

C# Essentials Arrays

Koen Bloemen



**DE HOGESCHOOL
MET HET NETWERK**

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





- Arrays

Arrays

- 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

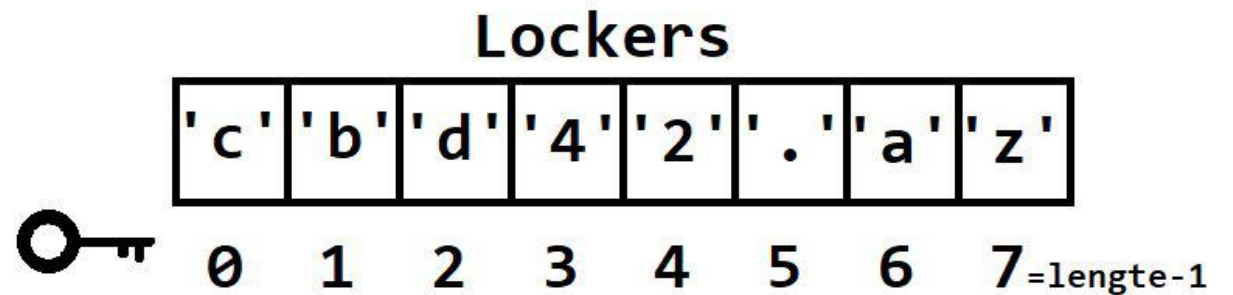
- Oplossing is arrays!

```
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

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



Arrays

- 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

- Declaratie

```
double[] getallen = new double[7]; // eerste index: 0, laatste index: 6
string[] namen = new string[9];
```

- Initialisatie

```
string[] namen = new string[5] {"Peter", "Sven", "Piet", "Tom", "Anna"};
string[] namen = new string[] {"Peter", "Sven", "Piet", "Tom", "Anna"};
string[] namen = {"Peter", "Sven", "Piet", "Tom", "Anna"};
```

- Achteraf veranderen

```
namen[0] = "Jack";
```

Arrays

- Loopen

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

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

Arrays

- Test oefening
- Zoek kleinste en grootste getal uit een reeks van getallen
- Gebruik: arrays, loops (for of foreach)
- Declareer en initialiseer op verschillende manieren

Arrays

- Array returnen uit funciemethod
 - Funciemethod gebruiken

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

- De funciemethod

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

Arrays

- Array doorgeven als parameter
 - Functiemethod gebruiken

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

- De functiemethod

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

- Analooq voor void method
- Gebruik ref om array aan te passen binnen method

Arrays

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

Arrays

- Test oefening
- Zoek kleinste en grootste getal uit een reeks van getallen
- Pas de vorige oefening aan:
 - Array als parameter
 - Return array van kleinste en grootste getal