

Oefening Custom Generic Class

Maak een Console App en schrijf hierin een generieke klasse **GenericList<T>**, die een lijst met elementen van type T bevat.

Sla de lijst met elementen op in een array met een beperkte capaciteit die wordt doorgegeven als parameter van de constructor van de klasse.

Voeg methoden toe om een item toe te voegen, om een item per index **op te vragen (via indexer)**, om een item per index te verwijderen, om een item op een gegeven positie in te **voegen(via indexer)**, om de lijst te wissen. Gebruik, indien nodig, methode `Resize` om de interne array `T[] elements` om de array van grootte te wijzigen

Voeg een methode toe om een item op waarde te zoeken (**bool teruggeven of de waarde zich al dan niet bevindt in de lijst**) en maak een override van de methode `ToString()` die de gegevens uit de lijst tekstueel uitschrijft naar de console (elk item in de lijst op nieuwe lijn).

Gebruik de volgende Code om de console app te testen :

```
class Program
{
    static void Main()
    {
        GenericList<int> elements = new GenericList<int>();

        // Lege lijst
        Console.WriteLine(elements);
        Console.WriteLine("Aantal: {0}", elements.Count);
        Console.WriteLine("Capaciteit: {0}", elements.Capacity);

        // Auto-grow functionaliteit
        elements = new GenericList<int>(3);
        elements.Add(1);
        elements.Add(2);
        elements.Add(3);
        elements.Add(4);

        Console.WriteLine("\n" + elements);
        Console.WriteLine("Aantal: {0}", elements.Count);
        Console.WriteLine("Capaciteit: {0}", elements.Capacity);

        // Insert, RemoveAt
        elements.Clear();

        elements.Insert(0, 4);
        elements.Insert(0, 3);
        elements.Insert(0, 2);
    }
}
```

```

elements.Insert(0, 1);

elements.RemoveAt(0);
elements.RemoveAt(elements.Count - 1);

Console.WriteLine("\n" + elements);
Console.WriteLine("Aantal: {0}", elements.Count);
Console.WriteLine("Capaciteit: {0}", elements.Capacity);

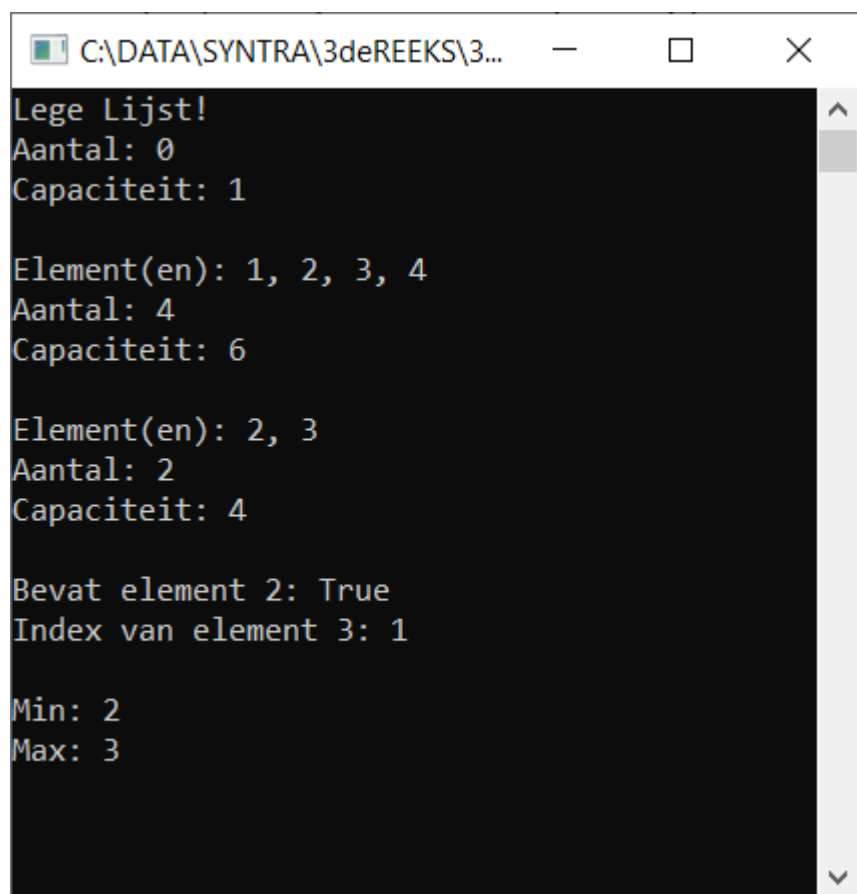
// Contains, IndexOf
Console.WriteLine("\nBevat element 2: {0}", elements.Contains(2));
Console.WriteLine("Index van element 3: {0}", elements.IndexOf(3));

// Max, Min
Console.WriteLine("\nMin: {0}", elements.Min());
Console.WriteLine("Max: {0}", elements.Max());

Console.ReadKey();
}
}

```

Gewenst resultaat:



```
C:\DATA\SYNTRA\3deREEKS\3...  
Lege Lijst!  
Aantal: 0  
Capaciteit: 1  
  
Element(en): 1, 2, 3, 4  
Aantal: 4  
Capaciteit: 6  
  
Element(en): 2, 3  
Aantal: 2  
Capaciteit: 4  
  
Bevat element 2: True  
Index van element 3: 1  
  
Min: 2  
Max: 3
```

Vul de Generic class `GenericList<T>` aan:

```
public class GenericList<T> where T : IComparable
{
    // Constant Fields
    private const int CapacityByDefault = 1;

    private const string Separator = ", ";

    // Array of elements
    private T[] elements;

    // Constructor
    public GenericList(int capacity = CapacityByDefault)
    {
        this.Count = 0;
        this.Capacity = capacity;
        this.elements = new T[capacity];
    }

    // Properties
    public int Count { get; private set; }

    public int Capacity { get; private set; }

    public T this[int index]
    {
        get
        {
            if (index < 0 || index >= this.Count)
                throw new IndexOutOfRangeException("Index is buiten range!");

            return this.elements[index];
        }
        set
        {
            this.elements[index] = value;
        }
    }

    public override string ToString() {
        if (this.Count == 0)
            return "Lege Lijst!";
        StringBuilder result = new StringBuilder();
        result.Append("Element(en): ");
        for (int i = 0; i < this.Count; i++) {
            result.AppendFormat("{0}", this.elements[i].ToString());
        }
    }
}
```

```

        if (i + 1 < this.Count)
            result.Append(Separator);
    }

    return result.ToString();
}

// Private methods
private void Resize(int capacity)
{
    if (capacity > this.Capacity)
    {
        this.Capacity *= 2;
        Array.Resize(ref this.elements, this.Capacity);
    }
}

//...
}

```