(T3)討論LinqToObject的Where  
CourseGUID: 5ba9a6fe-7475-4b0c-8b99-bbcf7f5e2e1c  
=======================================================================  
(T3)討論LinqToObject的Where  
=======================================================================  
0. Summary

-----------

1. New Project

1.1. Create New Project : Sample

-----------

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

0. Summary

1.

Where

1.1.

Where is a Linq query operator which

contains a predicate condition to filter the data,

just like the WHERE keyword in TSQL.

A predicate is a function to test each element for a condition.

The where query operator is optional.

1.2.

Enumerable.Where<TSource>(this IEnumerable<TSource> source, Func<TSource, Boolean> filter)

Reference:

[https://msdn.microsoft.com/en-us/library/bb534803(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb534803%28v=vs.110%29.aspx)

Filters a sequence of values based on a predicate.

1.3.

Enumerable.Where<TSource>(this IEnumerable<TSource> source, Func<TSource, Int32, Boolean> filter)

Reference:

[https://msdn.microsoft.com/en-us/library/bb534647(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb534647%28v=vs.110%29.aspx)

Filters a sequence of values based on a predicate

which has its source and source index as input.

predicate here is a function to test each source element for a condition.

The second parameter of the function represents the index of the source element.

Each element's index is used in the logic of the predicate function.

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

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





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

2. Sample : Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

namespace Sample

{

    class Program

    {

        static void Main(string[] args)

        {

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

            //Enumerable.Where<TSource>(this

 IEnumerable<TSource> source, Func<TSource, Boolean> filter)

            Console.WriteLine("1. WhereSample1 ================== ");

            WhereSample1();

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

            //Enumerable.Where<TSource>(this

 IEnumerable<TSource> source, Func<TSource, Int32, Boolean> filter)

            Console.WriteLine("2. WhereSample2 ================== ");

            WhereSample2();

            Console.ReadLine();

        }

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

        //Enumerable.Where<TSource>(this

IEnumerable<TSource> source, Func<TSource, Boolean> filter)

        //Reference:

        //[https://msdn.microsoft.com/en-us/library/bb534803(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb534803%28v=vs.110%29.aspx)

        //Filters a sequence of values based on a predicate.

        static void WhereSample1()

        {

            List<int> intList =

                new List<int> { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

            //1.1. intList.Where(num => IsOdd(num)); ---------

            Console.WriteLine("1.1. intList.Where(num => IsOdd(num)); --------- ");

            IEnumerable<int> intOddV1 =

                intList.Where(num => IsOdd(num));

            foreach (int intOddV1Item in intOddV1)

            {

                Console.WriteLine(intOddV1Item);

            }

            //1.2. intList.Where(num => IsOdd(num)); ---------

            Console.WriteLine("1.2. intList.Where(IsOdd); --------- ");

            IEnumerable<int> intOddV2 = intList.Where(IsOdd);

            foreach (int intOddV2Item in intOddV2)

            {

                Console.WriteLine(intOddV2Item);

            }

            //1.3. intList.Where(num => IsOdd(num)); ---------

            Console.WriteLine("1.3. intList.Where(i => i % 2 != 0) --------- ");

            IEnumerable<int> intOddV3 = intList.Where(i => i % 2 != 0);

            foreach (int intOddV3Item in intOddV3)

            {

                Console.WriteLine(intOddV3Item);

            }

        }

        static bool IsOdd(int i)

        {

            return i % 2 != 0;

        }

        //1. WhereSample1 ==================

        //1

        //3

        //5

        //7

        //9

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

        //Enumerable.Where<TSource>(this

 IEnumerable<TSource> source, Func<TSource, Int32, Boolean> filter)

        //Reference:

        //[https://msdn.microsoft.com/en-us/library/bb534647(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb534647%28v=vs.110%29.aspx)

        //Filters a sequence of values based on a predicate

        //which has its source and source index as input.

        //predicate here is a function to test each source element for a condition.

        //The second parameter of the function represents the index of the source element.

        //Each element's index is used in the logic of the predicate function.

        static void WhereSample2()

        {

            List<int> intList = new List<int> { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

            //2.1. Get Odd Number and Index ---------

            Console.WriteLine("2.1. Get Odd Number and Index --------- ");

            IEnumerable<string> oddIntAndIndexStrs = intList.Select((intNumber, index) => $"intNumber:{intNumber},index:{index}" );

            foreach (string oddIntAndIndexStrsItem in oddIntAndIndexStrs)

            {

                Console.WriteLine(oddIntAndIndexStrsItem);

            }

            //intNumber: 1,index: 0

            //intNumber: 2,index: 1

            //intNumber: 3,index: 2

            //intNumber: 4,index: 3

            //intNumber: 5,index: 4

            //intNumber: 6,index: 5

            //intNumber: 7,index: 6

            //intNumber: 8,index: 7

            //intNumber: 9,index: 8

            //intNumber: 10,index: 9

            //2.2. Get Odd Index ---------

            Console.WriteLine("2.2. Get Odd Index --------- ");

            IEnumerable<int> oddIndexes = intList

                .Select((num, index) => new { Number = num, Index = index })

                .Where(anonymousObject => anonymousObject.Number % 2 != 0)

                .Select(anonymousObject => anonymousObject.Index);

            foreach (int oddIndexesItem in oddIndexes)

            {

                Console.WriteLine($"oddIndexesItem : {oddIndexesItem}");

            }

            //oddIndexesItem : 0

            //oddIndexesItem : 2

            //oddIndexesItem : 4

            //oddIndexesItem : 6

            //oddIndexesItem : 8

        }

    }

}

Text

Description automatically generated

