eigenschap die het aantal elementen in de array aangeeft.

- length

Hiermee kunnen we als volgt over de array **itereren**:

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

De foreach lus

Syntax

- ```
for(type element : array){
 statement;
}
```

## Example

- ```
for(int n : numbers){  
    System.out.println(n);  
}
```


Arrays van objecten

Arrays kunnen zowel **primitieve datatypen** bevatten als **objecten**

→ In het **laatste geval** spreken we van een **array van objecten**.

◆ `String[] lines = new String[3];`

naam	[0]	[1]	[2]
lines	null	null	null

→ `lines[0] = "Hello World !";`
→ `lines[1] = "Hello Mars !";`
→ `lines[2] = "Hello Venus !";`

Strings en arrays

Een String kan omgezet worden naar een array aan de hand van onderstaande methoden.

De methodes

- `split(String regex)`
- `toCharArray()`

Opdracht arrays



Time to get coding!

Maak een programma dat op de commandline wordt uitgevoerd en waaraan je een zin moet meegeven

- zet deze om in een string array
 - itereer over deze array met een for-each lus
 - Print telkens de waarde naar de console
-
- Optie: zorg ervoor dat het programma niet faalt als je verkeerde input meegeeft

Breid dit programma uit zodat er wordt gezocht naar hoofdletters in de array (hint String omzetten naar een charArray) Het programma zal de respectievelijke hoofdletter omzetten naar een lowercase letter (kijk indien nodig in de Java Character API).

```
powerpoint.end();
```

```
MegaBook m = Noelvaes.getMegaBook();
```

```
Student student = Classroom.getStudent();  
student.takeBook(m);
```

```
student.beginExcercises();
```



Multidem Array's

- Een array bestaande uit array's.
- Aangezien array objecten zijn kunnen ze ook in een Array geplaatst worden
- 3 manieren:
 - `int [][] multiDem = new int[4][0];`
 - `int[0] = new int[4]`
 - `int[0] = new int[3]`
 - `int [][] multiDem2 = new int[4][3];`
 - `int[0][0] = 5;`
 - `int[0][1] = 5;`
 - `int[][] multiDem3 = {{1,2,3},{1,2,3}}`

Lookup tables

- Handige manier om set van data bij een te houden waarvan indexen overeenstemmen met de te onthouden data.
- Voorbeelden
 - schoenmaat met bijhorende aantal in cm
 - maandtabellen
 - inch/cm convertors