

C# Essentials Generic lists

Koen Bloemen



**DE HOGESCHOOL
MET HET NETWERK**

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





- Generic lists

Generic lists

- Nadelen van arrays
 - Lengte ligt vast
 - Waarde toevoegen of verwijderen:
 - Kopie maken van array naar nieuwe array met nieuwe grootte
 - Elke waarde opschuiven
- Oplossing
 - Gebruik `List<T>` uit `System.Collections.Generic`
 - Kan je extra elementen aan toevoegen.
 - Gebruikt inwendig een array. Wanneer volle capaciteit \Rightarrow groeien
 - Is generiek
 - Je kan het type `T` kiezen
 - Als je zelf een generieke class of method maakt \Rightarrow geen aparte versie voor elk datatype nodig!
 - Heeft methods en properties beschikbaar

Generic lists

- Declaratie en initialisatie

```
// Declaratie
List<string> studenten;
// Initialisatie
studenten = new List<string>();

// In één regel
List<int> leeftijd = new List<int>();

// Initialiseren met array
int[] nummers = new int[3] { 1, 2, 3 };
List<int> nummersList = new List<int>(nummers);
```

Generic lists

Add(T element)	Voegt element toe aan einde van List
AddRange(collection)	List/array toevoegen aan andere List
Clear()	Wist alle elementen van List
Count	Telt het aantal element in List (property)
ToArray()	Kopieert List naar nieuwe 1D array
GetRange(int startIndex, int aantal)	Maakt kopie van stukje van de lijst
IndexOf(T element) IndexOf(T element, int startIndex) IndexOf(T element, int startIndex, int aantal)	Zoekt de index van een element uit List
Insert(int index, T element)	Voegt element toe op bepaalde index, de elementen erna schuiven 1 plaats op
Remove(T item)	Verwijdert element met bepaalde waarde
RemoveAt(int index)	Verwijdert element op bepaalde index
Reverse() Reverse(int startIndex, int aantal)	draait de volgorde van de elementen om
Sort()	Sorteert de elementen in de List

Generic lists

- Operator
 - [] : om elementen op te zoeken of aan te passen (zoals in array)

```
int[] nummers = new int[3] { 1, 2, 3 };  
List<int> nummersList = new List<int>(nummers);  
int getal = nummersList[2];
```

Generic lists

- Elementen toevoegen aan een list

```
List<int> waarden = new List<int>{ 10, 20, 30, 40, 50, 60, 70, 80 };  
// Waarden toevoegen aan List  
waarden.Add(90);  
waarden.Add(100);
```

Generic lists

- AddRange() om array (of zelfs andere List) toe te voegen aan de List

```
int[] getallen = new int[] { 10, 20, 30, 40, 50, 60, 70, 80 };  
List<int> waarden = new List<int>();  
// Voeg een array toe aan de List  
waarden.AddRange(getallen);
```


Generic lists

- Elementen inserten in List

```
int[] getallen = new int[]{ 10, 20, 30, 40, 50, 60, 70, 80 };  
List<int> waarden = new List<int>();  
// Voeg een array toe aan de List  
waarden.AddRange(getallen);  
// Voeg waarde 1000 toe op index 3 van de List (indices starten vanaf 0!!!)  
// De waarden erna schuiven dan 1 plaats op  
waarden.Insert(3, 1000);
```

- Element verwijderen uit List

```
// Verwijder element op index 3 van de List  
waarden.RemoveAt(3);  
// OF verwijder element met waarde 1000 (die als eerste voorkomt!) uit de List  
waarden.Remove(1000);
```

Generic lists

- List omzetten naar array

```
int[] getallen = new int[]{ 10, 20, 30, 40, 50, 60, 70, 80 };  
List<int> waarden = new List<int>();  
// Voeg een array toe aan de List  
waarden.AddRange(getallen);  
// List omzetten naar array  
int[] waardenArray = waarden.ToArray();
```

- Stukje van list omzetten naar array

```
// Neem enkel het stukje van de list:  
// - vanaf index 2  
// - 3 elementen als aantal (lengte)  
int[] waardenArray = waarden.GetRange(2, 3).ToArray(); // 30, 40, 50
```

Generic lists

- List sorteren

```
string[] voornamen = { "Wouter", "Paul", "Andreas", "Niels", "Kathleen",  
"Paul", "Silvia", "Patricia"};  
List<string> waarden = new List<string>(voornamen);  
  
// Waarden sorteren van klein naar groot  
waarden.Sort();  
  
// Waarden sorteren van groot naar klein  
waarden.Sort();  
waarden.Reverse();
```

Generic lists

- Zoeken naar index in list

```
string[] voornamen = { "Wouter", "Paul", "Andreas", "Niels", "Kathleen", "Paul",  
"Silvia", "Patricia"};  
List<string> waarden = new List<string>(voornamen);  
  
waarden.IndexOf("Lise"); // => -1: niet gevonden  
waarden.IndexOf("Paul", 3); // vanaf index 3 zoeken => 5 is het eerste voorkomen  
waarden.IndexOf("Paul", 2, 3); // vanaf index 2 zoeken, aantal elementen: 3 => -1  
waarden.IndexOf("Paul", 2, 4); // vanaf index 2 zoeken, aantal elementen: 4 => 5
```