

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 : de klasse Lotto
- ✓ Arrays : declareren, instantiëren en initialiseren
- ✓ Loopen over arrays
- ✓ foreach ipv for

Arrays

- Een array is een **datastructuur**, deze kan een collectie van variabelen bijhouden en er via 1 variabelenaam naar refereren.
- Let wel een array kan alleen een collectie van homogene data bijhouden, m.a.w. elk element in de array moet van hetzelfde type zijn.
- Een array in Java is geïmplementeerd als een object
- Een array kan meerdere dimensies hebben

Voorbeeld : Lotto

Schrijf een programma in Java om de Lotto te simuleren.

Specificaties : *Lotto is een trekkingsspel van de Nationale Loterij. Bij de Lotto-trekkingen worden elke woensdag en zaterdag 6 willekeurige nummers (en een bonusnummer) uit een reeks nummers van 1 tot 45 uit de trekkingstrommel getrokken. De spelers proberen te voorspellen welke nummers uit de trommel vallen.*



Demo klasse (zonder arrays)



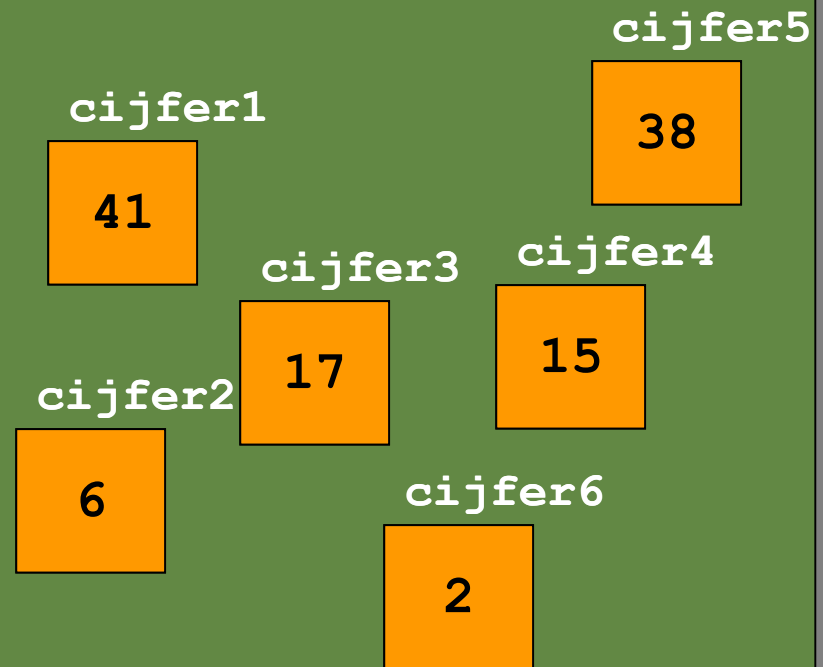
```
public class DemoLotto {  
    public static void main(String[] args) {  
  
        Scanner scan = new Scanner(System.in);  
        System.out.println(Geef je keuze voor 6 lottocijfers);  
        Lotto lotto = new Lotto(scan.nextInt(), scan.nextInt(), scan.nextInt(),  
                               scan.nextInt(), scan.nextInt(), scan.nextInt());  
        lotto.doeLottoTrekking();  
        boolean gewonnen = lotto.bepaalOfErWinstIs();  
  
        //vereenvoudigde implementatie  
        if (gewonnen) {  
            System.out.println("Gewonnen");  
        }  
        else System.out.println("Helaas, volgende week beter!");  
    }  
}
```

-> Schrijf de klasse Lotto

De klasse Lotto (zonder arrays)

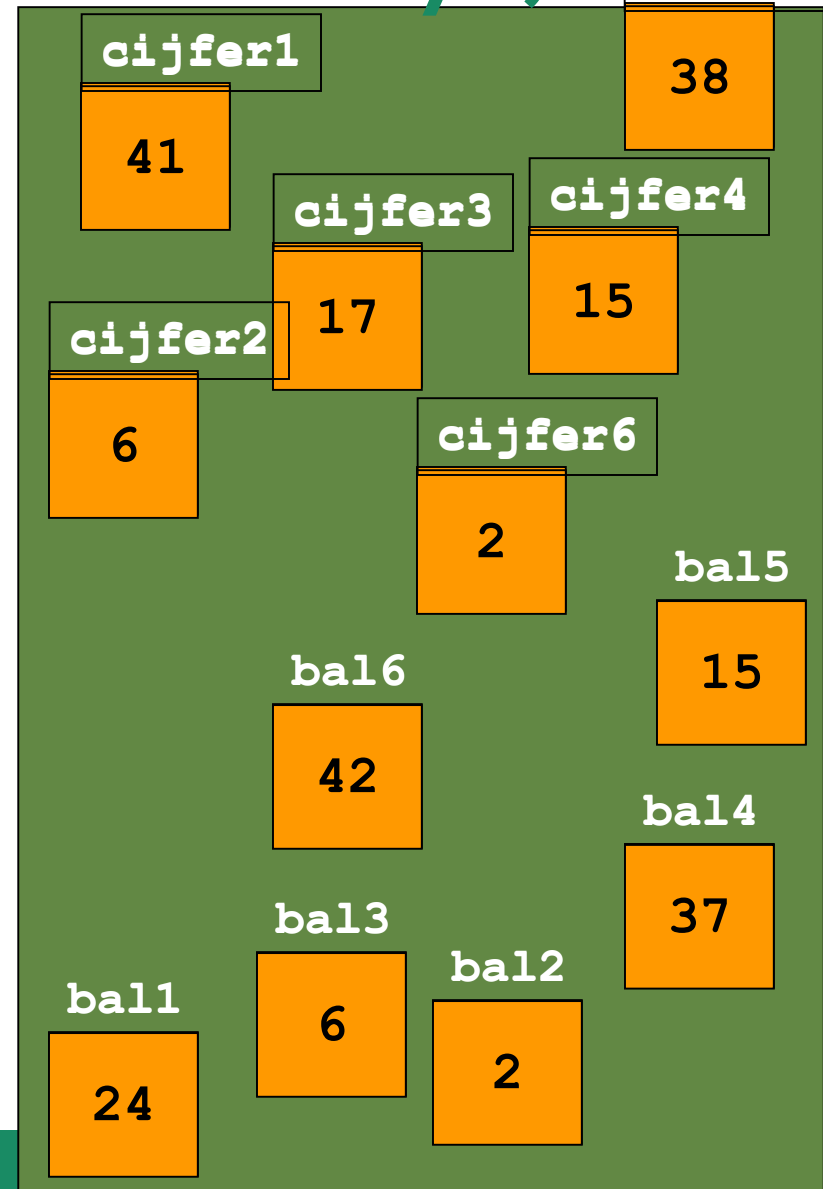
```
class Lotto {  
    int cijfer1, cijfer2, cijfer3,  
        cijfer4, cijfer5, cijfer6;  
  
    public Lotto(int keuze1, int  
        keuze2, int keuze3, int  
        keuze4, int keuze5, int  
        keuze6) {  
        cijfer1 = keuze1;  
        cijfer2 = keuze2;  
        cijfer3 = keuze3;  
        cijfer4 = keuze4;  
        cijfer5 = keuze5;  
        cijfer6 = keuze6;  
    }  
}
```

Digitaal geheugen



De klasse Lotto (zonder arrays)

```
class Lotto {  
    int cijfer1, cijfer2, cijfer3,  
        cijfer4, cijfer5, cijfer6;  
    int bal1, bal2, bal3, bal4, bal5, bal6;  
  
    public Lotto(int keuzel1, int  
        keuze2, int keuze3, int  
        keuze4, int keuze5, int  
        keuze6) {  
        cijfer1 = keuzel1;  
        cijfer2 = keuze2;  
        cijfer3 = keuze3;  
        cijfer4 = keuze4;  
        cijfer5 = keuze5;  
        cijfer6 = keuze6;  
    }  
  
    public void doeLottoTrekking() {  
        bal1 = (int) (Math.random() * 45) + 1;  
        bal2 = (int) (Math.random() * 45) + 1;  
        bal3 = (int) (Math.random() * 45) + 1;  
        bal4 = (int) (Math.random() * 45) + 1;  
        bal5 = (int) (Math.random() * 45) + 1;  
        bal6 = (int) (Math.random() * 45) + 1;  
    }  
}
```

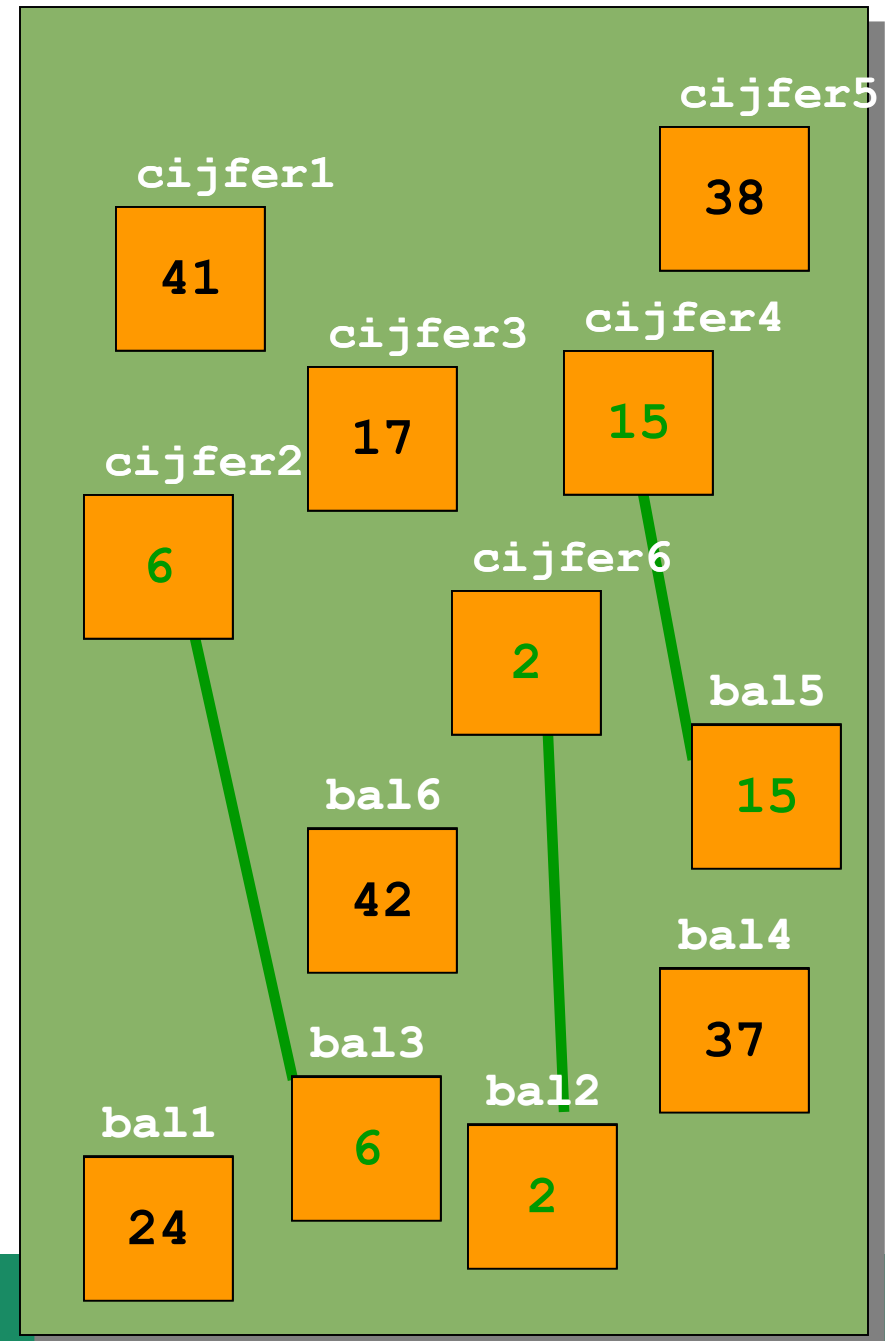


Winstbepaling

cijfer2 == bal3
cijfer4 == bal5
cijfer6 == bal2

→ 3 juiste

→ 5€ winst !



Methode winstbepaling ?

```
public boolean bepaalOfErWinstIs() {
    int aantalJuist = 0; //lokale variabele

    if (bal1 == cijfer1 || bal1 == cijfer2
        || bal1 == cijfer3 || bal1 == cijfer4
        || bal1 == cijfer5 || bal1 == cijfer6) {
        aantalJuist++;
    }

    if (bal2 == cijfer1 || bal2 == cijfer2
        || bal2 == cijfer3 || bal2 == cijfer4
        || bal2 == cijfer5 || bal2 == cijfer6) {
        aantalJuist++;
    }

    ... //doe dit voor de 6 getrokken ballen...

    return (aantalJuist >= 3);
}
```

Methode winstbepaling ? met een switch ??

```
public boolean bepaalOfErWinstIs() {  
    int aantalJuist = 0;  
  
    switch (ball) {  
        case cijfer1:  
        case cijfer2:  
        case cijfer3:  
        case cijfer4:  
        case cijfer5:  
        case cijfer6: aantalJuist++;  
            break;  
  
        default: break;  
    }  
  
    ...  
}
```

constant expression required

Wat als je 100 cijfers mag kiezen en er 25 ballen getrokken worden ?

1^e probleem: **groeidend aantal variabelen (velden):**

```
class Lotto {  
    int cijfer1, cijfer2, cijfer3, cijfer4, cijfer5,  
    cijfer6, cijfer7, cijfer8, cijfer9, cijfer10,  
    cijfer11, cijfer12, cijfer13, cijfer14, cijfer15,  
    cijfer16, cijfer17, cijfer18, cijfer19, cijfer20,  
    cijfer21, cijfer22, ...  
  
    ...  
    cijfer91, cijfer92, cijfer93, cijfer94, cijfer95  
    cijfer96, cijfer97, cijfer98, cijfer99, cijfer100;  
  
    int bal1, bal2, bal3, bal4, bal5,  
    bal6, bal7, bal8, bal9, bal10,  
    bal10, bal12, bal13, bal14, bal15,  
    bal16, bal17, bal18, bal19, ba20,  
    bal21, bal22, bal23, bal24, bal25;  
  
}
```

Wat als je 100 cijfers mag kiezen en er 25 ballen getrokken worden ?

2^e probleem: **groeïend aantal te vergelijken combinaties:**

```
public void bepaalOfErWinstIs() {  
    int aantalJuist = 0;  
  
    if (ball == cijfer1 || ball == cijfer2 ||  
        ... || ball == cijfer100) {  
        aantalJuist++;  
    }  
  
    ...  
    if (bal25 == cijfer1 || bal25 == cijfer2 || ...  
        ... || bal25 == cijfer100) {  
        aantalJuist++;  
    }  
    ...  
}
```

25 if-structuren met elk 100 te testen condities.....

Wat als je 100 cijfers mag kiezen en er 25 ballen getrokken worden ?

3^e probleem: je **code is niet onderhoudbaar !**

bij wijzigende aantallen, moet je steeds én de velden én de selectiestructuren aanpassen !!!!

aantal ballen = 6 of 25 of ...

aantal cijfers = 6 of 100 of ...

➔ TODO: vergelijk elke getrokken bal met elk gekozen cijfer

```
final int AANTAL_BALLEN = 25;
final int AANTAL_CIJFERS = 100;

for (int i = 1; i <= AANTAL_BALLEN; i++) {
    for (int j = 1; j <= AANTAL_CIJFERS; j++) {
        //vergelijk ie bal met je cijfer
    }
}
```

Oplossing met arrays

```
int[] balRij = new int[6];  
  
for (int i = 0; i < balRij.length; i++) {  
    balRij[i] = (int)(Math.random() * 45) + 1;  
}
```

- ✓ één variabele balRij voor de ganse array
- ✓ met een lus kunnen al de ballen geïnitialiseerd worden



balRij

Lotto - trekking van zaterdag 24/11/2018

•

24 26 34 42 43 45 32

Arrays : declaratie, instantiatie en initialisatie

//1) declaratie van een array

```
int[] balRij;
```

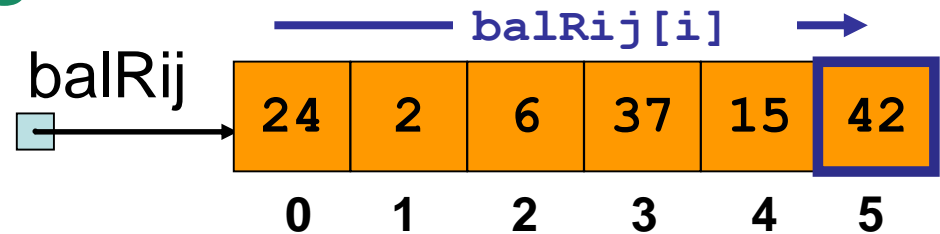
//2) instantiatie van een array

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

//3) initialisatie van de array

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

Een array als object



Lengte van de rij opvragen:

```
int lengte = balRij.length;
```

Let op **NIET: length()**
dit is geen methode
maar data

(Laatste) element van de rij opvragen:

```
int laatstGetrokkenBal = balRij[5];
```

```
int laatstGetrokkenBal = balRij[lengte - 1];
```

```
int laatstGetrokkenBal = balRij[balRij.length - 1];
```

→ Druk de elementen van de rij af:

```
for (int idx = 0; idx < balRij.length; idx++) {  
    System.out.print(balRij[idx] + " ");  
}
```


Declaratie

```
//STAP1: declaratie  
int[] balRij;
```

- naam kiezen voor je variabele (hier: balRij)
- type van de elementen vastleggen! (hier: int)
- Declaratie resulteert in een referentie, niet meer dan dat!

```
//enkele declaraties  
int[] trekking;  
char[] klinkers;  
String[] namen;
```

```
System.out.println(trekking);
```

null

```
System.out.println(trekking.length);
```

balRij



trekking



klinkers



namen



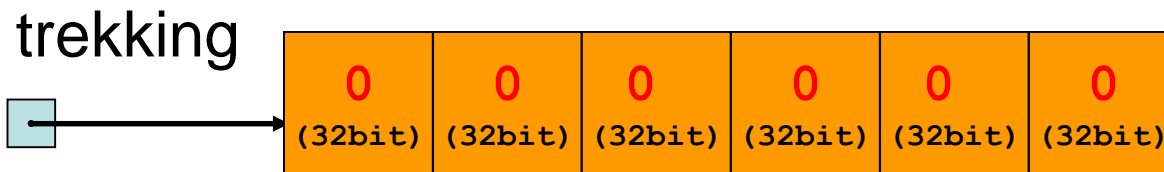
NullPointerException

Instantiatie

```
//STAP1: declaratie van een variabele  
int[] trekking;
```

```
//STAP2: instantiatie - zoek een plaats in het geheugen  
trekking = new int[6];
```

*Het aantal elementen in de array wordt vastgelegd: **6***
*Elk element in de array wordt standaard ingesteld op default (volgens type): **default voor int-waarde is 0***



```
System.out.println(trekking);
```

```
[I@3e25a5
```

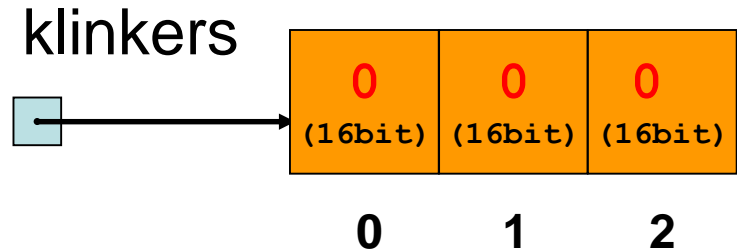
```
System.out.println(trekking.length);
```

```
6
```

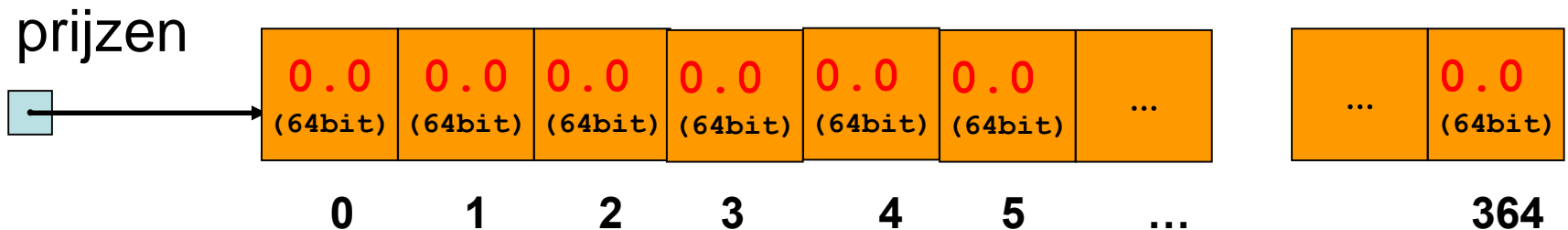
Instantiatie

```
//STAP1: declaratie  
char[] klinkers;
```

```
//STAP2: instantiatie  
klinkers = new char[3];
```



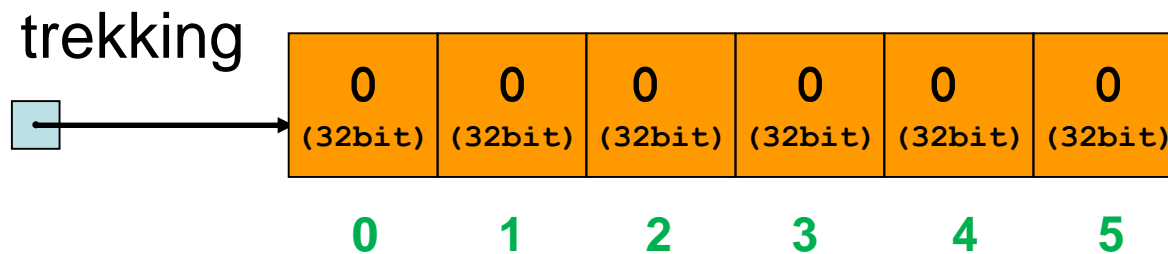
```
//STAP1+2 samen: declaratie én instantiatie  
double[] prijzen = new double[365];
```



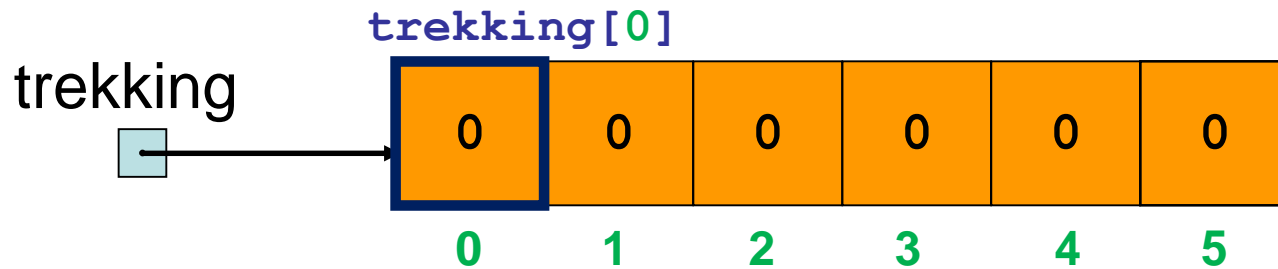
Instantiatie

```
//STAP1+2 samen: declaratie én instantiatie  
int[] trekking = new int[6];
```

- *naam* : **trekking**
- *type van de elementen*: **int**
- *aantal elementen*: **6**
- *initiële waarde elementen*: **0**
- *index van de elementen*: **0 -> 5**

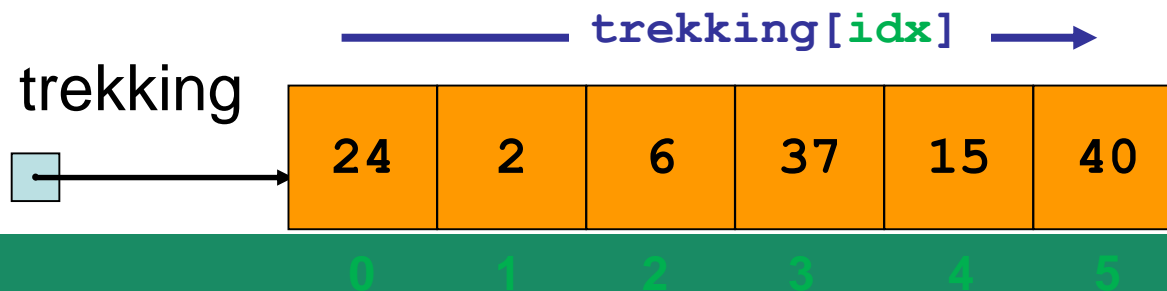


Initialisatie



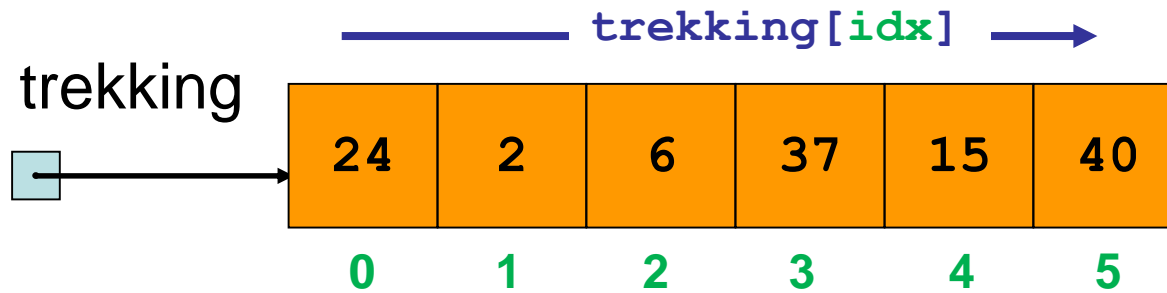
//STAP3: het opvullen van de array met concrete waarden

```
trekking[0] = (int) (Math.random() * 45) + 1;  
trekking[1] = (int) (Math.random() * 45) + 1;  
...  
trekking[5] = (int) (Math.random() * 45) + 1;
```



Initialisatie (for lus)

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



Initialisatie voorbeelden

```
//aanmaken van enkele arrays
int[] trekking = new int[6];
char[] klinkers = new char[3];
double[] prijzen = new double[365];
String[] dagen = new String[7];
```

```
//opvullen van enkele arrays
for (int idx = 0; idx < 6; idx++) {
    trekking[idx] = (int) (Math.random() * 45) + 1;
}

for (int i = 0; i < prijzen.length; i++) {
    prijzen[i] = Math.random() * 100;
}

klinkers[0] = 'a';
klinkers[1] = 'e';
klinkers[2] = 'i';

dagen[0] = "maandag";
dagen[1] = new String("dinsdag");
dagen[dagen.length - 1] = "zondag";
```

Initialisatie (initializer syntax)

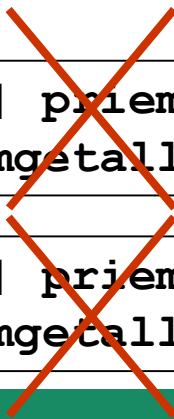
= via initializer syntax kan declaratie, instantiatie en initialisatie gebundeld worden in 1 java instructie

```
char[] klinkers = {'a', 'e', 'i', 'o', 'u'};
```

```
int[] priemgetallen = {2, 3, 5, 7, 11, 13, 17, 19};
```

```
String[] dagen = { "maandag", "dinsdag", "woensdag",  
                  "donderdag", "vrijdag",  
                  "zaterdag", "zondag"};
```

Let op : uitsplitsen in meerdere onderdelen kan niet!



```
int[] priemgetallen;  
priemgetallen = {2, 3, 5, 7, 11, 13, 17, 19};
```

```
int[] priemgetallen = new int[8];  
priemgetallen = {2, 3, 5, 7, 11, 13, 17, 19};
```


Maak en vul een array met de getallen 1 tot en met 10

→Eén voor één:

```
int[] getallen = new int[10];  
getallen[0] = 1;  
getallen[1] = 2;  
...  
getallen[9] = 10;
```

→Via een for-lus:

```
int[] getallen = new int[10];  
  
for (int i = 0; i < getallen.length; i++) {  
    getallen[i] = i+1;  
}
```

```
int[] getallen = new int[10];  
  
for (int i = 1; i <= getallen.length; i++) {  
    getallen[i] = i;  
}
```

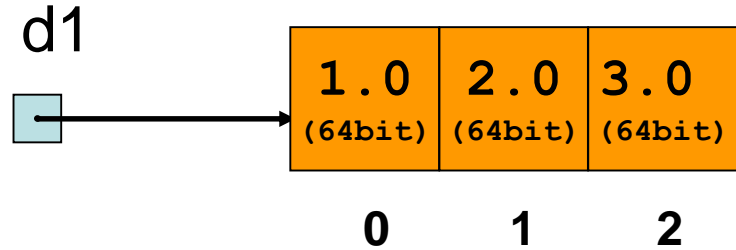
ArrayIndexOutOfBoundsException

→Via de initializer syntax:

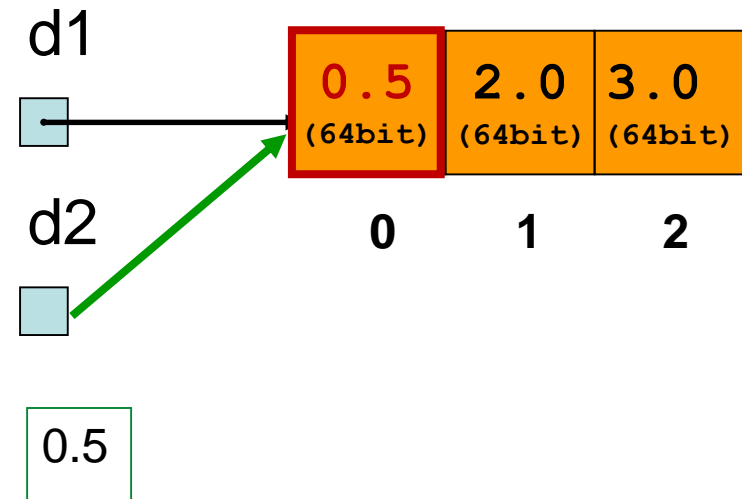
```
int[] getallen = {1,2,3,4,5,6,7,8,9,10};
```

Een array is een object

```
double[] d1 = {1.0, 2.0, 3.0};  
double[] d2;
```



```
d2 = d1;  
d2[0] /= 2;  
System.out.println(d1[0]);
```



Demo klasse (met arrays !)



```
public class DemoLotto {  
    public static void main(String[] args) {  
  
        Scanner scan = new Scanner(System.in);  
        System.out.println(Geef je keuze voor 6 lottocijfers);  
        int[] keuzes = new int[6];  
        for(int i = 0; i < 6, i++){  
            keuzes[i] = scan.nextInt();  
        }  
        Lotto lotto = new Lotto(keuzes);  
        lotto.doeLottoTrekking();  
        boolean gewonnen = lotto.bepaalOfErWinstIs();  
  
        //vereenvoudigde implementatie  
        if (gewonnen) {  
            System.out.println("Gewonnen");  
        }  
        else System.out.println("Helaas, volgende week beter!");  
    }  
}
```

De klasse Lotto (met arrays !)

```
public class Lotto {
    final int AANTAL_CIJFERS = 6;
    final int AANTAL_BALLEN = 6;
    int[] keuze;
    int[] trekking;

    public Lotto(int[] keuzes) {
        keuze = new int[AANTAL_CIJFERS];
        trekking = new int[AANTAL_BALLEN];
        for (int i = 0; i < AANTAL_CIJFERS; i++) {
            keuze[i] = keuzes[i];
        }
    }

    public void doeLottoTrekking() {
        for (int i = 0; i < AANTAL_BALLEN; i++) {
            trekking[i] = (int) (Math.random() * 45) + 1;
        }
    }

    ...
}
```

De klasse Lotto (met arrays !)

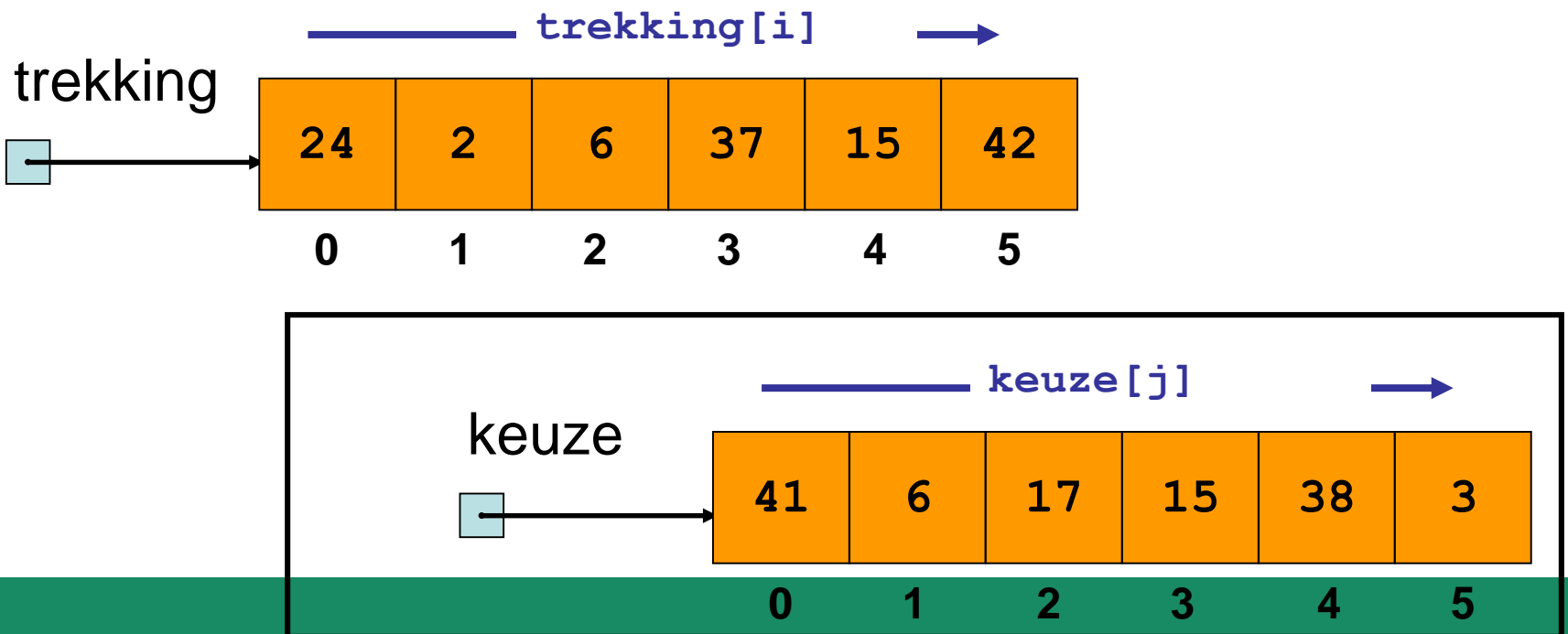
```
// vervolg
public boolean bepaalOfErWinstIs() {
    int aantalJuist = 0; //lokale variabele

    for (int i = 0; i < AANTAL_BALLEN; i++) {
        for (int j = 0; j < AANTAL_CIJFERS; j++) {
            if (trekking[i] == keuze[j]) {
                aantalJuist++;
            }
        }
    }

    return (aantalJuist >= 3);
} // einde klasse Lotto
```

Loopen over Arrays

```
for (int i = 0; i < 6; i++) {  
    for (int j = 0; j < 6; j++) {  
        if (trekking[i] == keuze[j]) {  
            aantalJuist++;  
        }  
    }  
}
```

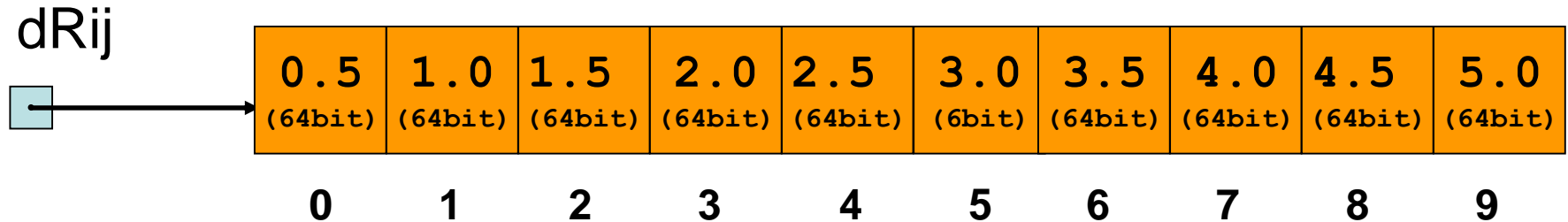


Loopen over Arrays

```
public void bepaalOfErWinstIs() {  
    int aantalJuist = 0; //lokale variabele  
  
    for (int i = 0; i < AANTAL_BALLEN; i++) {  
        for(int j = 0; j < AANTAL_CIJFERS; j++) {  
            if (trekking[i] == keuze[j]) {  
                aantalJuist++;  
  
                if (aantalJuist == 3) {  
                    return true; //verlaat hier al  
                                de methode  
                }  
            }  
        }  
  
        return false; //op dit punt zijn er geen 3  
                      // juiste keuzes gemaakt  
    }  
}
```

Maak een array van 10 reële getallen en vul op met de veelvouden van 0.5, te beginnen bij 0.5

(1x 0.5, 2x 0.5, ...)



```
//declaratie en instantiatie van de array  
double[] dRij = new double[10];
```

```
//opvullen van de array (via for lus)  
for (int i = 0; i < dRij.length; i++) {  
    dRij[i] = 0.5 * (i + 1);  
}
```

```
// Alternatief via initializer syntax:  
double[] dRij = {0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5};
```


dRij



| | | | | | | | | | |
|----------------|----------------|----------------|----------------|----------------|---------------|----------------|----------------|----------------|----------------|
| 0.5 (64bit) | 1.0 (64bit) | 1.5 (64bit) | 2.0 (64bit) | 2.5 (64bit) | 3.0 (6bit) | 3.5 (64bit) | 4.0 (64bit) | 4.5 (64bit) | 5.0 (64bit) |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

Hoe druk je de elementen van de array af (naast elkaar) ?

Uitvoer =

0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0

```
for (int i = 0; i < dRij.length; i++) {  
    System.out.print(dRij[i] + " ");  
}
```

Hoe druk je de 'even_index' elementen van de array af (naast elkaar) ?

Uitvoer =

0.5 1.5 2.5 3.5 4.5

```
for (int i = 0; i < dRij.length; i+=2) {  
    System.out.print(dRij[i] + "\t");  
}
```

Hoe druk je de 'even-waarde' elementen van de array af (naast elkaar) ?

Uitvoer =

2.0 4.0

```
for (int i = 0; i < dRij.length; i++) {  
    if (dRij[i] == (int)dRij[i] && (int)dRij[i] % 2 == 0) {  
        System.out.print(dRij[i] + "\t");  
    }  
}
```

Maak een array voor de dagen van de week en vul ze op

```
String[] dagen = new String[7]; //STAP1+STAP2
```

```
dagen[0] = "maandag";
```

```
dagen[1] = "dinsdag";
```

```
dagen[2] = "woensdag";
```

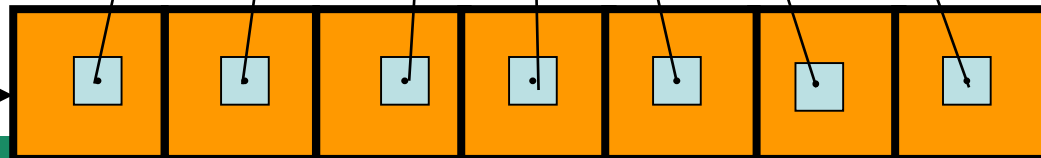
```
dagen[3] = "donderdag";
```

```
dagen[4] = "vrijdag";
```

```
dagen[5] = "zaterdag";
```

```
dagen[6] = "zondag";
```

dagen



maandag

dinsdag

woensdag

donderdag

vrijdag

zaterdag

zondag

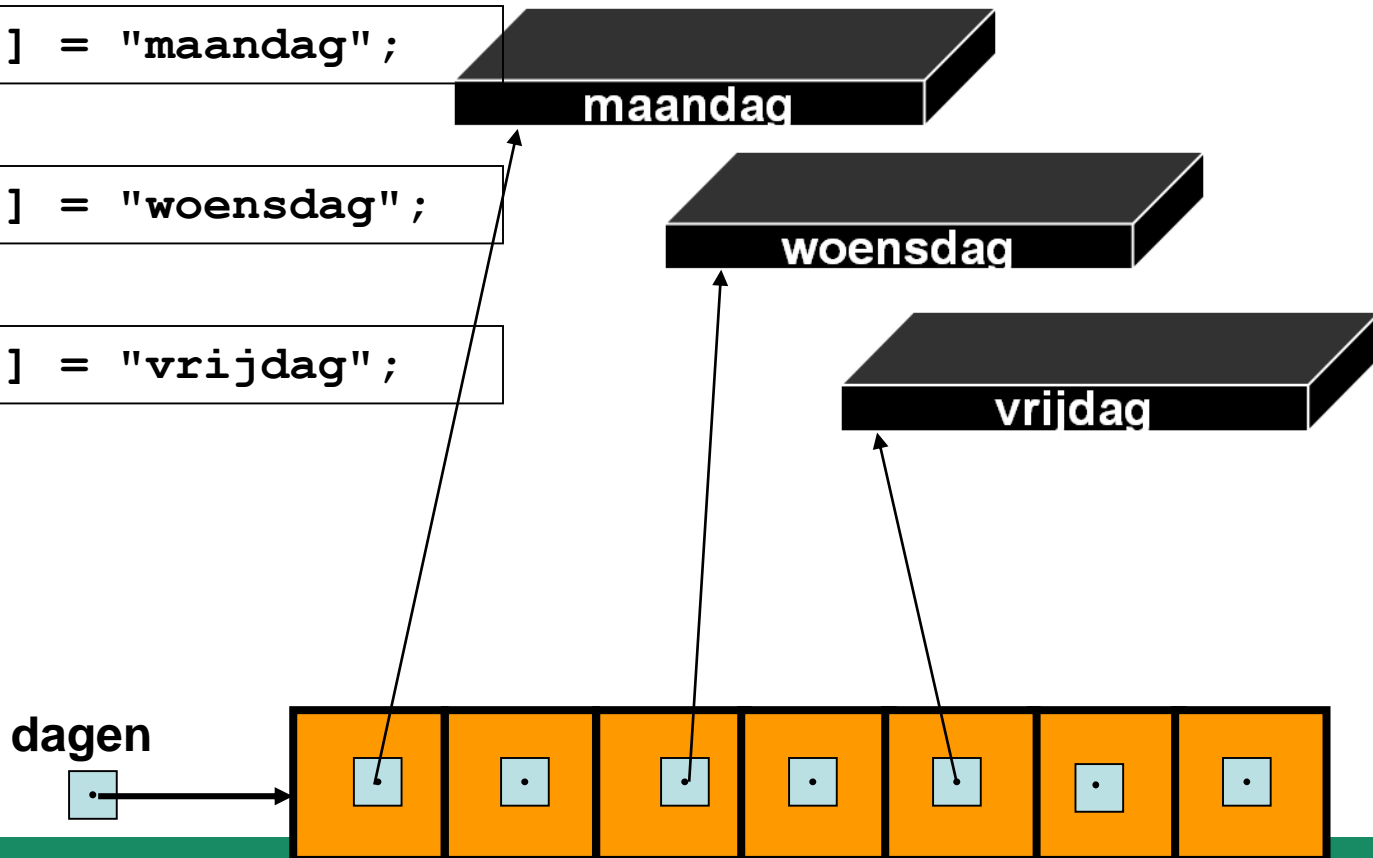
Maak een array voor de dagen van de week en vul er enkele op

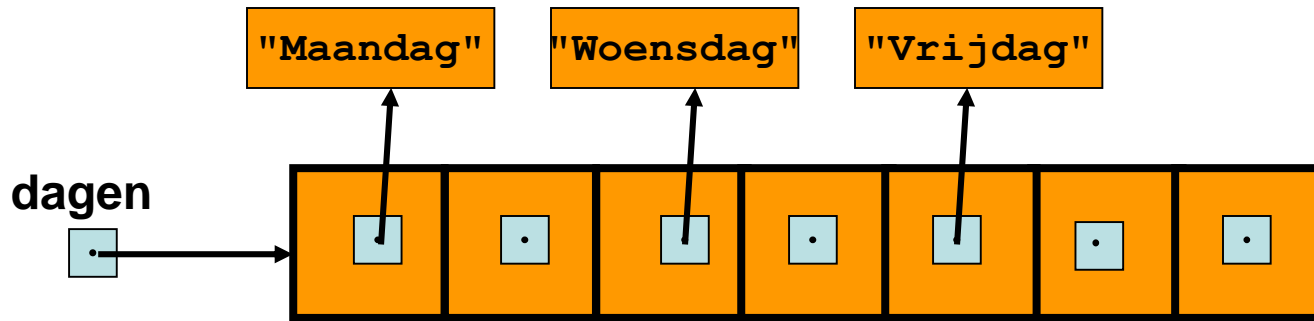
```
String[] dagen = new String[7]; //STAP1+STAP2
```

```
dagen[0] = "maandag";
```

```
dagen[2] = "woensdag";
```

```
dagen[4] = "vrijdag";
```





Druk de dagen van de week af:

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

Uitvoer:

```
Maandag  
null  
Woensdag  
null  
Vrijdag  
null  
null
```

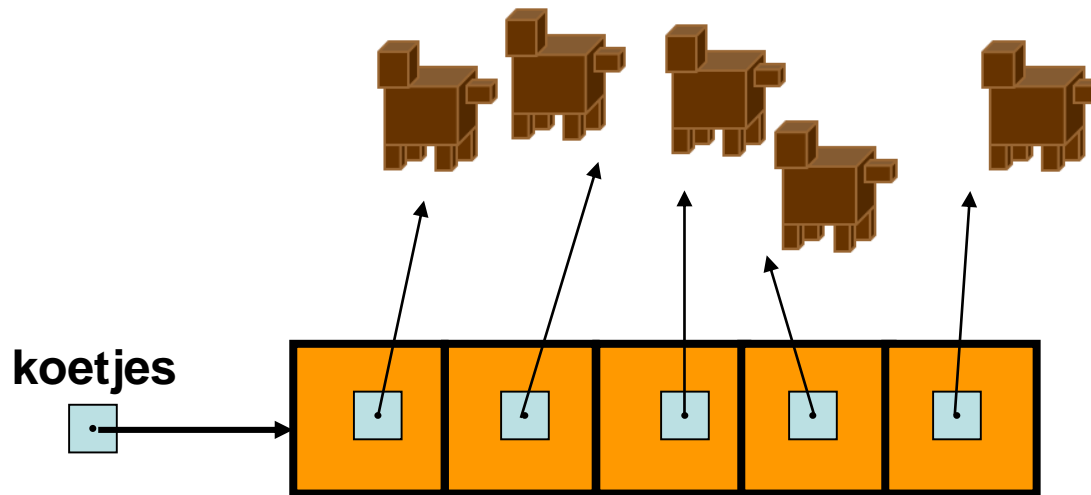
```
Maandag  
Woensdag  
Vrijdag
```

Arrays opvullen met objecten

```
Koe[] koetjes; //STAP1
```

```
koetjes = new Koe[5]; //STAP2
```

```
for (int i = 0; i < koetjes.length; i++) {  
    koetjes[i] = new Koe(); //STAP3  
}
```



lopen over elk element van de array

for <-> foreach (enhanced for loop)

```
char[] graden = {'A','B','C'};  
for(int i = 0; i < graden.length; i++)  
    // for i goes from 0 to graden.length  
    System.out.print(graden[i]);  
    // print graden[i]
```

```
for(char graad : graden)  
    //foreach (voor elke) graad in graden  
    System.out.print(graad);  
    // print de graad
```

Zie vervolg in Java OO Collections