(T10)比較SystemString、SystemTextStringBuilder。Ram的Stack、Heap  
CourseGUID: 29f1196a-1950-41a4-b9c1-dd13a9e92d92  
=======================================================================  
(T10)比較SystemString、SystemTextStringBuilder。Ram的Stack、Heap  
=======================================================================  
0. Summary

-----------

1. New Project

1.1. Create New Project

-----------

2. Program.cs  
=======================================================================

0. Summary

1.

System.String V.S. System.Text.StringBuilder

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

1.1.

System.Text.StringBuilder object is mutable

and good for performance,

but System.String is immutable.

Therefore, heavy string manipulation should

use System.Text.StringBuilder.

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

1.2.

System.String

//System.String

//string str1 = "IT";

//str1 += "Handy";

//str1 += "Guy";

//str1 += " Tutorial";

//str1 += " is";

//str1 += " awesome.";

//Console.WriteLine(str1);

1.2.1.

RAM contains Stack and Heap area.

Stack is for storing object reference variable.

Heap is for storing the object.

\* Firstly, str1 pointed "IT"

\* Secondly, str1 pointed "ITHandy",

and "IT" became an orphaned object in heap until it is garbage collected.

\* Thirdly, str1 pointed "ITHandyGuy",

and "ITHandy" became an orphaned object in heap until it is garbage collected.

\* Do so until str1 finally is pointing "ITHandyGuy Tutorial is Awesome.",

and rest of string objects became orphaned objects in heap until they are garbage collected.

    Stack    |   Heap

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

                |   "IT"

                |   "ITHandy"

                |   "ITHandyGuy"

                |   "ITHandyGuy Tutorial"

                |   "ITHandyGuy Tutorial is"

 str1   -------> "ITHandyGuy Tutorial is Awesome."

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

1.3.

System.Text.StringBuilder

//StringBuilder strBuilder =

//    new StringBuilder("IT");

//strBuilder.Append("Handy");

//strBuilder.Append("Guy");

//strBuilder.Append(" Tutorial");

//strBuilder.Append(" is");

//strBuilder.Append(" awesome.");

//Console.WriteLine(strBuilder.ToString());

1.3.1.

No matter how many times the string is manipulated,

StringBuilder always points to the same object instance.

    Stack         |   Heap

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

                     |

 StringBuilder ---> "ITHandyGuy Tutorial is Awesome."

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

1. New Project

1.1. Create New Project

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

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

Name: **Sample**







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

2. Program.cs

using System;

using System.Text;

namespace Sample

{

    class Program

    {

        static void Main(string[] args)

        {

            // 1. System.string ------------------------

            string str1 = "IT";

            str1 += "Handy";

            str1 += "Guy";

            str1 += " Tutorial";

            str1 += " is";

            str1 += " awesome.";

            Console.WriteLine(str1);

            // 2. System.Text.string ------------------------

            StringBuilder strBuilder =

                new StringBuilder("IT");

            strBuilder.Append("Handy");

            strBuilder.Append("Guy");

            strBuilder.Append(" Tutorial");

            strBuilder.Append(" is");

            strBuilder.Append(" awesome.");

            Console.WriteLine(strBuilder.ToString());

            // 3. System.string ------------------------

            string strInt = string.Empty;

            for (int i = 0; i < 100; i++)

            {

                strInt += i + " ";

            }

            Console.WriteLine(strInt);

            // The System.string is manipulated 100 times,

            // thus, it will create 99 orphaned objects

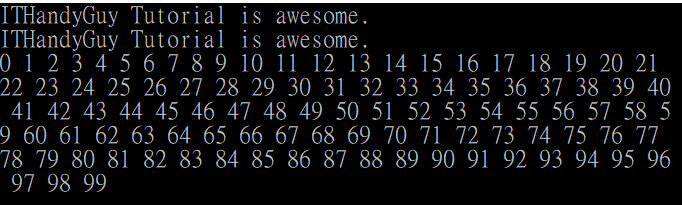
            // until they are garbage collected.

            Console.ReadLine();

        }

    }

}



/\*

1.

System.String V.S. System.Text.StringBuilder

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

1.1.

System.Text.StringBuilder object is mutable

and good for performance,

but System.String is immutable.

Therefore, heavy string manipulation should

use System.Text.StringBuilder.

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

1.2.

System.String

//System.String

//string str1 = "IT";

//str1 += "Handy";

//str1 += "Guy";

//str1 += " Tutorial";

//str1 += " is";

//str1 += " awesome.";

//Console.WriteLine(str1);

1.2.1.

RAM contains Stack and Heap area.

Stack is for storing object reference variable.

Heap is for storing the object.

\* Firstly, str1 pointed "IT"

\* Secondly, str1 pointed "ITHandy",

and "IT" became an orphaned object in heap until it is garbage collected.

\* Thirdly, str1 pointed "ITHandyGuy",

and "ITHandy" became an orphaned object in heap until it is garbage collected.

\* Do so until str1 finally is pointing "ITHandyGuy Tutorial is Awesome.",

and rest of string objects became orphaned objects in heap until they are garbage collected.

    Stack    |   Heap

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

             |   "IT"

             |   "ITHandy"

             |   "ITHandyGuy"

             |   "ITHandyGuy Tutorial"

             |   "ITHandyGuy Tutorial is"

 str1   -------> "ITHandyGuy Tutorial is Awesome."

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

1.3.

System.Text.StringBuilder

//StringBuilder strBuilder =

//    new StringBuilder("IT");

//strBuilder.Append("Handy");

//strBuilder.Append("Guy");

//strBuilder.Append(" Tutorial");

//strBuilder.Append(" is");

//strBuilder.Append(" awesome.");

//Console.WriteLine(strBuilder.ToString());

1.3.1.

No matter how many times the string is manipulated,

StringBuilder always points to the same object instance.

    Stack       |   Heap

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

                |

 StringBuilder ---> "ITHandyGuy Tutorial is Awesome."

\*/