



Programming Basics

Hoofdstuk 7

Arrays

DE HOGESCHOOL MET HET NETWERK

Hogeschool PXL – Elfde-Liniestraat 24 – B-3500 Hasselt
www.pxl.be - www.pxl.be/facebook



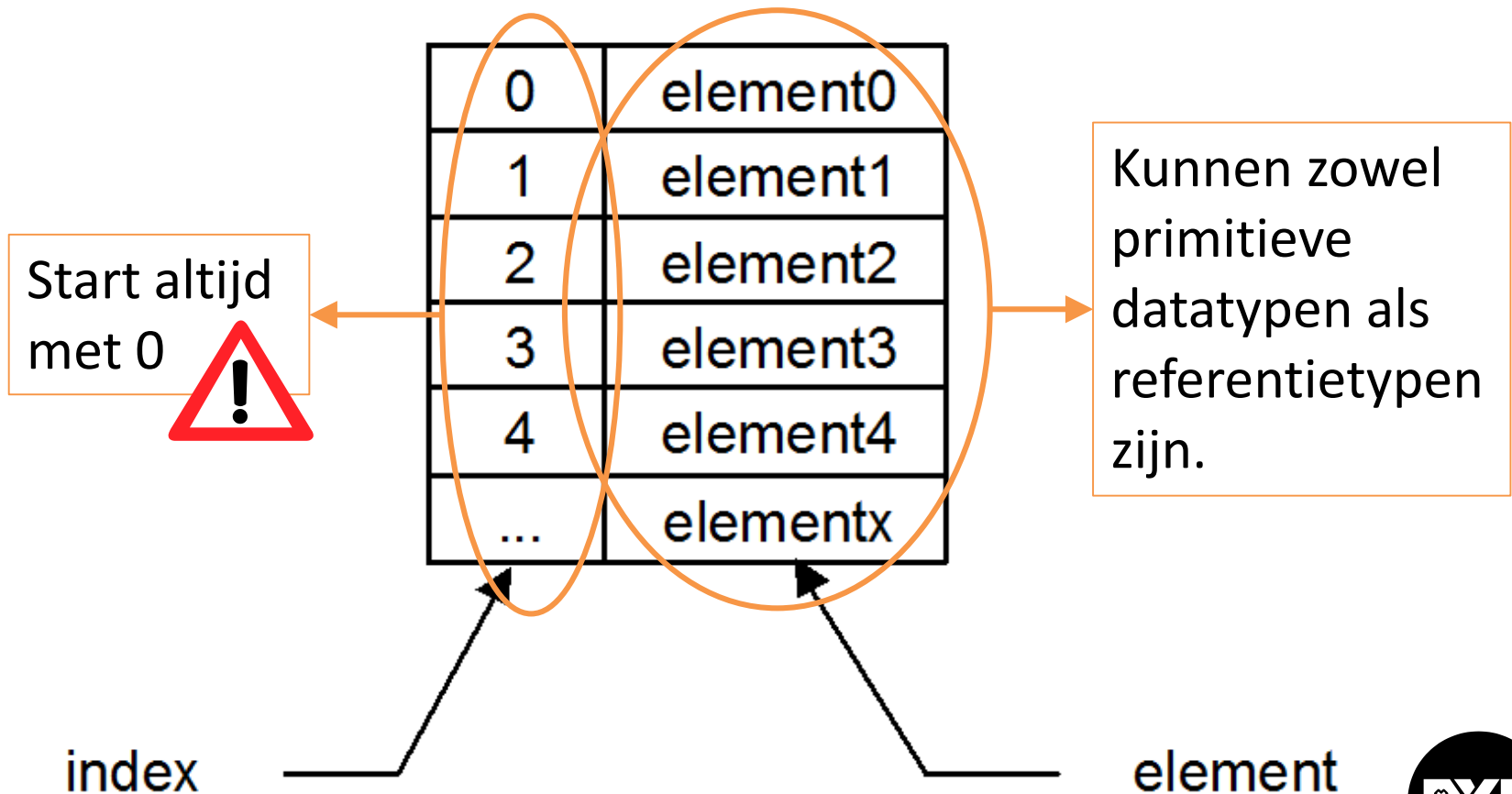
Inhoud

1. Inleiding
2. Arrays maken
3. Arrays gebruiken
4. De uitgebreide *for*-lus (*for each*)
5. Arrays van objecten
6. Arrays van arrays
7. Lookup tables
8. Samenvatting



1. Inleiding

Een bijzonder type object is de *array*.



= verzameling van elementen van hetzelfde type waarbij ieder element voorzien is van een nummer (index) dat de plaats van dat element in de *array* aanduidt.

Wanneer gebruik je een *array*: wanneer je een grote hoeveelheid getallen (of objecten: bv woorden) in het geheugen moet bewaren is het bijna onmogelijk (niet efficiënt) om voor al deze gegevens een andere variabele te gebruiken → *array*

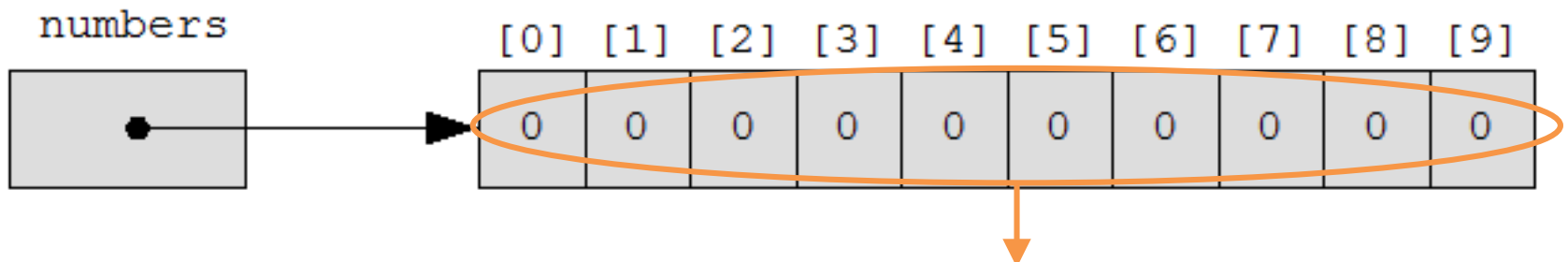


2. Arrays maken

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

Mag variabele zijn.

Arrays hebben een vaste lengte en kunnen nadien niet meer gewijzigd worden.



Wordt automatisch geïnitieerd op 0 (null voor referentietypen/false voor booleans).

Declaratie en initialisatie :

```
int[] numbers = { 1, 3, 6, 8, 4 };
```

||

```
int[] numbers;
```

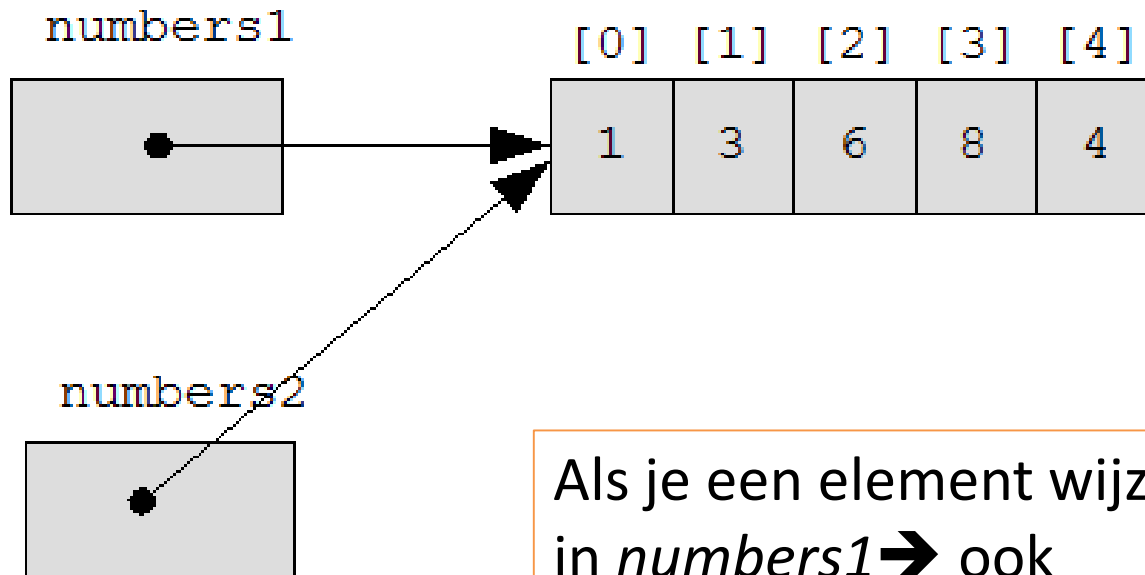
```
numbers = new int[] { 1, 3, 6, 8, 4 };
```

= anonieme *array*

Lengte van de *array* is niet gespecificeerd (wordt bepaald door het aantal geïnitieerde elementen)

Een array toekennen aan een ander arrayobject :

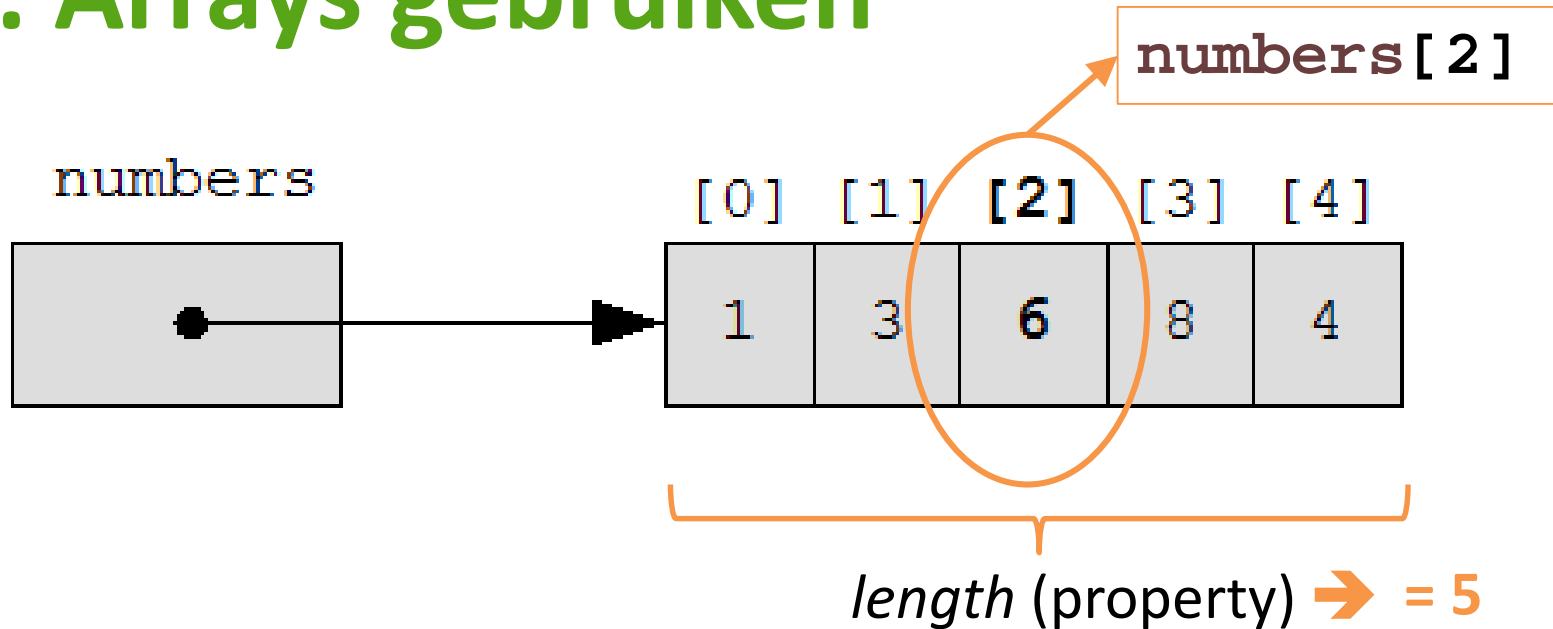
```
int[] numbers1 = { 1, 3, 6, 8, 4 };  
int[] numbers2 = numbers1;
```



Als je een element wijzigt
in *numbers1* → ook
gewijzigd in *numbers2*



3. Arrays gebruiken



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

Waarom '<'?



Een waarde aan een array-element toekennen:

```
numbers[2] = 12;  
numbers[4] = getal;  
numbers[5] = -4; → java.lang.ArrayIndexOutOfBoundsException
```

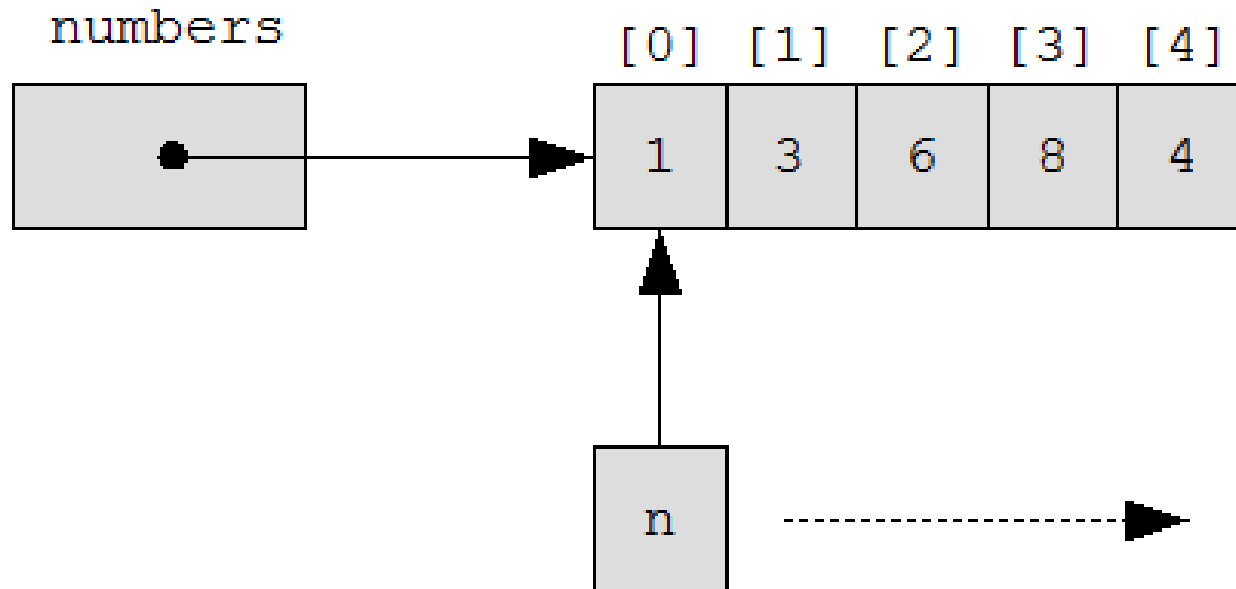


4. De uitgebreide *for*-lus (*for each*)

```
int[] numbers = { 1, 3, 6, 8, 4 };  
  
for (int n : numbers) {  
    System.out.println(n);  
}
```

De variabele *n* neemt tijdens de iteratie één voor één de waarde aan van de elementen uit de *array*.





Voordeel: korte notatie

Nadeel:

- Je beschikt niet over de indexwaarde.
- Enkel geschikt als je alle waarden van de tabel wil doorlopen.

Opdracht 1: *Arrays gebruiken*

- a)
- Maak een *array* van 20 getallen en vul deze met veelvouden van 7.
 - Druk de *array* in volgorde af op het scherm met een *for each*-lus.
 - Maak een *array* van *boolean*-waarden en druk deze *array* af met een *for each*-lus.
 - Schrijf een programma waarbij je in een tabel van lengte 10 de volgende waarden plaatst 1, 3, 5, 7, 9, 11, 13, 15, 17, 19.
 1. door de tabel te declareren en vervolgens waarden toe te kennen.
 2. door de tabel onmiddellijk te initialiseren.
- Druk deze waarden af door gebruik te maken van de *for each*-lus.
Druk daarna de waarden in omgekeerde volgorde af.




b) Probeer (zonder het programma uit te voeren) de output van onderstaande code te geven.

```
int[] a = new int[6];
int[] b = new int[6];

for (int i = 0; i < a.length; i++) {
    a[i] = i * 2;
}
b = a;
b[5] = 99;

for (int i = 0; i < a.length; i++) {
    System.out.printf("a[%d] = %2d b[%d] = %2d %n",
                      i, a[i], i, b[i]);
}
```

= newline



c) Probeer (zonder het programma uit te voeren) de output van onderstaande code te geven.

```
int[] a = new int[6];
int[] b = new int[6];

for (int i = 0; i < a.length; i++) {
    a[i] = i * 2;
    b[i] = a[i];
}
b[5] = 99;

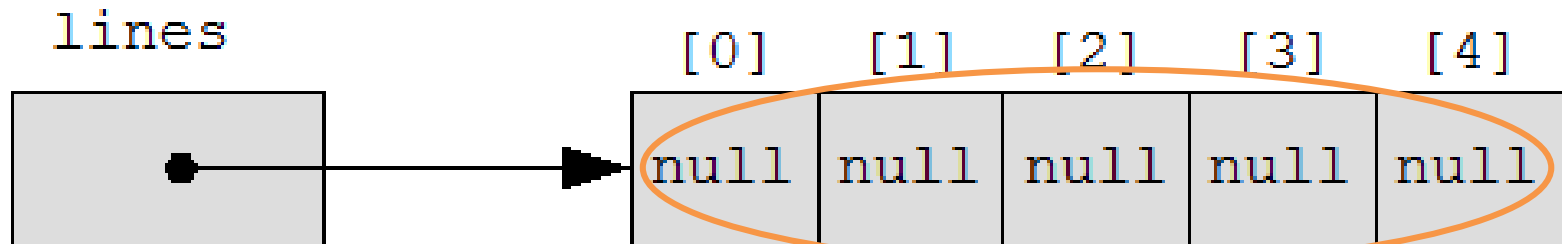
for (int i = 0; i < a.length; i++) {
    System.out.printf("a[%d] = %2d b[%d] = %2d  %n",
                      i, a[i], i, b[i]);
}
```



5. Arrays van objecten

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

Hier worden geen 5 strings aangemaakt, maar wel een *array* van 5 referenties naar een 5 *String*-objecten.



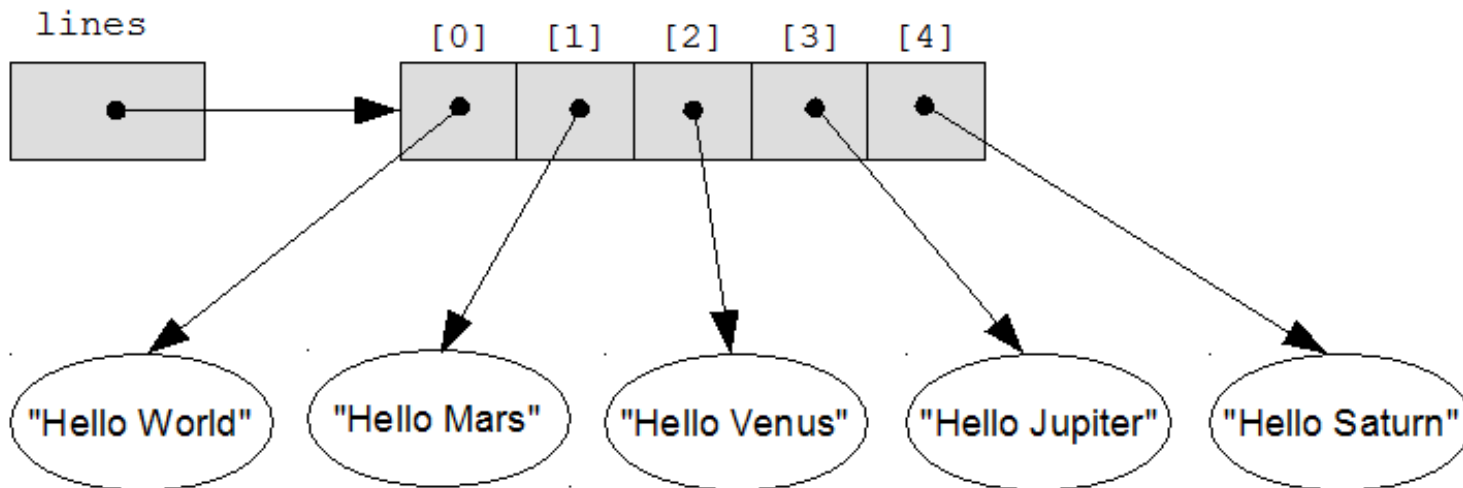
Automatisch geïnitialiseerd op **null**!



5. Arrays van objecten

➔ 5 *String*-objecten creëren en de referenties toekennen aan de elementen van de *array*.

```
lines[0] = "Hello World";  
lines[1] = "Hello Mars";  
lines[2] = "Hello Venus";  
lines[3] = "Hello Jupiter";  
lines[4] = "Hello Saturn";
```



Of: *array* initialiseren tijdens declaratie van de *array*.

```
String[] lines = { "Hello World",  
                  "Hello Mars",  
                  "Hello Venus",  
                  "Hello Jupiter",  
                  "Hello Saturn" }
```



Methoden die een array teruggeven (zie JavaDoc)

```
String text = "I just want to say hello!";  
String[] words = text.split(" ");  
for(String word: words) {  
    System.out.println(word);  
}
```

Output: I
just
want
to
say
hello!

= scheidingstekst

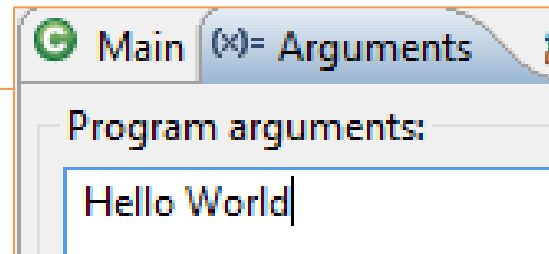
Kan om het even welke scheidingstekst zijn (", " en ";" worden veel gebruikt bij .csv-bestanden).

Woorden (*argumenten*) meegeven bij het opstarten van een programma

```
public class HelloApp {  
    public static void main(String[] args)  
    {  
        for(String arg: args) {  
            System.out.println(arg);  
        }  
    }  
}
```

Output: Hello
World

Het is gebruikelijk deze woorden mee te geven aan de commandolijn. Maar we kunnen dat ook vanuit Eclipse: *Run > Run Configurations... > Arguments > Program arguments*: typ hier de woorden gescheiden door een spatie.



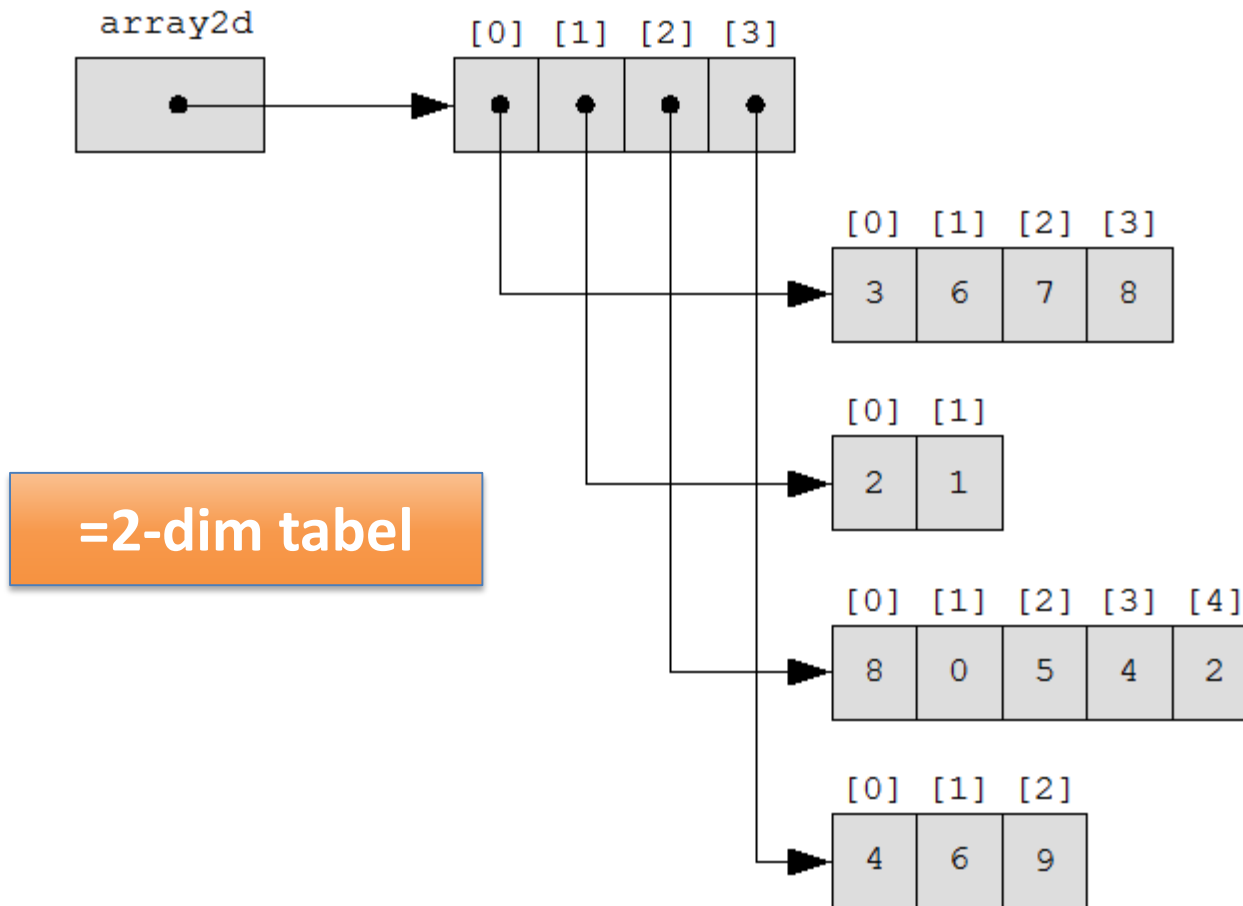
Opdracht 2: *Arrays van objecten*

Maak een programma dat een regel tekst aan de gebruiker vraagt en de woorden onder elkaar afdrukt op het scherm.



6. Arrays van arrays (= meerdimensionale array)

```
int[][] array2d = { { 3, 6, 7, 8 }, { 2, 1 },  
                   { 8, 0, 5, 4, 2 }, { 4, 6, 9 } };
```



Of: `int[][] array2d = new int[4][];` → Creatie van array van arrays

```
array2d[0] = new int[4];  
array2d[1] = new int[2];  
array2d[2] = new int[5];  
array2d[3] = new int[3];
```

Voor elk element uit de array wordt een array van integers gecreëerd.

```
array2d[0][0] = 3;  
array2d[0][1] = 6;  
array2d[0][2] = 7;  
array2d[0][3] = 8;  
array2d[1][0] = 2;  
array2d[1][1] = 1;  
array2d[2][0] = 8;  
array2d[2][1] = 0;  
array2d[2][2] = 5;  
array2d[2][3] = 4;  
array2d[2][4] = 2;  
array2d[3][0] = 4;  
array2d[3][1] = 6;  
array2d[3][2] = 9;
```

Elementen krijgen een waarde

1 element

Afdruk 2-dim. array:

```
for (int i = 0; i < array2d.length; i++) {  
    for (int j = 0; j < array2d[i].length; j++) {  
        System.out.print(array2d[i][j] + "\t");  
    }  
    System.out.println();  
}
```

Output?

Of met *for each*:

```
for (int[] row : array2d) {  
    for (int el : row) {  
        System.out.print(el + "\t");  
    }  
    System.out.println();  
}
```

3	6	7	8	
2	1			
8	0	5	4	2
4	6	9		



Indien de *arrays* allemaal dezelfde lengte hebben, kan je een 2-dim. tabel als volgt declareren:

```
int[][] table = new int[4][7];
```

= 4 rijen en 7
kolommen

Zowel de hoofd-*array* als de sub-*array* worden tegelijkertijd gecreëerd. De hoofd-*array* wordt ingevuld met referenties naar de sub-*arrays*.

Opdracht 3: *2-dimensionale arrays*

- a) Maak een tabel van 4 rijen en 6 kolommen. Vul de tabel met het product van het rijnummer en kolomnummer. Druk de matrix af op het scherm en gebruik hierbij een *for each*-lus.



b) Probeer (zonder het programma uit te voeren) de output van onderstaande code te geven.

```
int x = 0;
int a[][];
a = new int[5][];
// kolommen creëren
for (int i = 0; i < a.length; i++) {
    a[i] = new int[i + 1];
}
// tabel vullen
for (int i = 0; i < a.length; i++) {
    for (int j = 0; j < i + 1; j++) {
        a[i][j] = x++;
    }
}
// tabel drukken
for (int i = 0; i < a.length; i++) {
    for (int j = 0; j < i + 1; j++) {
        System.out.printf("a(%d,%d) = %d \t", i, j, a[i][j]);
    }
    System.out.println();
}
```



7. Lookup tables

Gegevens worden opgezocht a.d.h.v. hun index.

(lange if else- of switch case-instructies worden zo vermeden)

```
String[] days = { "maandag", "dinsdag",  
                  "woensdag", "donderdag",  
                  "vrijdag", "zaterdag",  
                  "zondag" };  
  
int day = 2;  
System.out.println(days[day - 1]);
```



Opdracht 4: *Lookup tables*

Maak een lookup table met het aantal dagen van de maand. Laat de gebruiker een bepaald maandnummer ingeven en toon a.d.h.v. je lookup table hoeveel dagen deze maand bevat.



8. Call-by-value ↔ Call-by-reference

```
public static void main(String[] args) {  
    int a0 = 1;  
    int a1 = 2;  
    int a2 = 3;  
    wijzig(a0, a1, a2);  
    System.out.println(a0);  
}
```

= call-by-value

```
public static void wijzig(int a0, int a1, int a2) {  
    a0 = 0;  
    a1 = 0;  
    a2 = 0;  
}
```



```
public static void main(String[] args) {  
    int[] a = new int[3];  
    a[0] = 1;  
    a[1] = 2;  
    a[2] = 3;  
    wijzig(a);  
    System.out.println(a[0]); → 0  
}
```

= call-by-reference

```
public static void wijzig(int[] a) {  
    a[0] = 0;  
}
```

7. Samenvatting

- Arrays:
 - Aanmaken
 - Invullen
 - Opvragen
- Arrays van
 - Primitieve datatypes
 - Objecten
- Array van arrays = meerdimensionale arrays

