

Hand-out

Programmeren

08. Arrays

1. Inleiding

In dit hoofdstuk gaan we werken met arrays. Arrays zijn vaak een lastig onderwerp omdat het altijd even duurt voor je ze begrijpt. In deze hand-out leggen we de Array in zijn algemene vorm uit, d.w.z. zodat je ze strakjes ook in een andere taal kunt toepassen. Je hebt bijvoorbeeld geen associatieve arrays in C#, hier voor gebruik je een List object. Het List object wordt in deze hand-out nog niet besproken omdat dit een vrij complex onderwerp is.

2. Wat is een Array

Een array bestaat uit een of meerdere kolommen met variabelen van het zelfde type. Een rij geef je aan met blokhaken []. Een array is in principe een verzameling of collectie van gegevens.

0	1	2	3	4	5	6	7	8	9
22	56	100	5	70	86	23	9	90	15

De bovenste rij is de positie van de array en de onderste rij is de waarde.

LET OP : Arrays beginnen altijd te tellen bij 0, index 1 is dus 0.

3. Hoe maak je een Array

Een array kent drie soorten :

1. Geïndexeerde array
2. Associatieve array
3. Multidimensionale array

In het bovenste voorbeeld zie je een geïndexeerde array, de positie van de array word aangegeven via een integer. Dit noemen we de index van het array element.

Het maken van een geïndexeerde array is eenvoudig :

```
1. int[] intArray = {22,56,23,78,28,700,376,28,20};  
2. Console.WriteLine(intArray[3]) // 78;
```

In een associatieve array gebruiken we ook een index, echter gebruiken we hier geen integer maar een string om aan te geven welk array element we willen ophalen. Vandaar dus associatief, want we kunnen later associëren met een naam of noemer.

id	name	age	gender	color	food
"22"	"boy"	36	"m"	"red"	"wafels"

C# kent geen vorm voor associatieve arrays op de manier als hier boven, om associatieve arrays te maken gebruiken we het List object, meer daarover later.

LET OP : We hoeven niet perse een array direct met de inhoud van de elementen te definiëren, we kunnen ook lege ruimte reserveren per element, dit doen we als volgt :

```
1. int[] intArray = new int[10]; //reserveer 10 lege elementen
2. Console.WriteLine(intArray[3]) // 0;
```

Op deze manier hebben we dus 10 lege array elementen gereserveerd die we later in kunnen gaan vullen

4. Data types

Zoals je hier boven hebt gezien kunnen we dus een array van integers aanmaken via het datatype int plus daar blokhaakjes achter te zetten []. Dat wil zeggen, maak een array van integers. En dit kan ook met alle andere types, bijvoorbeeld :

```
1. int[] intArray = {22,56,23,78,28,700,376,28,20};
2. string[] strArray = {"een", "twee", "drie"};
3. bool[] blnArray = {true, 0, 1, true};
4. char[] chrArray = {'A','B','x','y'}
```

5. Loopen met een array

Een array is handig omdat het een collectie aan elementen bevat, en hier kunnen we vervolgens ook doorheen lopen via een loop. We zeggen dan "voor ieder element in de array,... doe iets...".

```
1. int[] intArray = new int[10]; //reserveer 10 lege elementen
2.
3. foreach(int intResult in intArray)
4. {
5.     Console.WriteLine(intResult) // 0;
6. }
```

Een voorbeeld met een while loop :

```
1. int[] intNumbers = {23,78,36,90,548,58,23};
2. int intIndex = 1;
3. int intLengthArray = intNumbers.Length;
4.
5. while(intIndex < intLengthArray)
6. {
7.     Console.WriteLine("Nummer : {0}", intNumbers[intIndex].ToString());
8. }
```

Een voorbeeld met een for loop :

```
1. int[] intNumbers = {23,78,36,90,548,58,23};
2.
3. for(int i = 0; i < intNumbers.Length;i++)
4. {
5.     Console.WriteLine("Nummer : {0}", intNumbers[i].ToString());
6. }
```

6. Een Array sorteren

We kunnen de elementen in een array natuurlijk ook rangschikken. Als volgt :

```
1. int[] intArray = {22,56,23,78,28,700,376,28,20};  
2. Array.Sort(intArray);
```

7. Problemen

Het werken met arrays kan de volgende problemen opleveren :

- Je moet van tevoren de grootte van de array opgeven. Als je meer elementen hebt die je in een array wilt opslaan, dan moet je de array eerst vergroten. Dit doe je als volgt :

```
1. int[] intArray = {22,56,23,78,28,700,376,28,20};  
2. Array.Resize(ref intArray, 30);
```

8. Demo

In dit stuk gaan we klassikaal een demo maken. Het is de bedoeling dat je mee doet en zo stapsgewijs houvast krijgt op het onderwerp, maak daar waar nodig is aantekeningen in je schrift of in je kladblok. Als je klaar bent ga je zelf aan de slag dus let goed op!

9. Opdracht

Maak een programma dat alle namen (voor en achter) van je klasgenoten op het scherm laat zien met daarbij het bijbehorende index nummer.

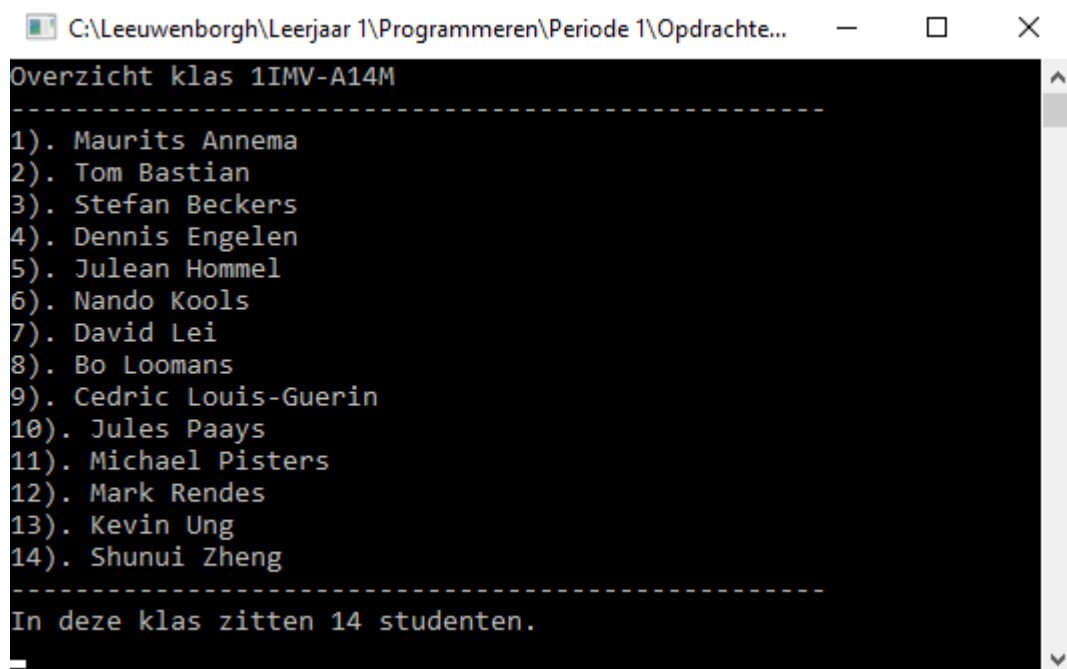
De programma eisen zijn :

- De namen moeten in een array staan.
- De loop waarin je de array op het scherm toont moet een for loop zijn.
- Sorteer de namen op alfabetische volgorde via de Sort() methode.

Extra punten kunnen worden verdiend als je :

- Het aantal leerlingen in de klas onder aan je resultaat op het scherm zet, bijvoorbeeld "In deze klas zitten x studenten". Het aantal mag alleen niet hard in de code staan en moet dus uit een variabele komen, welke dat is bepaal je zelf. Er is er echter maar een die hier voor wordt goedgekeurd.

Als je klaar bent met deze opdracht ga je je werk samen met de docent bekijken. veel succes!



```
C:\Leeuwenborgh\Leerjaar 1\Programmeren\Periode 1\Opdrachte...
Overzicht klas 1IMV-A14M
-----
1). Maurits Annema
2). Tom Bastian
3). Stefan Beckers
4). Dennis Engelen
5). Julean Hommel
6). Nando Kools
7). David Lei
8). Bo Loomans
9). Cedric Louis-Guerin
10). Jules Paays
11). Michael Pisters
12). Mark Rendes
13). Kevin Ung
14). Shunui Zheng
-----
In deze klas zitten 14 studenten.
```

De code kun je de volgende les vinden op GitHub (rep 08.-Arrays)