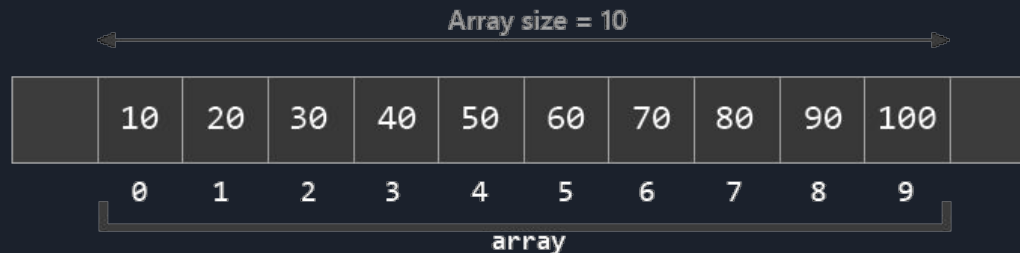




ARRAYS

WHAT ARE ARRAYS

- EEN ARRAY IS EEN VERZAMELING VAN ELEMENTEN MET EEN VAST AANTAL WAARDEN VAN HETZELFDE TYPE.
- DE LENGTE VAN EEN ARRAY WORDT BEPAALD WANNEER DE ARRAY WORDT GEMAAKT.
 - NA HET MAKEN STAAT DE LENGTE VAST.
- ELK ITEM IN EEN ARRAY WORDT EEN ELEMENT GENOEMD EN ELK ELEMENT IS TOEGANKELIJK VIA DE NUMERIEKE INDEX.





HOW TO DECLARE AN ARRAY?

- **NET ALS DECLARATIES VOOR VARIABELEN VAN ANDERE TYPEN, HEEFT EEN ARRAY-DECLARATIE TWEE COMPONENTEN:**
 1. **HET TYPE ARRAY**
 2. **DE NAAM VAN DE ARRAY.**
- **HET TYPE VAN EEN ARRAY WORDT GESCHREVEN ALS TYPE [], WAARBIJ TYPE HET GEGEVENSTYPE VAN DE INGESLOTEN ELEMENTEN IS;**
- **DE HAAKJES ZIJN SPECIALE SYMBOLEN DIE AANGEVEN DAT DEZE VARIABELE EEN ARRAY BEVAT.**



HOW TO DECLARE AN ARRAY?

- DE GROOTTE VAN DE ARRAY MAAKT GEEN DEEL UIT VAN HET TYPE (DAAROM ZIJN DE HAAKJES LEEG).
- DE NAAM VAN EEN ARRAY KAN ELKE GEWENSTE NAAM ZIJN, OP VOORWAARDE DAT DEZE DE REGELS EN CONVENTIES VOLGT

```
vb:          // DECLARES AN ARRAY OF INTEGERS  
            int[ ] numbers;
```



BUT HOW MANY ELEMENTS CAN THIS ARRAY HOLD?

- **GOEDE VRAAG! WE MOETEN GEHEUGEN TOEWIJZEN AAN DE ARRAY. IN DIT GEDEELTE BESLISSEN WE HOEVEEL ELEMENTEN ONZE ARRAY KAN BIJHOUDEN.**

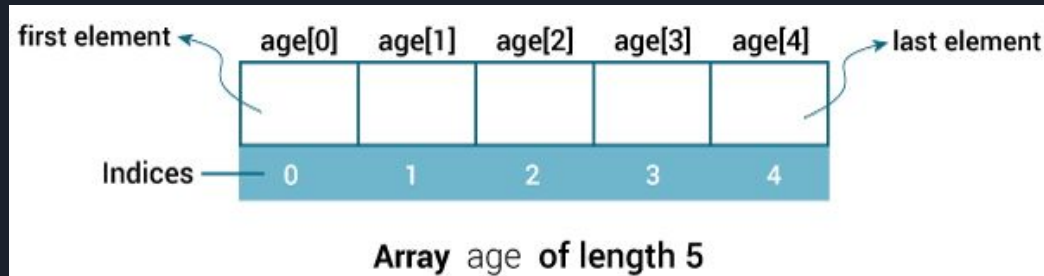
VB: `numbers = new int[10];`

- **HIER IS DE GROOTTE VAN DE ARRAY 10.**
 - **DIT BETEKENT DAT DEZE 10 ELEMENTEN KAN BEVATTEN (10 INT WAARDEN).**
 - **DE GROOTTE VAN EEN ARRAY WORDT OOK WEL DE LENGTE VAN EEN ARRAY GENOEMD.**
- **IN JAVA KUNNEN WE DE DECLARATIE EN INITIALISATIE OOK IN 1 STATEMENT DOEN:**

VB: `int[] ages = new int [5];`

JAVA ARRAY INDEX

- ELK ELEMENT IN EEN ARRAY WORDT GEASSOCIEERD MET EEN POSITIE [INDEX], WE KUNNEN ELEMENTEN BENADEREN VIA DEZE INDEX.



- ALS DE LENGTE VAN EEN ARRAY N IS, IS HET EERSTE ELEMENT VAN DE ARRAY `arrayName[0]` EN HET LAATSTE ELEMENT IS `arrayName[n - 1]`



HOW TO INITIALIZE ARRAYS IN JAVA?

- IN JAVA KUNNEN WE ARRAYS INITIALISEREN TIJDENS HET DECLAREREN OF JE KAN DEZE LATER IN HET PROGRAMMA INITIALISEREN VOLGENS JE BEHOEFTE.
- HIER KAN JE ZIEN HOE JE TIJDENS DE DECLARATIE OOK ONMIDDELIJK EEN INITIALISATIE KAN DOEN

VB: `int[] age = {12, 4, 17, 22, 53};`

- DIT STATEMENT MAAKT EEN ARRAY MET DE NAAM AGE EN INITIALISEERT DEZE MET DE WAARDEN TUSSEN CURLY BRACKETS.
 - DE LENGTE VAN DE ARRAY WORDT BEPAALD DOOR HET AANTAL WAARDEN TUSSEN DE CURLY BRACKETS GESCHEIDEN DOOR KOMMA'S [5 ELEMENTEN IN DIT VB]



OPDRACHT 1

ARRAY OF OBJECTS

- *ARRAYS KUNNEN ZOWEL PRIMITIVE TYPES ALS REFERENCE TYPES BEVATTEN.*
 - *IN DIT GEVAL SPREKEN VAN EEN ARRAY VAN OBJECTEN.*
 - *DE ARRAY HEEFT EEN AANTAL VERWIJZINGEN NAAR OBJECTEN VAN HETZELFDE TYPE*
- *MAKEN DEZE ALS VOLGT AAN*

vb: `String[] lines = new String[5];`





ARRAY OF OBJECTS

- **HEBBEN IN HET VORIGE VOORBEELD GEEN VIJF STRING OBJECTEN GECREËRD, MAAR WEL EEN ARRAY MET VIJF REFERENTIES NAAR EEN STRING OBJECT.**
 - **DEZE WORDEN GEÏNITIALISEERD MET DE WAARDE NULL.**
- **MOETEN DAAROM DE VIJF STRING OBJECTEN AFZONDERLIJK CREËREN EN DE REFERENTIES TOEKENNEN AAN DE ELEMENTEN VAN DE ARRAY.**

vb: String lines = new String[5];

lines[0] = "Hello World";

lines[1] = "Hello Mars";

...

lines[4] = "Hello Saturn";



ARRAYS CLASS IN JAVA

- DE KLASSE `Arrays` IN HET PAKKET `java.util` MAAKT DEEL UIT VAN HET `JAVA COLLECTION FRAMEWORK`.
- DEZE KLASSE BIEDT STATISCHE METHODEN VOOR HET DYNAMISCH MAKEN EN BEHEREN VAN `JAVA-ARRAYS`.
- DEZE KLASSE BESTAAT UIT ALLEEN STATISCHE METHODEN EN DE METHODEN VAN DE KLASSE `Object`.
 - DE METHODEN VAN DEZE KLASSE KUNNEN WORDEN GEBRUIKT DOOR DE KLASSENAAM ZELF.



OPDRACHT 2



MULTIDIMENSIONAL ARRAYS

- ARRAYS DIE WE TOT NU TOE HEBBEN GEZIEN, WORDEN ONE DIMENSIONAL ARRAYS GENOEMD.
- WE KUNNEN OOK MULTIDIMENSIONAL ARRAYS IN JAVA MAKEN.
- EEN MULTIDIMENSIONAL ARRAY IS EEN ARRAY VAN ARRAYS.
 - WIL ZEGGEN, DAT ELK ELEMENT VAN EEN MULTIDIMENSIONAL ARRAY ZELF EEN ARRAY IS.
- HIER MAKEN WE EEN MULTIDIMENSIONALE INT ARRAY MET DE NAAM MATRIX.

```
VB:  int [][] matrix = new int [2] [3];  
      int [][] matrix = { { 1, 4, 4}, { 4, -1, 25} };
```

HOW CAN WE VISUALIZE THIS?

```
int [][] matrix = {{ 1, 4, 39 }, { 4, -1, 25 }, { 8, 14, 2 }};
```

COLUMNS			
ROWS	[0][0]: 1	[0][1]: 4	[0][2]: 39
	[1][0]: 4	[1][1]: -1	[1][2]: 25
	[2][0]: 8	[2][1]: 14	[2][2]: 2



OPDRACHT 3