using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

namespace DotNet\_1

{

class Program

{

delegate List<int> objDelegate(List<int> abc);

static void Main(string[] args)

{

Console.WriteLine("------------------2D Array-------------------");

TwoDArray obj2DArray = new TwoDArray();

obj2DArray.Print2DArray();

Console.WriteLine("----------------------------------------------");

Console.WriteLine("------------------Multiple Inheritance-------------------");

MultipleInheritance M = new MultipleInheritance();

M.GetMessage();

Console.WriteLine("----------------------------------------------");

Console.WriteLine("------------------Virtual class-------------------");

BaseClass objBaseClass = new BaseClass();

objBaseClass.PrintMessage();

ChildClass objChildClass = new ChildClass();

objChildClass.PrintMessage();

Console.WriteLine("----------------------------------------------");

Console.WriteLine("------------------Delegate and Lambda Expression-------------------");

DelegateExample objDelegateExample = new DelegateExample();

objDelegate \_delegateobj = objDelegateExample.GetDivisibleBy3List;

List<int> numberlist = new List<int> {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30 };

Console.Write("Number divisible by 3 from List : ");

var result =\_delegateobj(numberlist);

foreach (var item in result)

{

Console.Write(item + ",");

}

Console.WriteLine();

Console.WriteLine("----------------------------------------------");

Console.WriteLine("------------------Extend the functionality of the System.String-------------------");

string validEmail = "user@abc.com";

string invalidEmail = "user@abc";

Console.WriteLine(validEmail + " - IsValid Email? {0}", validEmail.IsValidEmail());

Console.WriteLine(invalidEmail + " - IsValid Email? {0}", invalidEmail.IsValidEmail());

Console.ReadLine();

}

}

#region 2DArray

public class TwoDArray

{

public void Print2DArray()

{

string[,] Matrix = new string[,]

{

{"a","b","c" },

{"d","e","f" },

{"g","h","i" }

};

for (int i = 0; i <= Matrix.GetUpperBound(0); i++)

{

Console.WriteLine("{0} {1} {2}", Matrix[i, 0], Matrix[i, 1], Matrix[i, 2]);

}

}

}

#endregion

#region Multiple Inheritance and Virtual Method

public class MultipleInheritance : IBase1, IBase2

{

public void GetMessage()

{

Console.WriteLine("Multiple Inheritance!");

}

}

public interface IBase1

{

void GetMessage();

}

public interface IBase2

{

void GetMessage();

}

public class ChildClass : BaseClass

{

public override void PrintMessage()

{

Console.WriteLine("Message From Child Class");

}

}

public class BaseClass

{

public virtual void PrintMessage()

{

Console.WriteLine("Message From Base Class");

}

}

#endregion

#region Delegate

public class DelegateExample

{

public List<int> GetDivisibleBy3List(List<int> numlist)

{

return numlist.FindAll(i => i % 3 == 0);

}

}

#endregion Delegate

#region Regular Expression

public static class StringExtensions

{

public static bool IsValidEmail(this string email)

{

return Regex.Match(email, @"\w+([-+.']\w+)\*@\w+([-.]\w+)\*\.\w+([-.]\w+)\*", RegexOptions.IgnoreCase).Success;

}

}

#endregion

}

