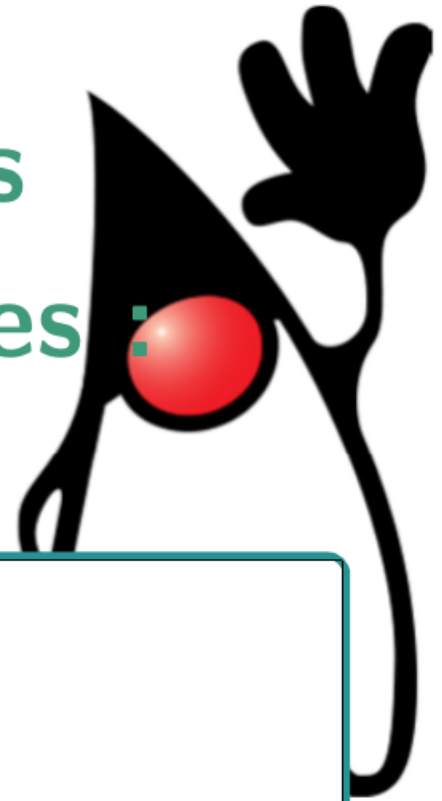


Programming Fundamentals

Arrays in meerdere dimensies :



Klasgroep	1EO-ICT
Opleiding	Bachelor Elektronica-ICT
Theorie	DI: 8:45 - 9:45 E032 WOE: 11:45 - 12:45 J007
(Werk)Labo	DI: 10:00 - 12:00 + 12:00 - 12:45 + 13:30 - 14:30 WOE: 13:30 - 15:30 + 15:30 - 17:30
Docent	Katja Verbeeck
Contact	katja.verbeeck@odisee.be

Inhoud

Arrays in meerdere dimensies

- Voorbeelden 2 dimensies
- Voorbeeld 3 dimensies

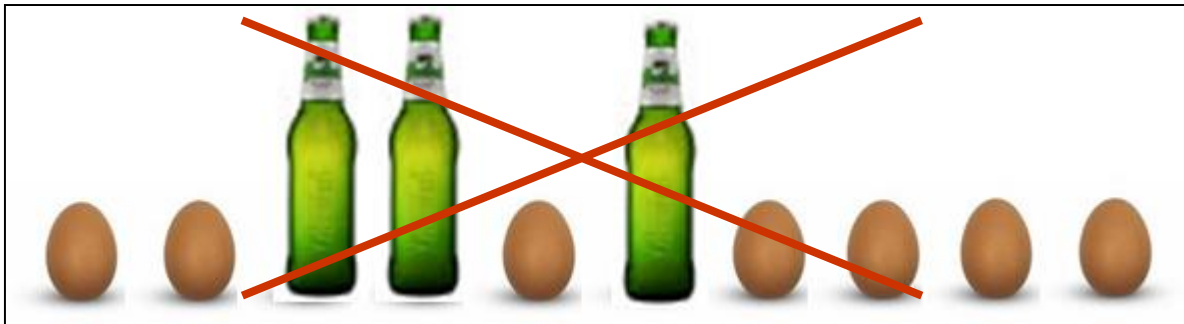
Ongeordend elementen



Geordende elementen



Alle elementen in de rij van zelfde type



meerdere dimensies



Declaratie, instantiatie, initialisatie

//1) declaratie van een array

```
int[] lotto;
```

//2) instantiatie van een array

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

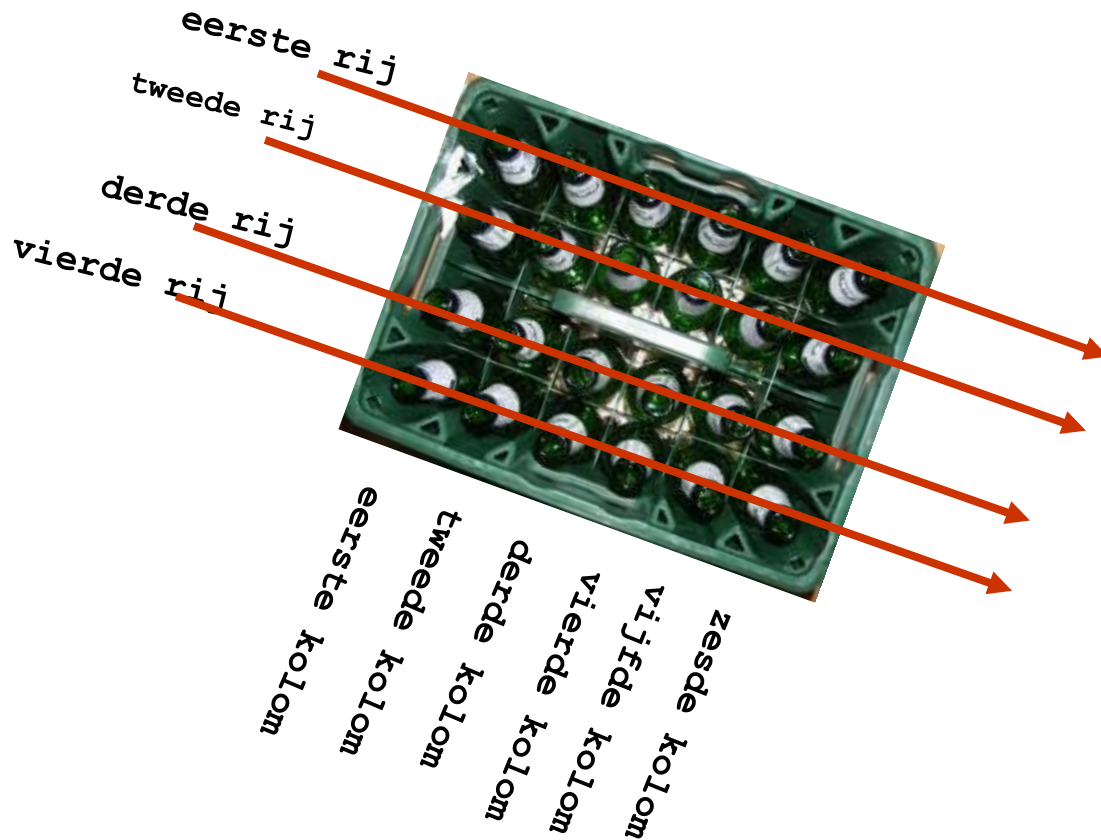
//3) initialisatie van de array

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

Ordenen in meerdere dimensies



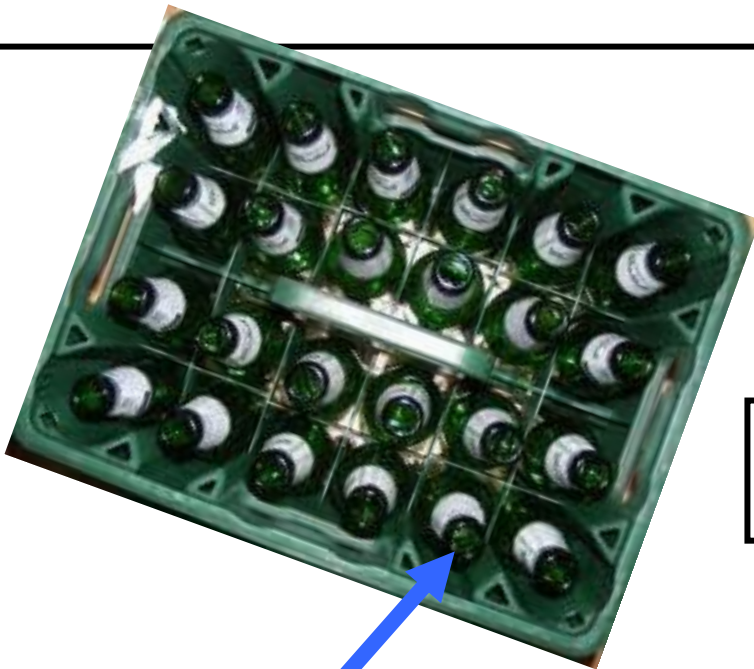
Ordenen in 2 dimensies : rijen en kolommen



```
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];
```


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`)

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


Ordenen in 2 dimensies

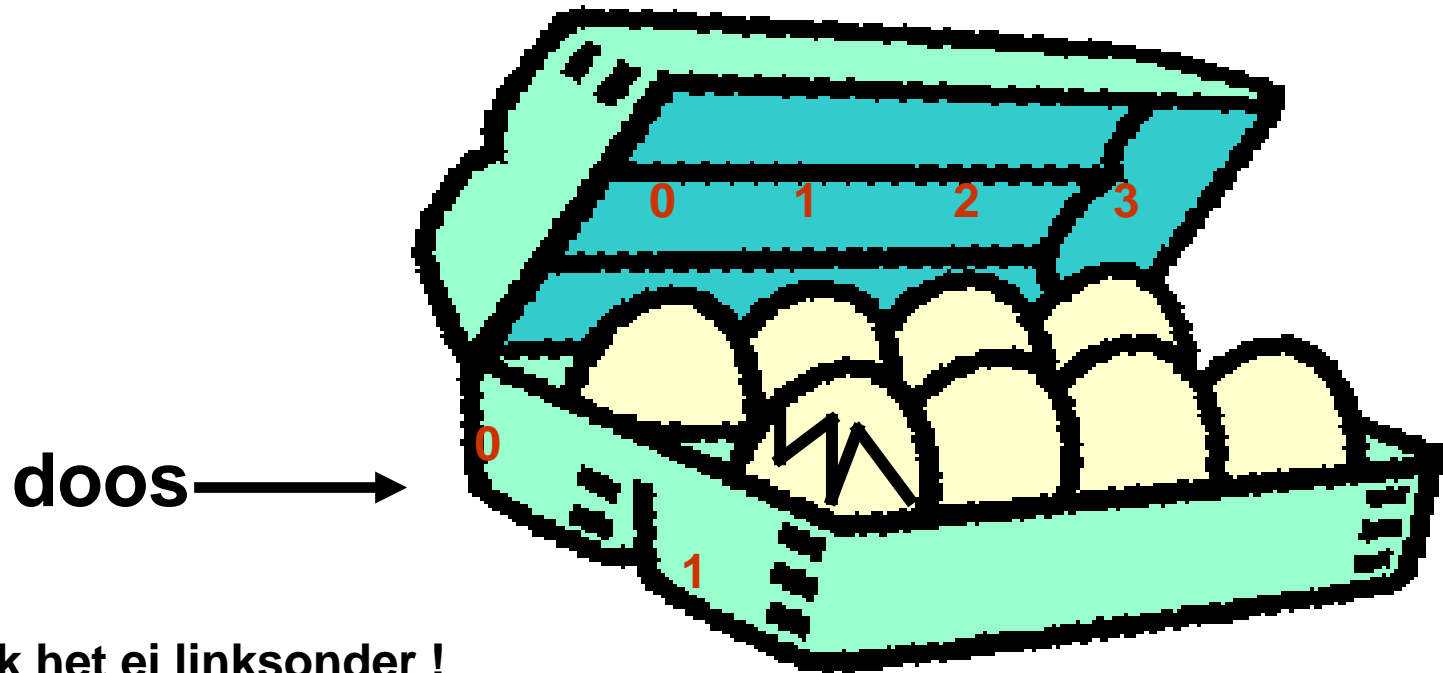
Bij het aanmaken van een nieuw 2dim array object
krijgen alle elementen een default waarde :
voor boolean is die default waarde `false`
er is nog geen enkel eetje gebroken

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

Uitvoer:

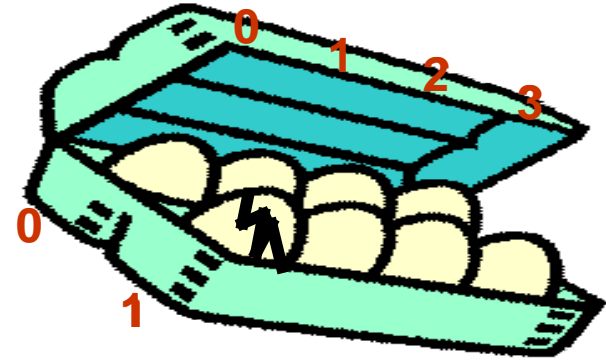
```
false false false false  
false false false false
```

Ordenen in 2 dimensies



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

doos →



Voorstelling in het geheugen

doos

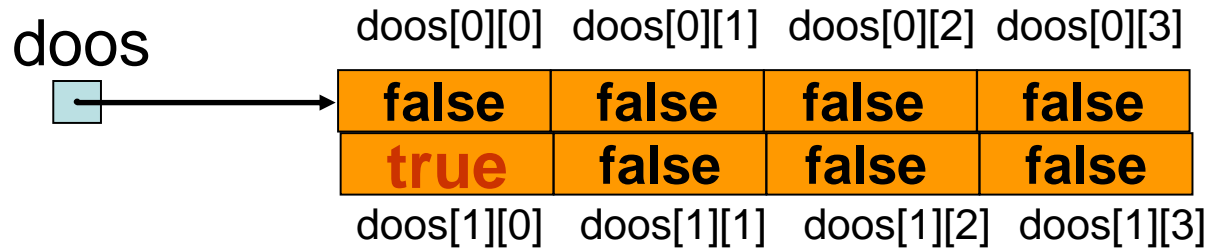
→	false	false	false	false
	true	false	false	false

```
boolean isGebroken;
```

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

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

Lengte van een 2dim array

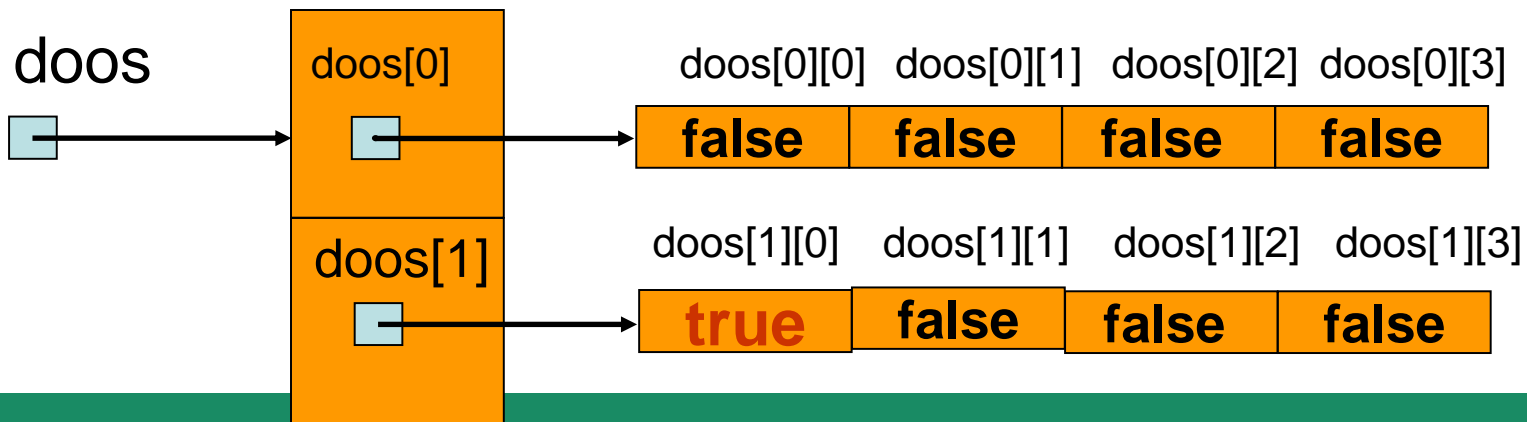


```
int lengte = doos.length; //uitvoer ?
```

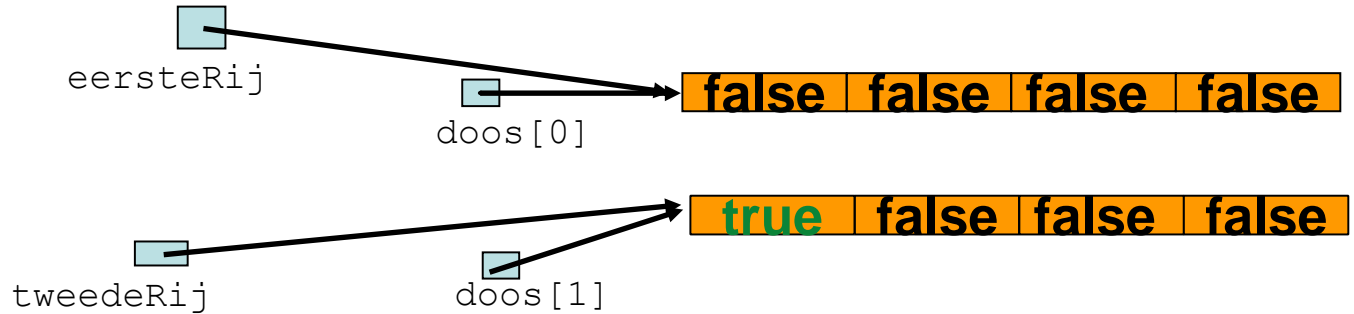
2 !

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];
```

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 (doos[i][j]) {
            aantalGebroken++;
        }
    }
}

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


Schrijf een programma dat een opgegeven aantal keer naar een naam (voornaam en familienaam) vraagt. Bewaar de informatie in een 2 dim array. Vervolgens worden de initialen afgedrukt

Kristien	Van Assche
Peter	Demeester
Tim	Vermeulen

```
//1) declaratie
```

```
String[][] docenten;
```

```
//2) instantiatie
```

```
docenten = new String[3][2];
```

//3) initialisatie van de rij

```
docenten[0][0] = "Kristien";  
docenten[0][1] = "Van Assche";
```

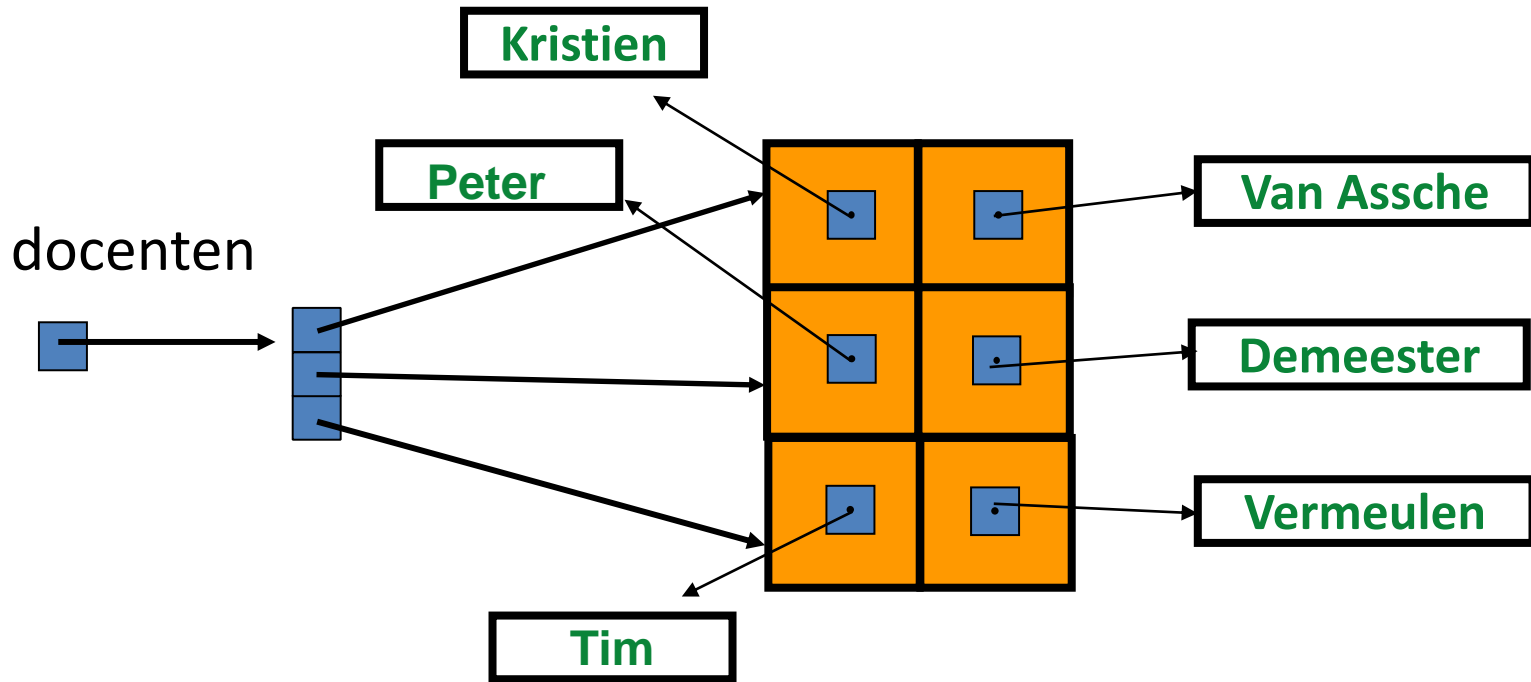
```
docenten[1][0] = "Peter";  
docenten[1][1] = "Demeester";
```

```
docenten[2][0] = "Tim";  
docenten[2][1] = "Vermeulen";
```

//3) of via de initializer syntax:

```
String[][] docenten = {  
    {"Kristien", "Van Assche" },  
    { "Peter", "Demeester" },  
    {"Tim", "Vermeulen" } };
```

Voorstelling in het geheugen



Klasse DemoDocentenICT

```
public class DemoDocentenICT {  
    public static void main(String[] args) {  
        Scanner scan = new Scanner(System.in);  
        System.out.println("Hoeveel namen wil u ingeven?");  
        int aantal = scan.nextInt();  
        Docenten ict = new Docenten(aantal);  
  
        // inlezen van de namen  
        for(int i=1; i <= aantal; i++){  
            System.out.print("Voornaam: ");  
            String voornaam = scan.next();  
            scan.nextLine();  
            System.out.print("Achternaam: ");  
            String naam = scan.nextLine();  
            ict.voegDocentToe(voornaam,naam);  
        }  
  
        // teruggeven van alle initialen  
        System.out.println("Dit zijn de initialen van de docenten : ");  
        for(int i = 0; i < aantal; i++) {  
            System.out.println(ict.geefInitialenDocent(i));  
        }  
    }  
}
```

Klasse Docenten

```
public void voegDocentToe(String voornaam, String
naam){
    for(int rij = 0; rij < docenten.length ; rij++){
        if(docenten[rij][0] == null){
            docenten[rij][0] = voornaam;
            docenten[rij][1] = naam;
            return;
        }
    }
}
```

Klasse Docenten

```
    public String geefInitialenDocent (int idx){  
        String initiaal = "";  
  
        if(idx >=0 && idx < docenten.length && docenten[idx][0] != null) {  
  
            String vnaam = docenten[idx][0];  
            String famNaam = docenten[idx][1];  
  
            initiaal = "" + vnaam.charAt(0);  
  
            while ((idx = famNaam.indexOf(" ")) != -1) {  
                initiaal += " " + famNaam.charAt(0);  
                famNaam = famNaam.substring(idx + 1);  
            }  
  
            initiaal += " " + famNaam.charAt(0);  
        }  
        return initiaal;  
    }  
}
```

Klasse Docenten

Merk op!

`System.out.println` gebeurt alleen in de main methode! De andere methoden zijn onafhankelijk van hoe er met de gebruiker gecommuniceerd moet worden.

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 AppartementsGebouw

```
public class AppartementsGebouw {
    int[][][] appt;

    public AppartementsGebouw(int hoogte, int breedte, int diepte){

        appt = new int[hoogte][breedte][diepte];

    }

    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;
    }
}
```

Methode om appartementen te verhuren

```
public void voegNieuweHuurderToe(int verdieping, int apptnr,  
                                int bewoners) {  
    int diepte = appt[0][0].length;  
    int rij = apptnr / diepte;  
    int kol = (apptnr % diepte) - 1;  
  
    if (appt[verdieping][rij][kol] == 0) {  
        appt[verdieping][rij][kol] = bewoners;  
    }  
}
```

Methode om de status van alle verhuurde appartementen te geven

```
public String geefApptStatusOverzicht() {
    String overzicht = "\n";
    int hoogte = appt.length;
    int breedte = appt[0].length;
    int diepte = appt[0][0].length;

    for (int i = 0; i < hoogte; i++) {
        overzicht += "Overzicht voor verdieping_" + i + "\n";

        for (int j = 0; j < breedte; j++) {
            for (int k = 0; k < diepte; k++) {
                overzicht += "\t" + appt[i][j][k];
            }
            overzicht += "\n";
        }

        overzicht += "\n";
    }
    return overzicht;
}
```

Main methode

```
public class DemoGebouw {
    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);
        System.out.println("Geef de afmetingen van het appartement :
                            (#verdiepingen, #gangen, lengte van een gang)");

        // appartement aanmaken
        AppartementsGebouw gebouw = new AppartementsGebouw(scan.nextInt(), scan.nextInt(),
                                                            scan.nextInt());

        // bewoner toevoegen
        System.out.println("\nGeef de verdieping, het appartementnummer "
                            + "en het aantal personen dat hier komt wonen : ");

        gebouw.voegNieuweHuurderToe(scan.nextInt(), scan.nextInt(), scan.nextInt());

        //overzicht uitprinten
        System.out.println(gebouw.geefApptStatusOverzicht());
        System.out.println("In totaal wonen er " + gebouw.telAantalBewoners() +
                            " aantal personen in dit gebouw.");
    }
}
```

Werking :

```
Output - Document (run-single) ...
Geef de afmetingen van het appartement : (#verdiepingen, #gangen, lengte van een gang)
4 2 10

Geef de verdieping, het appartementnummer en het aantal personen dat hier komt wonen :
2 15·5

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    0    0    0    0    0    0

Overzicht voor verdieping_2
  0    0    0    0    0    0    0    0    0    0
  0    0    0    0    5    0    0    0    0    0

Overzicht voor verdieping_3
  0    0    0    0    0    0    0    0    0    0
  0    0    0    0    0    0    0    0    0    0

In totaal wonen er 5 aantal personen in dit gebouw.
BUILD SUCCESSFUL (total time: 14 seconds)
```