(T12)討論DefaultIfEmpty、FirstOrDefault、LastOrDefault、ElementAtOrDefault、SingleOrDefault  
CourseGUID: 5ba9a6fe-7475-4b0c-8b99-bbcf7f5e2e1c  
=======================================================================  
(T12)討論DefaultIfEmpty、FirstOrDefault、LastOrDefault、ElementAtOrDefault、SingleOrDefault  
=======================================================================  
0. Summary

-----------

1. New Project

1.1. Create New Project : Sample

-----------

2. Sample : Program.cs  
=======================================================================

0. Summary

1.

First, FirstOrDefault, Last, LastOrDefault, ElementAt, ElementAtOrDefault, Single, SingleOrDefault, and DefaultIfEmpty are Element Operators which retrieve a single element from a sequence.  All of these have their own overloading methods.

-----------------------------------------------

2.

First V.S. FirstOrDefault

---------------------------------

2.1.

First V.S. FirstOrDefault

--------------

2.1.1.

First

throws InvalidOperationException

if the sequence does not contain any elements

or if no element in the sequence satisfies the condition

--------------

2.1.2.

FirstOrDefault

returns default value of type

if the sequence does not contain any elements

or if no element in the sequence satisfies the condition

For the reference type, the default value is null.

---------------------------------

2.2.

Enumerable.First<TSource>

(this IEnumerable<TSource> source)

OR

Enumerable.FirstOrDefault<TSource>

(this IEnumerable<TSource> source)

Reference:

[https://msdn.microsoft.com/en-us/library/bb291976(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb291976%28v=vs.110%29.aspx)

[https://msdn.microsoft.com/en-us/library/bb340482(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb340482%28v=vs.110%29.aspx)

Returns the first element of a sequence.

---------------------------------

2.3.

Enumerable.First<TSource>

(this IEnumerable<TSource> source, Func<TSource, Boolean> predicate)

OR

Enumerable.FirstOrDefault<TSource>

(this IEnumerable<TSource> source, Func<TSource, Boolean> predicate)

Reference:

[https://msdn.microsoft.com/en-us/library/bb535050(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb535050%28v=vs.110%29.aspx)

[https://msdn.microsoft.com/en-us/library/bb549039(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb549039%28v=vs.110%29.aspx)

Returns the first element in a sequence that satisfies a specified condition.

-----------------------------------------------

3.

Last V.S. LastOrDefault

---------------------------------

3.1.

Last V.S. LastOrDefault

--------------

3.1.1.

Last

throws InvalidOperationException

if the sequence does not contain any elements

or if no element in the sequence satisfies the condition

--------------

3.1.2.

LastOrDefault

returns default value of type

if the sequence does not contain any elements

or if no element in the sequence satisfies the condition

For the reference type, the default value is null.

---------------------------------

3.2.

Enumerable.Last<TSource>

(this IEnumerable<TSource> source)

OR

Enumerable.LastOrDefault<TSource>

(this IEnumerable<TSource> source)

Reference:

[https://msdn.microsoft.com/en-us/library/bb358775(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb358775%28v=vs.110%29.aspx)

[https://msdn.microsoft.com/en-us/library/bb301849(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb301849%28v=vs.110%29.aspx)

Returns the last element of a sequence.

---------------------------------

3.3.

Enumerable.Last<TSource>

(this IEnumerable<TSource> source, Func<TSource, Boolean> predicate)

OR

Enumerable.LastOrDefault<TSource>

(this IEnumerable<TSource> source, Func<TSource, Boolean> predicate)

Reference:

[https://msdn.microsoft.com/en-us/library/bb549138(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb549138%28v=vs.110%29.aspx)

[https://msdn.microsoft.com/en-us/library/bb548915(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb548915%28v=vs.110%29.aspx)

Returns the last element of a sequence that satisfies a specified condition.

-----------------------------------------------

4.

ElementAt V.S. ElementAtOrDefault

---------------------------------

4.1.

Enumerable.ElementAt<TSource>

(this IEnumerable<TSource> source, Int32 index)

Reference:

[https://msdn.microsoft.com/en-us/library/bb299233(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb299233%28v=vs.110%29.aspx)

Returns the element at a specified index in a sequence.

Throws ArgumentNullException if source is null.

Throws ArgumentOutOfRangeException

if index is less than 0 or greater than or equal to the number of elements in source.

---------------------------------

4.2.

Enumerable.ElementAtOrDefault<TSource>

(this IEnumerable<TSource> source, Int32 index)

Reference:

[https://msdn.microsoft.com/en-us/library/bb494386(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb494386%28v=vs.110%29.aspx)

Returns the element at a specified index in a sequence

or a default value if the index is out of range.

Throws ArgumentNullException if source is null.

-----------------------------------------------

5.

Single V.S. SingleOrDefault

---------------------------------

5.1.

Single V.S. SingleOrDefault

--------------

5.1.1.

Single

throws ArgumentNullException if source is null.

Throws InvalidOperationException

if the input sequence contains more than one element.

or if The input sequence is empty.

--------------

5.1.2.

SingleOrDefault

throws ArgumentNullException if source is null.

Throws InvalidOperationException

if the input sequence contains more than one element.

---------------------------------

5.2.

--------------

5.2.1.

Enumerable.Single<TSource>

(this IEnumerable<TSource> source)

Reference:

[https://msdn.microsoft.com/en-us/library/bb155325(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb155325%28v=vs.110%29.aspx)

Returns the only element of a sequence,

and throws an exception

if there is not exactly one element in the sequence.

--------------

5.2.2.

Enumerable.SingleOrDefault<TSource>

(this IEnumerable<TSource> source)

Reference:

[https://msdn.microsoft.com/en-us/library/bb342451(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb342451%28v=vs.110%29.aspx)

Returns the only element of a sequence,

or a default value if the sequence is empty;

this method throws an exception

if there is more than one element in the sequence.

---------------------------------

5.3.

--------------

5.3.1.

Enumerable.Single<TSource>

(this IEnumerable<TSource> source, Func<TSource, Boolean> predicate)

Reference:

[https://msdn.microsoft.com/en-us/library/bb535118(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb535118%28v=vs.110%29.aspx)

Returns the only element of a sequence that satisfies a specified condition,

and throws an exception if more than one such element exists.

--------------

5.3.2.

Enumerable.SingleOrDefault<TSource>

(this IEnumerable<TSource> source, Func<TSource, Boolean> predicate)

Reference:

[https://msdn.microsoft.com/en-us/library/bb549274(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb549274%28v=vs.110%29.aspx)

Returns the only element of a sequence that satisfies a specified condition

or a default value if no such element exists;

this method throws an exception

if more than one element satisfies the condition.

-----------------------------------------------

6.

DefaultIfEmpty

---------------------------------

6.1.

Enumerable.DefaultIfEmpty<TSource>

(this IEnumerable<TSource> source)

Reference:

[https://msdn.microsoft.com/en-us/library/bb360179(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb360179%28v=vs.110%29.aspx)

Returns the elements of the specified sequence

or the type parameter's default value in a singleton collection

if the sequence is empty.

---------------------------------

6.2.

Enumerable.DefaultIfEmpty<TSource>

(this IEnumerable<TSource> source, TSource defaultValue)

Reference:

[https://msdn.microsoft.com/en-us/library/bb355419(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb355419%28v=vs.110%29.aspx)

Returns the elements of the specified sequence

or the specified value in a singleton collection

if the sequence is empty.

=============================================

1. New Project

1.1. Create New Project : Sample

File --> New --> Project... -->

Visual C# -->  **Console App** **(.Net Framework)** -->

Name: **Sample**

Graphical user interface, application, email

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, Excel

Description automatically generated

=============================================

2. Sample : Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

namespace Sample

{

    class Program

    {

        static void Main(string[] args)

        {

            // 1. ======================================

            //FirstSample()

            Console.WriteLine("1. FirstSample(); ================== ");

            FirstSample();

            // 2. ======================================

            //FirstOrDefaultSample()

            Console.WriteLine("2. FirstOrDefaultSample(); ================== ");

            FirstOrDefaultSample();

            //Last() is similar to First(),

            //Last() return last item,

            //First() return first item.

            //LastOrDefault() is similar to FirstOrDefault(),

            //LastOrDefault() return last item or

            //default value of type if last item was not found.

            //FirstOrDefault() return first item or

            //default value of type if first item was not found.

            // 3. ======================================

            //ElementAt\_ElementAtOrDefaultSample()

            Console.WriteLine("3. ElementAt\_ElementAtOrDefaultSample(); ================== ");

            ElementAt\_ElementAtOrDefaultSample();

            // 4. ======================================

            //Single\_SingleOrDefaultSample()

            Console.WriteLine("4. Single\_SingleOrDefaultSample(); ================== ");

            Single\_SingleOrDefaultSample();

            // 5. ======================================

            //DefaultIfEmptySample()

            Console.WriteLine("5. DefaultIfEmptySample(); ================== ");

            DefaultIfEmptySample();

            Console.ReadLine();

        }

       // 1. ======================================

        //FirstSample()

        static void FirstSample()

        {

            // 2.

            // First V.S. FirstOrDefault

            // ---------------------------------

            // 2.1.

            // First V.S. FirstOrDefault

            // --------------

            // 2.1.1.

            // First

            // throws InvalidOperationException

            // if the sequence does not contain any elements

            // or if no element in the sequence satisfies the condition

            // --------------

            // 2.1.2.

            // FirstOrDefault

            // returns default value of type

            // if the sequence does not contain any elements

            // or if no element in the sequence satisfies the condition

            // For the reference type, the default value is null.

            // ---------------------------------

            // 2.2.

            // Enumerable.First<TSource>

            // (this IEnumerable<TSource> source)

            // OR

            // Enumerable.FirstOrDefault<TSource>

            // (this IEnumerable<TSource> source)

            // Reference:

            // [https://msdn.microsoft.com/en-us/library/bb291976(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb291976%28v=vs.110%29.aspx)

            // [https://msdn.microsoft.com/en-us/library/bb340482(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb340482%28v=vs.110%29.aspx)

            // Returns the first element of a sequence.

            // ---------------------------------

            // 2.3.

            // Enumerable.First<TSource>

            // (this IEnumerable<TSource> source, Func<TSource, Boolean> predicate)

            // OR

            // Enumerable.FirstOrDefault<TSource>

            // (this IEnumerable<TSource> source, Func<TSource, Boolean> predicate)

            // Reference:

            // [https://msdn.microsoft.com/en-us/library/bb535050(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb535050%28v=vs.110%29.aspx)

            // [https://msdn.microsoft.com/en-us/library/bb549039(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb549039%28v=vs.110%29.aspx)

            // Returns the first element in a sequence that satisfies a specified condition.

            int[] intArr = { 1, 2, 3, 4, 5, 6, 7, 8, 9 };

            int[] emptyIntArr = { };

            // 1.1. -----------------------------------

            Console.WriteLine("1.1. intArr.First() -------------- ");

            int intArrFirst = intArr.First();

            Console.WriteLine($"intArr.First()=={intArrFirst}");

            //intArr.First()==1

            // 1.2. -----------------------------------

            Console.WriteLine("1.2. emptyIntArr.First() -------------- ");

            try

            {

                int emptyIntArrFirst = emptyIntArr.First();

                //Throws InvalidOperationException.

                Console.WriteLine($"emptyIntArr.First()=={emptyIntArrFirst}");

            }

            catch (Exception e)

            {

                Console.WriteLine(e);

            }

            // System.InvalidOperationException: Sequence contains no elements

            // at System.Linq.Enumerable.First[TSource](IEnumerable`1 source)

            // at Sample.Program.FirstSample() in d:\0\_mydocument\documents\visual studio 2017\Projects\Sample\Sample\Program.cs:line 39

            // 1.3. -----------------------------------

            Console.WriteLine("1.3. intArr.First(Func<TSource, Boolean> predicate) -------------- ");

            int intArrFirstOddNumber = intArr.First(x => x % 2 != 0);

            Console.WriteLine($"intArr.First(x => x % 2 != 0)=={intArrFirstOddNumber}");

            // intArr.First(x => x % 2 != 0)==1

            // 1.4. -----------------------------------

            Console.WriteLine("1.4. intArr.First(Func<TSource, Boolean> predicate) -------------- ");

            try

            {

                int intArrFirstMod2Equal3 = intArr.First(x => x % 2 == 3);

                //Throws InvalidOperationException, as no element in the sequence satisfies

                Console.WriteLine($"intArr.First(x => x % 2 == 3)=={intArrFirstMod2Equal3}");

            }

            catch (Exception e)

            {

                Console.WriteLine(e);

            }

            // System.InvalidOperationException: Sequence contains no matching element

            // at System.Linq.Enumerable.First[TSource](IEnumerable`1 source, Func`2 predicate)

            // at Sample.Program.FirstSample() in d:\0\_mydocument\documents\visual studio 2017\Projects\Sample\Sample\Program.cs:line 58

        }

       // 2. ======================================

        //FirstOrDefaultSample()

        static void FirstOrDefaultSample()

        {

            // 2.1.2.

            // FirstOrDefault

            // returns default value of type

            // if the sequence does not contain any elements

            // or if no element in the sequence satisfies the condition

            // For the reference type, the default value is null.

            int[] intArr = { 1, 2, 3, 4, 5, 6, 7, 8, 9 };

            int[] emptyIntArr = { };

            // 2.1. -----------------------------------

            Console.WriteLine("2.1. intArr.First(Func<TSource, Boolean> predicate) -------------- ");

            int intArrFirstMod2Equal3 = intArr.FirstOrDefault(x => x % 2 == 3);

            Console.WriteLine($"intArr.FirstOrDefault(x => x % 2 == 3)=={intArrFirstMod2Equal3}");

            //intArr.FirstOrDefault(x => x % 2 == 3)==0

            // 2.2. -----------------------------------

            Console.WriteLine("2.2. emptyIntArr.FirstOrDefault() -------------- ");

            int emptyIntArrFirstOrDefault = emptyIntArr.FirstOrDefault();

            Console.WriteLine($"emptyIntArr.FirstOrDefault()=={emptyIntArrFirstOrDefault}");

            //emptyIntArr.FirstOrDefault()==0

        }

        // 3. ======================================

        //ElementAt\_ElementAtOrDefaultSample()

        static void ElementAt\_ElementAtOrDefaultSample()

        {

            // 4.

            // ElementAt V.S. ElementAtOrDefault

            // ---------------------------------

            // 4.1.

            // Enumerable.ElementAt<TSource>

            // (this IEnumerable<TSource> source, Int32 index)

            // Reference:

            // [https://msdn.microsoft.com/en-us/library/bb299233(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb299233%28v=vs.110%29.aspx)

            // Returns the element at a specified index in a sequence.

            // Throws ArgumentNullException if source is null.

            // Throws ArgumentOutOfRangeException

            // if index is less than 0 or greater than or equal to the number of elements in source.

            // ---------------------------------

            // 4.2.

            // Enumerable.ElementAtOrDefault<TSource>

            // (this IEnumerable<TSource> source, Int32 index)

            // Reference:

            // [https://msdn.microsoft.com/en-us/library/bb494386(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb494386%28v=vs.110%29.aspx)

            // Returns the element at a specified index in a sequence

            // or a default value if the index is out of range.

            // Throws ArgumentNullException if source is null.

            int[] intArr = { 1, 2, 3, 4, 5, 6, 7, 8, 9 };

            int[] emptyIntArr = { };

            // 3.1. -----------------------------------

            Console.WriteLine("3.1. Enumerable.ElementAt<TSource>(this IEnumerable<TSource> source, Int32 index) -------------- ");

            int intArrElementAt0 = intArr.ElementAt(0);

            Console.WriteLine($"intArr.ElementAt(0)=={intArrElementAt0}");

            //intArr.ElementAt(0)==1

            // 3.2. -----------------------------------

            Console.WriteLine("3.2. Enumerable.ElementAt<TSource>(this IEnumerable<TSource> source, Int32 index) -------------- ");

            try

            {

                int emptyIntArrElementAt0 = emptyIntArr.ElementAt(0);

                Console.WriteLine($"emptyIntArr.ElementAt(0)=={emptyIntArrElementAt0}");

                //Throws ArgumentOutOfRangeException

            }

            catch (Exception e)

            {

                Console.WriteLine(e);

            }

            // 3.3. -----------------------------------

            Console.WriteLine("3.3. Enumerable.ElementAtOrDefault<TSource>(this IEnumerable<TSource> source, Int32 index) -------------- ");

            int emptyIntArrElementAtOrDefault0 = emptyIntArr.ElementAtOrDefault(0);

            Console.WriteLine($"emptyIntArr.ElementAtOrDefault(0)=={emptyIntArrElementAtOrDefault0}");

            //emptyIntArr.ElementAtOrDefault(0)==0

        }

       // 4. ======================================

        //Single\_SingleOrDefaultSample()

        static void Single\_SingleOrDefaultSample()

        {

            // 5.

            // Single V.S. SingleOrDefault

            // ---------------------------------

            // 5.1.

            // Single V.S. SingleOrDefault

            // --------------

            // 5.1.1.

            // Single

            // throws ArgumentNullException if source is null.

            // Throws InvalidOperationException

            // if the input sequence contains more than one element.

            // or if The input sequence is empty.

            // --------------

            // 5.1.2.

            // SingleOrDefault

            // throws ArgumentNullException if source is null.

            // Throws InvalidOperationException

            // if the input sequence contains more than one element.

            // ---------------------------------

            // 5.2.

            // --------------

            // 5.2.1.

            // Enumerable.Single<TSource>

            // (this IEnumerable<TSource> source)

            // Reference:

            // [https://msdn.microsoft.com/en-us/library/bb155325(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb155325%28v=vs.110%29.aspx)

            // Returns the only element of a sequence,

            // and throws an exception

            // if there is not exactly one element in the sequence.

            // --------------

            // 5.2.2.

            // Enumerable.SingleOrDefault<TSource>

            // (this IEnumerable<TSource> source)

            // Reference:

            // [https://msdn.microsoft.com/en-us/library/bb342451(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb342451%28v=vs.110%29.aspx)

            // Returns the only element of a sequence,

            // or a default value if the sequence is empty;

            // this method throws an exception

            // if there is more than one element in the sequence.

            // ---------------------------------

            // 5.3.

            // --------------

            // 5.3.1.

            // Enumerable.Single<TSource>

            // (this IEnumerable<TSource> source, Func<TSource, Boolean> predicate)

            // Reference:

            // [https://msdn.microsoft.com/en-us/library/bb535118(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb535118%28v=vs.110%29.aspx)

            // Returns the only element of a sequence that satisfies a specified condition,

            // and throws an exception if more than one such element exists.

            // --------------

            // 5.3.2.

            // Enumerable.SingleOrDefault<TSource>

            // (this IEnumerable<TSource> source, Func<TSource, Boolean> predicate)

            // Reference:

            // [https://msdn.microsoft.com/en-us/library/bb549274(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb549274%28v=vs.110%29.aspx)

            // Returns the only element of a sequence that satisfies a specified condition

            // or a default value if no such element exists;

            // this method throws an exception

            // if more than one element satisfies the condition.

            int[] intArr = { 1, 2, 3, 4, 5, 6, 7, 8, 9 };

            int[] intSingleArr = { 2 };

            int[] emptyIntArr = { };

            // 4.1. -----------------------------------

            Console.WriteLine("4.1. Enumerable.Single<TSource>(this IEnumerable<TSource> source) -------------- ");

            int intSingleArrSingle = intSingleArr.Single();

            Console.WriteLine($"intSingleArr.Single()=={intSingleArrSingle}");

            //intSingleArr.Single()==2

            // 4.2. -----------------------------------

            Console.WriteLine("4.2. Enumerable.Single<TSource>(this IEnumerable<TSource> source) -------------- ");

            try

            {

                int intArrSingle = intArr.Single();

                //Throws InvalidOperationException

                Console.WriteLine($"intArr.Single()=={intArrSingle}");

            }

            catch (Exception e)

            {

                Console.WriteLine(e);

            }

            // 4.3. -----------------------------------

            Console.WriteLine("4.3. Enumerable.Single<TSource>(this IEnumerable<TSource> source, Func<TSource, Boolean> predicate) -------------- ");

            try

            {

                int intArrSingleMod2 = intArr.Single(x => x % 2 == 0);

                //Throws InvalidOperationException because mroe than one results.

                Console.WriteLine($"intArr.Single(x => x % 2 == 0)=={intArrSingleMod2}");

            }

            catch (Exception e)

            {

                Console.WriteLine(e);

            }

            // 4.4. -----------------------------------

            Console.WriteLine("4.4. Enumerable.SingleOrDefault<TSource>(this IEnumerable<TSource> source) -------------- ");

            int emptyIntArrSingleOrDefault = emptyIntArr.SingleOrDefault();

            Console.WriteLine($"emptyIntArr.SingleOrDefault()=={emptyIntArrSingleOrDefault}");

            //emptyIntArr.SingleOrDefault()==0

            // 4.5. -----------------------------------

            Console.WriteLine("4.5. Enumerable.SingleOrDefault<TSource>(this IEnumerable<TSource> source) -------------- ");

            try

            {

                int intArrSingle = intArr.SingleOrDefault();

                //Throws InvalidOperationException

                Console.WriteLine($"intArr.SingleOrDefault()=={intArrSingle}");

            }

            catch (Exception e)

            {

                Console.WriteLine(e);

            }

            // 4.6. -----------------------------------

            Console.WriteLine("4.6. Enumerable.SingleOrDefault<TSource>(this IEnumerable<TSource> source, Func<TSource, Boolean> predicate) -------------- ");

            try

            {

                int intArrSingleMod2 = intArr.SingleOrDefault(x => x % 2 == 0);

                //Throws InvalidOperationException because mroe than one results.

                Console.WriteLine($"intArr.SingleOrDefault(x => x % 2 == 0)=={intArrSingleMod2}");

            }

            catch (Exception e)

            {

                Console.WriteLine(e);

            }

        }

       // 5. ======================================

        //DefaultIfEmptySample()

        static void DefaultIfEmptySample()

        {

            // 6.

            // DefaultIfEmpty

            // ---------------------------------

            // 6.1.

            // Enumerable.DefaultIfEmpty<TSource>

            // (this IEnumerable<TSource> source)

            // Reference:

            // [https://msdn.microsoft.com/en-us/library/bb360179(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb360179%28v=vs.110%29.aspx)

            // Returns the elements of the specified sequence

            // or the type parameter's default value in a singleton collection

            // if the sequence is empty.

            // ---------------------------------

            // 6.2.

            // Enumerable.DefaultIfEmpty<TSource>

            // (this IEnumerable<TSource> source, TSource defaultValue)

            // Reference:

            // [https://msdn.microsoft.com/en-us/library/bb355419(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb355419%28v=vs.110%29.aspx)

            // Returns the elements of the specified sequence

            // or the specified value in a singleton collection

            // if the sequence is empty.

            int[] intArr = { 1, 2, 3, 4, 5, 6, 7, 8, 9 };

            int[] emptyIntArr = { };

            // 5.1. -----------------------------------

            Console.WriteLine("5.1. Enumerable.DefaultIfEmpty<TSource>(this IEnumerable<TSource> source) -------------- ");

            IEnumerable<int> intArrDefaultIfEmpty = intArr.DefaultIfEmpty();

            foreach (int i in intArrDefaultIfEmpty)

            {

                Console.Write($" [ {i} ] ");

            }

            Console.WriteLine();

            // [ 1 ]  [ 2 ]  [ 3 ]  [ 4 ]  [ 5 ]  [ 6 ]  [ 7 ]  [ 8 ]  [ 9 ]

            // 5.2. -----------------------------------

            Console.WriteLine("5.2. Enumerable.DefaultIfEmpty<TSource>(this IEnumerable<TSource> source) -------------- ");

            IEnumerable<int> emptyIntArrDefaultIfEmpty = emptyIntArr.DefaultIfEmpty();

            foreach (int i in emptyIntArrDefaultIfEmpty)

            {

                Console.Write($" [ {i} ] ");

            }

            Console.WriteLine();

            // [ 0 ]

            // 5.3. -----------------------------------

            Console.WriteLine("5.3. Enumerable.DefaultIfEmpty<TSource>(this IEnumerable<TSource> source, TSource defaultValue) -------------- ");

            IEnumerable<int> emptyIntArrDefaultIfEmpty100 = emptyIntArr.DefaultIfEmpty(100);

            foreach (int i in emptyIntArrDefaultIfEmpty100)

            {

                Console.Write($" [ {i} ] ");

            }

            Console.WriteLine();

            // [ 100 ]

        }

    }

}

/\*

1.

First, FirstOrDefault, Last, LastOrDefault, ElementAt, ElementAtOrDefault, Single, SingleOrDefault, and DefaultIfEmpty are Element Operators which retrieve a single element from a sequence.  All of these have their own overloading methods.

-----------------------------------------------

2.

First V.S. FirstOrDefault

---------------------------------

2.1.

First V.S. FirstOrDefault

--------------

2.1.1.

First

throws InvalidOperationException

if the sequence does not contain any elements

or if no element in the sequence satisfies the condition

--------------

2.1.2.

FirstOrDefault

returns default value of type

if the sequence does not contain any elements

or if no element in the sequence satisfies the condition

For the reference type, the default value is null.

---------------------------------

2.2.

Enumerable.First<TSource>

(this IEnumerable<TSource> source)

OR

Enumerable.FirstOrDefault<TSource>

(this IEnumerable<TSource> source)

Reference:

[https://msdn.microsoft.com/en-us/library/bb291976(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb291976%28v=vs.110%29.aspx)

[https://msdn.microsoft.com/en-us/library/bb340482(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb340482%28v=vs.110%29.aspx)

Returns the first element of a sequence.

---------------------------------

2.3.

Enumerable.First<TSource>

(this IEnumerable<TSource> source, Func<TSource, Boolean> predicate)

OR

Enumerable.FirstOrDefault<TSource>

(this IEnumerable<TSource> source, Func<TSource, Boolean> predicate)

Reference:

[https://msdn.microsoft.com/en-us/library/bb535050(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb535050%28v=vs.110%29.aspx)

[https://msdn.microsoft.com/en-us/library/bb549039(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb549039%28v=vs.110%29.aspx)

Returns the first element in a sequence that satisfies a specified condition.

-----------------------------------------------

3.

Last V.S. LastOrDefault

---------------------------------

3.1.

Last V.S. LastOrDefault

--------------

3.1.1.

Last

throws InvalidOperationException

if the sequence does not contain any elements

or if no element in the sequence satisfies the condition

--------------

3.1.2.

LastOrDefault

returns default value of type

if the sequence does not contain any elements

or if no element in the sequence satisfies the condition

For the reference type, the default value is null.

---------------------------------

3.2.

Enumerable.Last<TSource>

(this IEnumerable<TSource> source)

OR

Enumerable.LastOrDefault<TSource>

(this IEnumerable<TSource> source)

Reference:

[https://msdn.microsoft.com/en-us/library/bb358775(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb358775%28v=vs.110%29.aspx)

[https://msdn.microsoft.com/en-us/library/bb301849(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb301849%28v=vs.110%29.aspx)

Returns the last element of a sequence.

---------------------------------

3.3.

Enumerable.Last<TSource>

(this IEnumerable<TSource> source, Func<TSource, Boolean> predicate)

OR

Enumerable.LastOrDefault<TSource>

(this IEnumerable<TSource> source, Func<TSource, Boolean> predicate)

Reference:

[https://msdn.microsoft.com/en-us/library/bb549138(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb549138%28v=vs.110%29.aspx)

[https://msdn.microsoft.com/en-us/library/bb548915(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb548915%28v=vs.110%29.aspx)

Returns the last element of a sequence that satisfies a specified condition.

-----------------------------------------------

4.

ElementAt V.S. ElementAtOrDefault

---------------------------------

4.1.

Enumerable.ElementAt<TSource>

(this IEnumerable<TSource> source, Int32 index)

Reference:

[https://msdn.microsoft.com/en-us/library/bb299233(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb299233%28v=vs.110%29.aspx)

Returns the element at a specified index in a sequence.

Throws ArgumentNullException if source is null.

Throws ArgumentOutOfRangeException

if index is less than 0 or greater than or equal to the number of elements in source.

---------------------------------

4.2.

Enumerable.ElementAtOrDefault<TSource>

(this IEnumerable<TSource> source, Int32 index)

Reference:

[https://msdn.microsoft.com/en-us/library/bb494386(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb494386%28v=vs.110%29.aspx)

Returns the element at a specified index in a sequence

or a default value if the index is out of range.

Throws ArgumentNullException if source is null.

-----------------------------------------------

5.

Single V.S. SingleOrDefault

---------------------------------

5.1.

Single V.S. SingleOrDefault

--------------

5.1.1.

Single

throws ArgumentNullException if source is null.

Throws InvalidOperationException

if the input sequence contains more than one element.

or if The input sequence is empty.

--------------

5.1.2.

SingleOrDefault

throws ArgumentNullException if source is null.

Throws InvalidOperationException

if the input sequence contains more than one element.

---------------------------------

5.2.

--------------

5.2.1.

Enumerable.Single<TSource>

(this IEnumerable<TSource> source)

Reference:

[https://msdn.microsoft.com/en-us/library/bb155325(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb155325%28v=vs.110%29.aspx)

Returns the only element of a sequence,

and throws an exception

if there is not exactly one element in the sequence.

--------------

5.2.2.

Enumerable.SingleOrDefault<TSource>

(this IEnumerable<TSource> source)

Reference:

[https://msdn.microsoft.com/en-us/library/bb342451(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb342451%28v=vs.110%29.aspx)

Returns the only element of a sequence,

or a default value if the sequence is empty;

this method throws an exception

if there is more than one element in the sequence.

---------------------------------

5.3.

--------------

5.3.1.

Enumerable.Single<TSource>

(this IEnumerable<TSource> source, Func<TSource, Boolean> predicate)

Reference:

[https://msdn.microsoft.com/en-us/library/bb535118(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb535118%28v=vs.110%29.aspx)

Returns the only element of a sequence that satisfies a specified condition,

and throws an exception if more than one such element exists.

--------------

5.3.2.

Enumerable.SingleOrDefault<TSource>

(this IEnumerable<TSource> source, Func<TSource, Boolean> predicate)

Reference:

[https://msdn.microsoft.com/en-us/library/bb549274(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb549274%28v=vs.110%29.aspx)

Returns the only element of a sequence that satisfies a specified condition

or a default value if no such element exists;

this method throws an exception

if more than one element satisfies the condition.

-----------------------------------------------

6.

DefaultIfEmpty

---------------------------------

6.1.

Enumerable.DefaultIfEmpty<TSource>

(this IEnumerable<TSource> source)

Reference:

[https://msdn.microsoft.com/en-us/library/bb360179(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb360179%28v=vs.110%29.aspx)

Returns the elements of the specified sequence

or the type parameter's default value in a singleton collection

if the sequence is empty.

---------------------------------

6.2.

Enumerable.DefaultIfEmpty<TSource>

(this IEnumerable<TSource> source, TSource defaultValue)

Reference:

[https://msdn.microsoft.com/en-us/library/bb355419(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb355419%28v=vs.110%29.aspx)

Returns the elements of the specified sequence

or the specified value in a singleton collection

if the sequence is empty.

\*/

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated