

C# Essentials Arrays

Koen Bloemen

Sander De Puydt

Kristof Palmaers

Wim Bervoets



**DE HOGESCHOOL
MET HET NETWERK**

Elfde-Liniestraat 24, 3500 Hasselt, www.pxl.be





Arrays

- Waarom
- Eigenschappen
- Syntax
- Lussen
- Arrays en methodes
- Samenvatting

Arrays - Waarom

- Probleem met onderstaande code
 - Wat met 10 getallen?
 - 100, 1000??

```
private static void LelijkeMethod()
{
    float gemiddelde = 0.0f;
    int getal0 = 100;
    int getal1 = 50;
    int getal2 = 20;
    int getal3 = 60;
    int getal4 = 90;
    int getal5 = 80;
    gemiddelde = (getal0 + getal1 + getal2 + getal3 + getal4 + getal5) / 6.0f;
    Console.WriteLine($"Gemiddelde: {gemiddelde}");
}
```


Arrays - Waarom

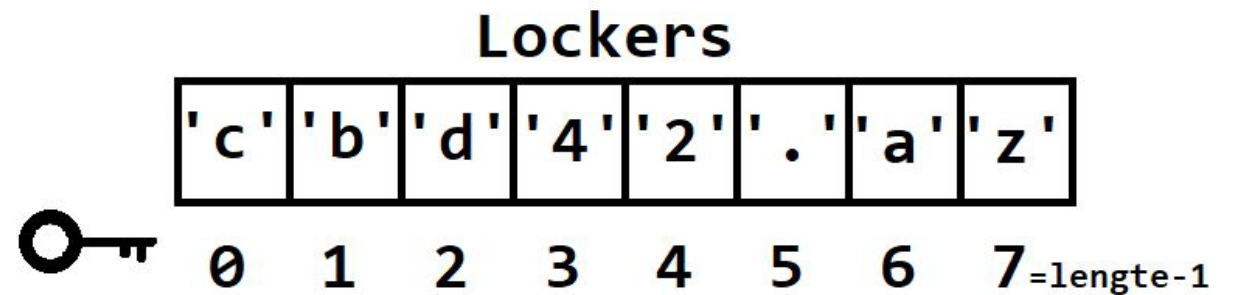
- Oplossing is arrays!

```
private static void LelijkeMethod()
{
    float gemiddelde = 0.0f;
    int getal0 = 100;
    int getal1 = 50;
    int getal2 = 20;
    int getal3 = 60;
    int getal4 = 90;
    int getal5 = 80;
    gemiddelde = (getal0 + getal1 + getal2 + getal3 + getal4 + getal5) / 6.0f;
    Console.WriteLine($"Gemiddelde: {gemiddelde}");
}
```

```
private static void MooieMethod()
{
    float gemiddelde = 0.0f;
    int[] getallen = new int[6] { 100, 50, 20, 60, 90, 80 };
    for (int i = 0; i < getallen.Length; i++)
    {
        gemiddelde += (float)getallen[i];
    }
    gemiddelde /= getallen.Length;
    Console.WriteLine($"Gemiddelde: {0}", gemiddelde);
}
```

Arrays - Eigenschappen

- Rij of reeks van gegevens van hetzelfde type
- Vaste lengte
- Zijn eigenlijk variabelen met een index



Arrays - Syntax

- **Declaratie:** vierkante haken geven aan dat het een array is

```
// declaratie van een array
int[] ditIsEenGedeclareerdeArrayVanInts;
double[] arrayVanDoubles;

// declaratie met grootte
int grootteVanArray = 4;
int[] eenLegeArrayVoorVierGetallen = new int[grootteVanArray];
```

Arrays - Syntax

- **Initialisatie:** vierkante haken geven aan dat het een array is

```
// initialisatie van een array
string[] studenten = new string[7]
                        { "Jos", "Jan", "Jef",
                          "Johan", "Jasper", "Jolanda",
                          "Jana" };

int[] scores = new int[7] { 20, 18, 16, 15, 14, 16, 18 };

// OPGELET: bij het initiëren van een array MOET je de lengte
// van de array meegeven!
```

Arrays - Syntax

- Werken met arrays: **Declaratie & Initialisatie**

Vraag 1: Welke array werkt niet?

```
public static void Main (string[] args) {  
    int[] getallen = new int[3];  
  
    int[] snelTestResultaten = new bool[];  
  
    string[] eten = new string[4] { "appel", "", "citroen"};  
  
    int[3] nummers;  
  
    double[] codes = new double[] { 2, 1, 0 };  
}
```


Arrays - Syntax

- **Uitlezen en Toewijzen:** toewijzingen van waarden doe je op basis van index

```
string[] fruit = new string[3] { "Appel", "Peer", "Kiwi"};
```

```
// Lees eerste waarde uit array fruit
```

```
Console.WriteLine(fruit[0]);
```

```
// derde waarde veranderen in een Pruim
```

```
fruit[2] = "Pruim";
```

Arrays - Syntax

- Rij of reeks van gegevens van hetzelfde type met vaste lengte

```
int[] getallen = new int[6] { 100, 50, 20, 60, 90, 80 };
```

- Zijn eigenlijk variabelen met een index
 - `getallen[3]; // 60`
 - `getallen[0]; // 100`
 - `getallen[getallen.Length - 1]; // 80`
 - `getallen[-1]; //Error`
 - Waarom?
 - `getallen[getallen.Length]; //Error`
 - Waarom?
- Gebruik index om variabele van array te lezen
 - `gemiddelde += getallen[i];`
- Of aanpassen
 - `getallen[i] = 100;`

Arrays - Lussen

- Lussen en arrays gaan hand in hand: gebruik lussen om al de waarden in een array op te vragen

```
for (int i = 0; i < getallen.Length; i++)  
{  
    Console.WriteLine(getallen[i]);  
}  
foreach (double getal in getallen)  
{  
    Console.WriteLine(getal);  
}
```

- For
 - As je (deel van) array element per element wil doorlopen
- Foreach
 - Als je **alle** array elementen wil doorlopen
 - **Enkel lezen**, niet wijzigen

Arrays - Lussen

- Werken met arrays

Vraag 2: Wat is de output?

```
public static void Main (string[] args) {  
    int[] getallen = new int[] { 0, 1, 2, 3 };  
    getallen[2] = 2;  
    getallen[1]++;  
    getallen[2] = getallen[1] + getallen[2];  
    getallen[3]--;  
    getallen[1] *= 2;  
  
    for (int i = 0; i < getallen.Length; i++){  
        Console.WriteLine(getallen[i]);  
    }  
}
```


Arrays - Lussen

- Werken met arrays

Vraag 3: Wat is de output?

By value / By reference?

```
public static void Main (string[] args) {  
    int[] getallen = new int[] { 0, 1, 2, 3 };  
    Verhoog(getallen);  
  
    for (int i = 0; i < getallen.Length; i++){  
        Console.WriteLine(getallen[i]);  
    }  
}  
  
public static void Verhoog (int[] rij){  
    rij[0]++;  
}
```

Arrays en methodes

- Array returnen uit functiemethode
 - Functiemethode gebruiken

```
int[] getallen = new int[10];  
getallen = MaakArray();
```

- De functiemethode

```
private int[] MaakArray()  
{  
    int[] nieuweGetallen = new int[10];  
    for (int i = 0; i < 10; i++)  
    {  
        nieuweGetallen[i] = i;  
    }  
    return nieuweGetallen;  
}
```

Arrays en methodes

- Array doorgeven als parameter
 - Functiemethode gebruiken

```
int resultaat = Som(getallen);
```

- De functiemethode

```
private int Som(int[] getallen)
{
    int som = 0;
    foreach (int getal in getallen)
    {
        som += getal;
    }
    return som;
}
```

Arrays en methodes

- Arrays: **Doorgeven als argument**
 - **Let op!**
 - Wanneer je een array doorgeeft als argument, dan wordt de referentie van de array doorgegeven en niet de waarde = **pass by reference**
- Gelijkaardig aan **ref** keyword van methodes

```
// het array argument werkt zoals het ref keyword
public static void Verhoog (ref int[] rij){
    rij[0]++;
}
public static void Verhoog2DoetExactHetzelfde (int[] rij){
    rij[0]++;
}
```


Arrays - Samenvatting

- Wanneer array gebruiken
 - Rij nodig van aantal gegevens
 - Hetzelfde type
 - Aantal elementen ligt vast
 - Loopen over gegevens en inlezen of overschrijven
 - For
 - Van begin tot eind
 - Van eind tot begin
 - Bepaald segment
 - Stapgrootte instellen: (i+=2 bijvoorbeeld in plaats van i++)
 - Foreach
 - Alle elementen
 - Read only

Arrays

Opdracht 1: Wat is de output?

- Schrijf een programma dat de gebruiker vraagt hoeveel uren je per weekdag C# studeert.
- Maak gebruik van een array waarin de namen zijn opgeslagen voor elke weekdag.
- Sla de uren op in een array. Vervolgens print je de uren af per weekdag.
- Maak een methode die het gemiddelde kan berekenen van het dagelijkse aantal uren.
- **Uitdaging:** Print de dag en hoeveelheid van de dag met de laagste hoeveelheid uren.