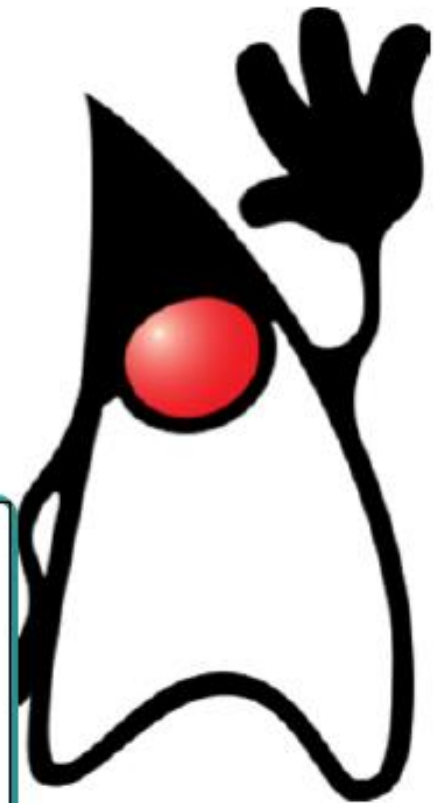


Arrays in Java

Klasgroep	1EO-ICT
Opleiding	Bachelor Elektronica-ICT
Lokaal	Groot Auditorium
Tijdstip	maandag lestijd 3
Docent	Katja Verbeeck
Contact	katja.verbeeck@odisee.be
Handboek	hfst 5 - p 135 - p 166



Inhoud

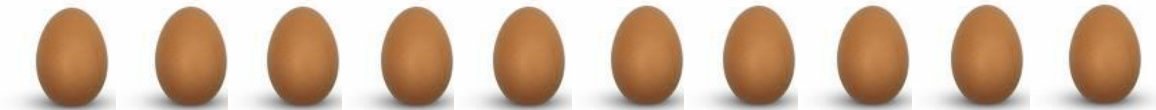
Arrays in meerdere dimensies

- **2 dimensies**
- **3 dimensies**

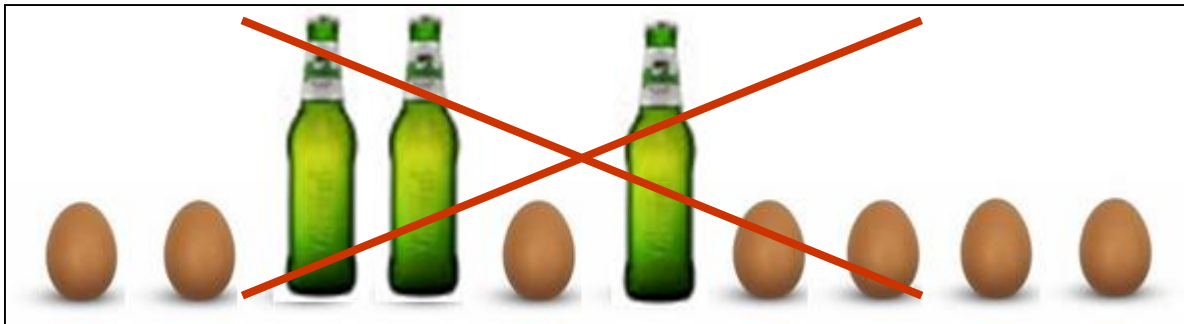
Ongeordend elementen



Geordende elementen



Alle elementen in de rij van zelfde type



meerdere dimensies



Declaratie, instantiatie, initialisatie

//1) declaratie van een array

```
int[] balRij;
```

//2) instantiatie van een array

```
balRij = new int[6];
```

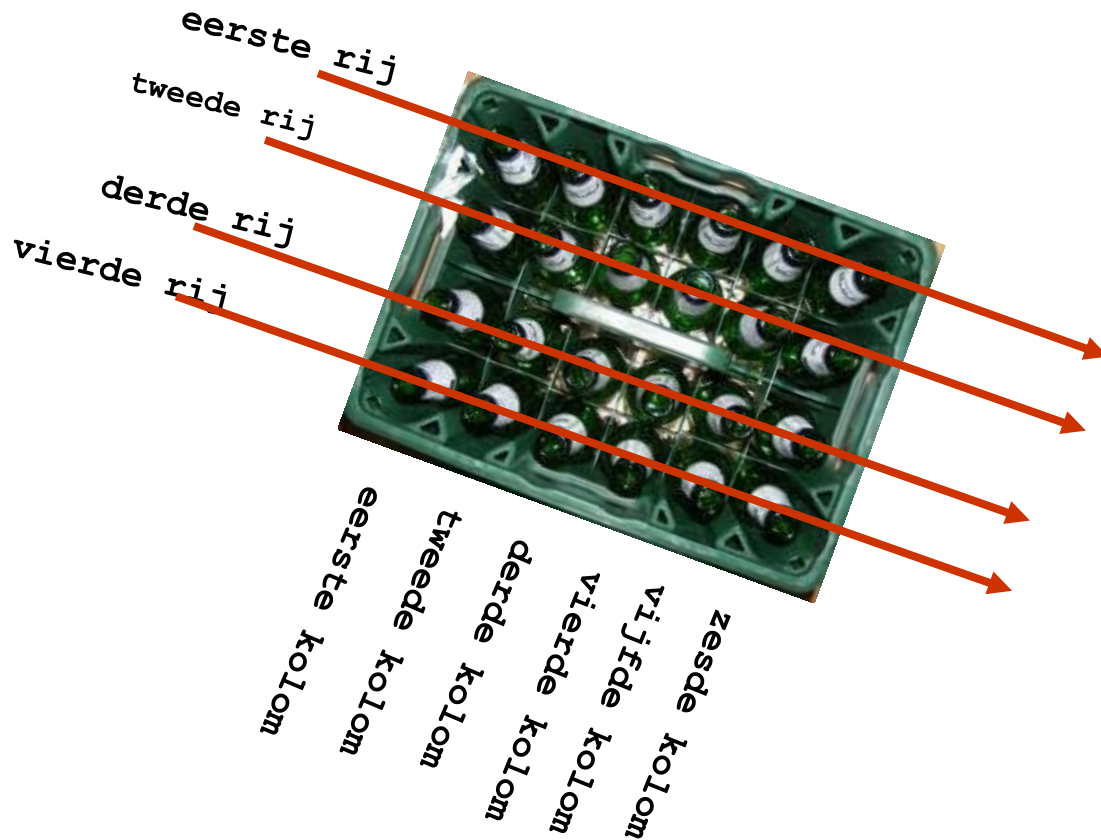
//3) initialisatie van de array

```
for (int i = 0 ; i < balRij.length ; i++) {  
    balRij[i] = (int) (Math.random() * 45) + 1;  
}
```

Ordenen in meerdere dimensies



Ordenen in 2 dimensies : rijen en kolommen

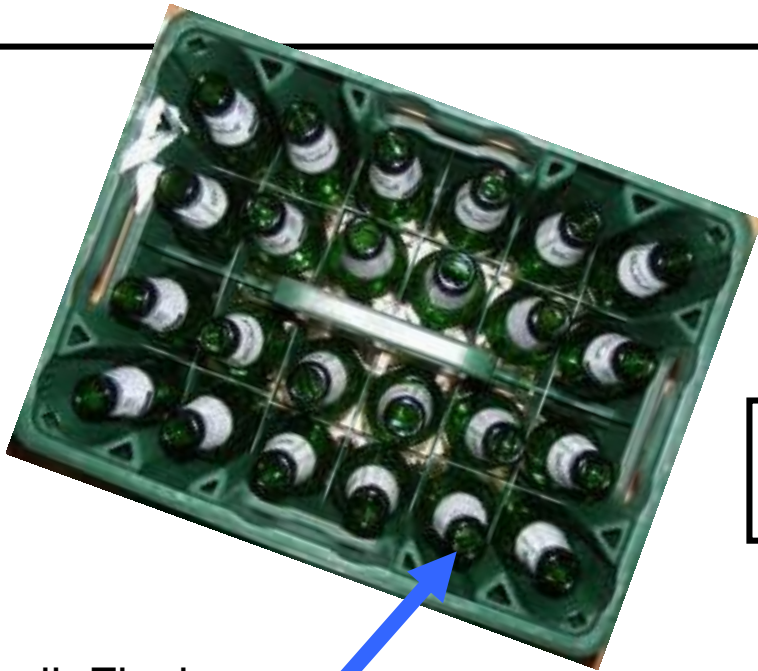


```
boolean[][] krat;           //declaratie van een 2 dim array  
krat = new boolean[4][6];  //aanmaken van het 2 dim array object
```



```
boolean[][] krat; //declaratie van een 2-dimensionale array
krat = new boolean[4][6]; //aanmaken van het array object

//opvullen met flesjes
for (int rij = 0; rij < 4; rij++) {    //vul de rijen
    for (int kol = 0; kol < 6; kol++) { //vul elk elem
        krat[rij][kol] = true;
    }
}
```



mijnFlesje



```
//neem een blikje uit het krat
boolean mijnFlesje = krat[3][4];
```

```
final int RIJEN = 4;
final int KOLOMMEN = 6;

boolean[][] krat;
krat = new boolean[RIJEN][KOLOMMEN];

for (int rij = 0; rij < RIJEN; rij++) {
    for (int kol = 0; kol < KOLOMMEN; kol++) {
        krat[rij][kol] = true;
    }
}
```

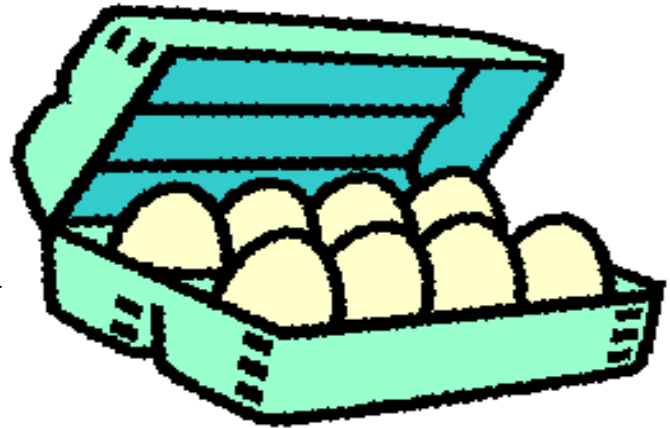

Ordenen in 2 dimensies

Gestructureerd breken van eitjes

Via een java programma houden we de status van de eitjes bij :

- Gebroken (`true`)
- Niet gebroken (`false`)

doos

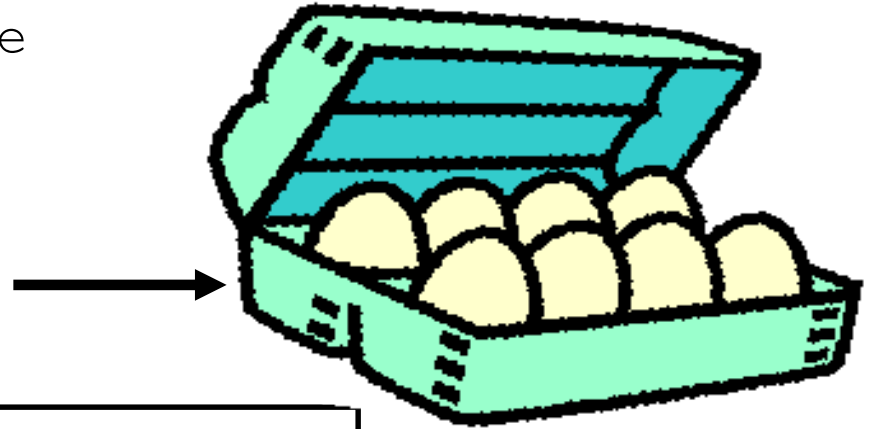


```
final int RIJEN = 2;  
final int KOLOMMEN = 4;  
  
boolean[][] doos;  
doos = new boolean[RIJEN][KOLOMMEN];
```

Ordenen in 2 dimensies

Bij het aanmaken van een nieuw 2dim array object krijgen alle elementen een default waarde :
voor boolean is die default waarde `false`
er is nog geen enkel eitje gebroken

doos

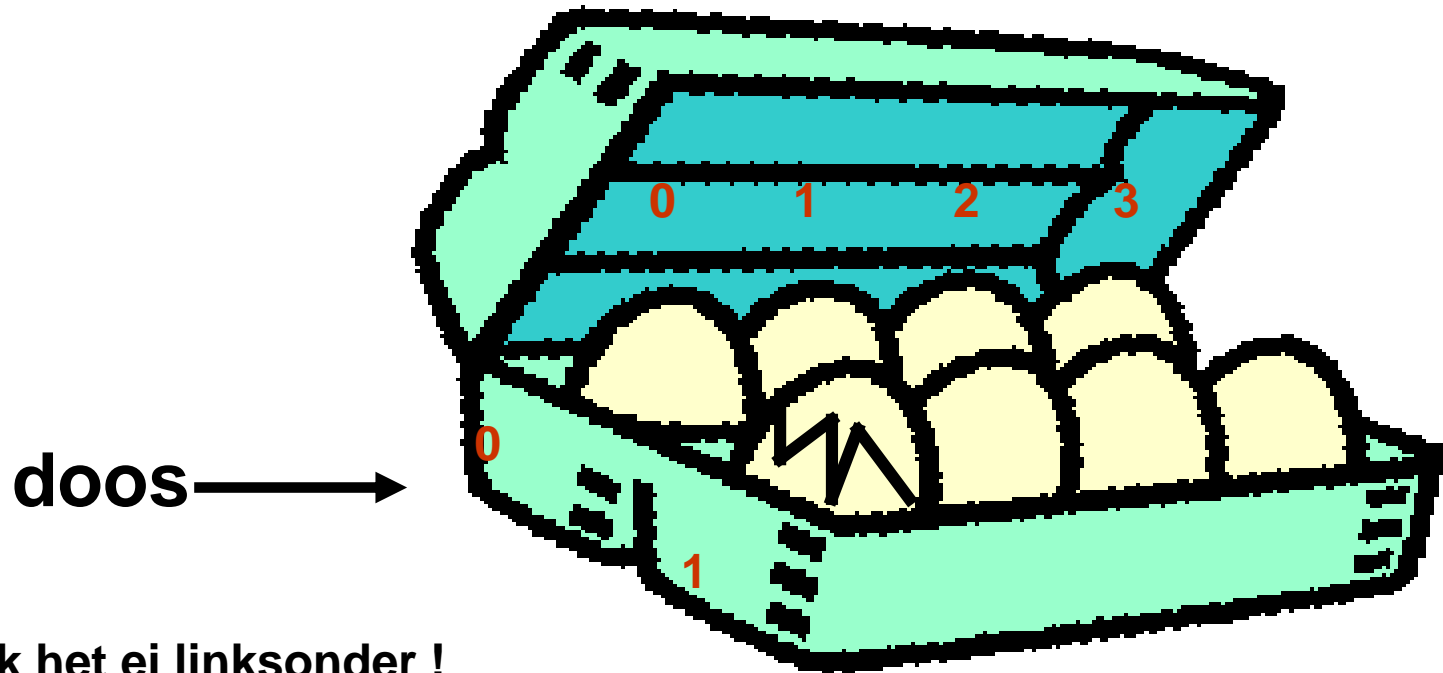


```
for (int i = 0; i < RIJEN; i++) {  
    for (int j = 0; j < KOLOMMEN; j++) {  
        System.out.print(doos[i][j] + ' ');  
    }  
  
    System.out.println();  
}
```

Uitvoer:

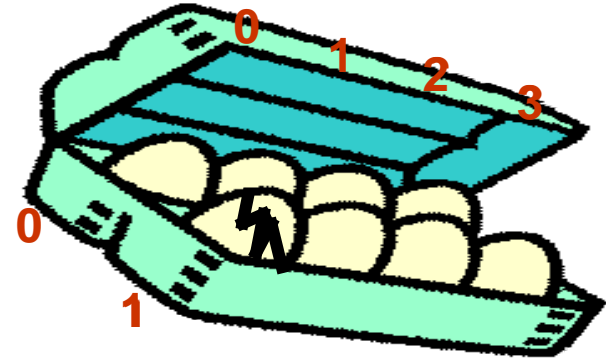
```
false false false false  
false false false false
```

Ordenen in 2 dimensies



```
//Breek het ei  
linksonder  
doos[1][0] = true;
```

doos →



Voorstelling in het geheugen

doos

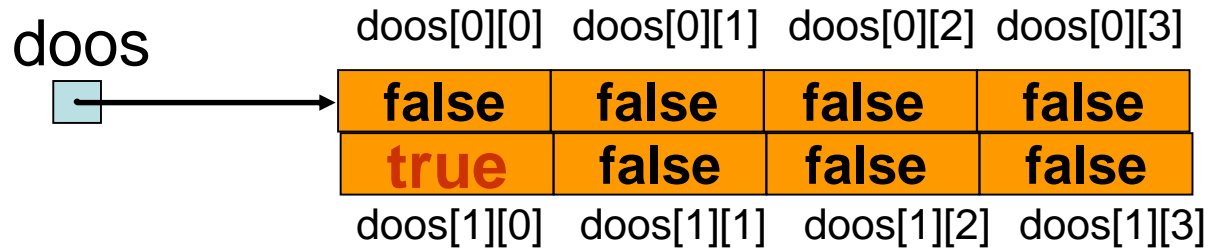
→	false	false	false	false
	true	false	false	false

```
boolean isGebroken;
```

```
isGebroken = doos[0][0]; //geeft false
```

```
isGebroken = doos[1][0]; //geeft true
```

Lengte van een 2dim array

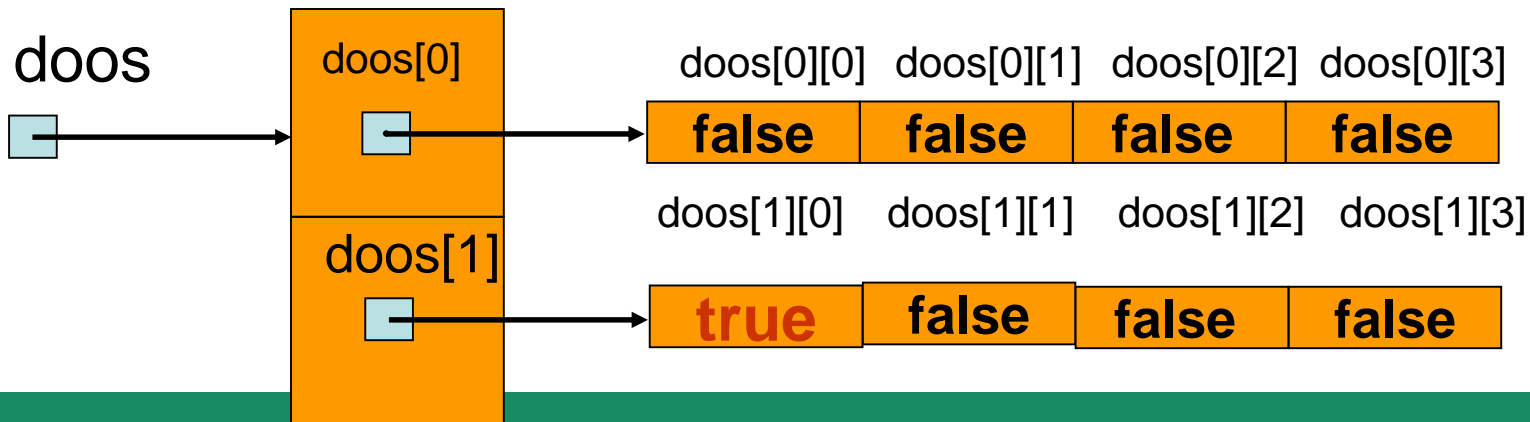


```
int lengte = doos.length; //uitvoer ?
```

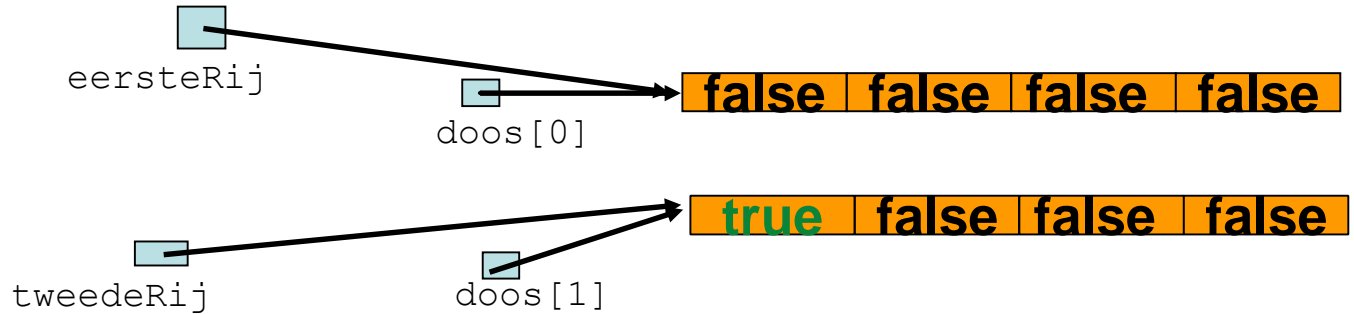
2 !

De lengte van een tweedimensionale array geeft het aantal rijen terug (de eerste dimensie van de 2dim array)

een tweedimensionale array bestaat eigenlijk uit 2 enkelvoudige rijen



Een 2 dim array is eigenlijk 2 x een 1 dim array

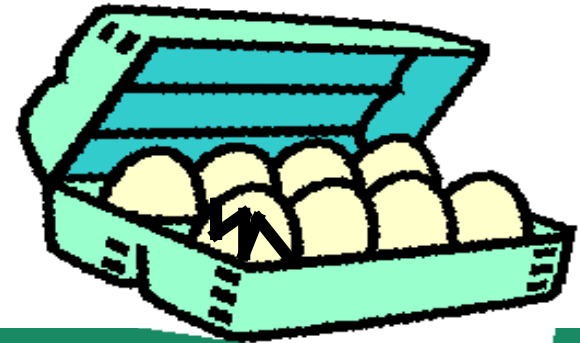


Als je van een tweedimensionele array slechts 1 dimensie opvraagt dan verkrijg je een eendimensionele array

```
boolean[] eersteRij = doos[0];  
boolean[] tweedeRij = doos[1];
```

```
boolean isGebroken;
```

```
isGebroken = eersteRij[0]; //geeft false  
isGebroken = tweedeRij[0]; //geeft true
```



Tel het aantal eitjes dat gebroken is

doos



false	false	false	false
true	false	false	false

```
int aantalGebroken = 0;

for (int i = 0; i < doos.length; i++) {
    for (int j = 0; j < ? ; j++) {
        if (doos[i][j] == true) {
            aantalGebroken++;
        }
    }
}

System.out.println("Er zijn " + aantalGebroken
    + " eitjes gebroken");
```


Tel het aantal eitjes dat gebroken is

```
int aantalGebroken = 0;

for (int i = 0; i < doos.length; i++) {
    for (int j = 0; j < doos[i].length; j++) {
        if (rij[i][j] == true) {
            aantalGebroken++;
        }
    }
}

System.out.println("Er zijn " + aantalGebroken +
    " eitjes gebroken");
```

De derde dimensie



Grondplan van een appartementsgebouw

app1	app2	app3	app4	app5	app6	app7	app8	app9	app10
app11	app12	app13	app14	app15	app16	app17	app18	app19	app20

De klasse Appartement houdt per appartement het aantal bewoners bij

```
public class Appartement {  
    final int HOOGTE = 8;  
    final int DIEPTE = 2;  
    final int BREEDTE = 10;  
  
    int[][][] appt = new int[HOOGTE][DIEPTE][BREEDTE];  
  
    public void voegNieuweHuurderToe() {...}  
    public void geefApptStatusOverzicht() {...}  
    public int telAantalBewoners()  
  
}
```

```

public void voegNieuweHuurderToe() {
    Scanner scan = new Scanner(System.in);
    System.out.println("\nOp welke verdieping komt de nieuwe huurder [0, "
        + (HOOGTE - 1) + "] ?");

    int verdieping = scan.nextInt();

    System.out.println("Kies een appartement [1, " + (DIEPTE * BREEDTE) + "]");
    int keuzeAppt = scan.nextInt() - 1;

    int rij = keuzeAppt / BREEDTE;
    int kol = keuzeAppt % BREEDTE;

    if (appt[verdieping][rij][kol] == 0) {
        System.out.println("Met hoeveel komt men hier wonen ?");
        appt[verdieping][rij][kol] = scan.nextInt();
    }
    else {
        System.out.println("Hier woont al iemand...");
    }
}
}

```

Op welke verdieping komt de nieuwe huurder [0, 7] ?

1

Kies een appartement [1, 20]

15

Met hoeveel komt men hier wonen ?

12

```

public void geefApptStatusOverzicht() {
    for (int i = 0; i < HOOGTE; i++) {
        System.out.println("Overzicht voor verdieping_" + i);

        for (int j = 0; j < DIEPTE; j++) {
            for (int k = 0; k < BREEDTE; k++) {
                System.out.print("\t" + appt[i][j][k]);
            }
            System.out.println();
        }

        System.out.println();
    }
}

```

Overzicht voor verdieping_0

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Overzicht voor verdieping_1

0	0	0	0	0	0	0	0	0	0
0	0	0	0	12	0	0	0	0	0

Overzicht voor verdieping_2

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

3 geneste for lussen !

```
public int telAantalBewoners() {  
    int teller = 0;  
  
    for (int i = 0; i < appt.length; i++) {  
        for (int j = 0; j < appt[i].length; j++) {  
            for (int k = 0; k < appt[i][j].length; k++) {  
                teller += appt[i][j][k];  
            }  
        }  
    }  
  
    return teller;  
}
```