12\_CSHARP\_Basics\_Assignment-1\_613786\_PravinGupta

1. Write a program in C# Sharp for a 2D array of size 3x3 and print the matrix

Solution:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

static void Main(string[] args)

{

int i, j;

int[,] arr1 = new int[3, 3];

Console.Write("\n\nRead a 2D array of size 3x3 and print the matrix:\n");

Console.Write("------------------------------------------------------\n");

Console.Write("Input elements in the matrix:\n");

for (i = 0; i < 3; i++)

{

for (j = 0; j < 3; j++)

{

Console.Write("element - [{0},{1}] : ", i, j);

arr1[i, j] = Convert.ToInt32(Console.ReadLine());

}

}

Console.Write("\nThe matrix is: \n");

for (i = 0; i < 3; i++)

{

Console.Write("\n");

for (j = 0; j < 3; j++)

Console.Write("{0}\t", arr1[i, j]);

}

Console.Write("\n\n");

}

}

}

**Output**:

Read a 2D array of size 3x3 and print the matrix:

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

Input elements in the matrix:

element - [0,0] : 1

element - [0,1] : 2

element - [0,2] : 3

element - [1,0] : 4

element - [1,1] : 5

element - [1,2] : 6

element - [2,0] : 7

element - [2,1] : 8

element - [2,2] : 9

The matrix is:

1 2 3

4 5 6

7 8 9

1. Write an application to implement multiple inheritance and overriding using virtual method.

Solution:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Inheritance

{

interface calc1

{

int add(int a, int b);

}

interface calc2

{

int sub(int x, int y);

}

interface calc3

{

int mul(int r, int s);

}

interface calc4

{

int div(int c, int d);

}

class Ab

{

public virtual void display()

{

Console.WriteLine("Ab Class");

}

}

class Calculation : Ab,calc1, calc2, calc3, calc4

{

public int result1;

public int add(int a, int b)

{

return result1 = a + b;

}

public int result2;

public int sub(int x, int y)

{

return result2 = x - y;

}

public int result3;

public int mul(int r, int s)

{

return result3 = r \* s;

}

public int result4;

public int div(int c, int d)

{

return result4 = c / d;

}

public override void display()

{

Console.WriteLine("Calculation class");

}

}

class Program

{

static void Main(string[] args)

{

Calculation c = new Calculation();

c.add(8, 2);

c.sub(20, 10);

c.mul(5, 2);

c.div(20, 10);

Console.WriteLine("Multiple Inheritance concept Using Interfaces :\n ");

Console.WriteLine("Addition: " + c.result1);

Console.WriteLine("Substraction: " + c.result2);

Console.WriteLine("Multiplication :" + c.result3);

Console.WriteLine("Division: " + c.result4);

c.display();

Console.ReadKey();

}

}

}

1. Application is maintaining a collection (List) of Integers as NumList. We need to print the numbers from NumList those are divisible by 3. Implement the functionality using:

A. Delegates

B. Lambda Expression.

Solution:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Delegates

{

class Program

{

public delegate void MyDelegate(List<Int32> x);

static void Main(string[] args)

{

List<Int32> a=new List<int>();

a.Add(3);

a.Add(1);

a.Add(9);

a.Add(78);

a.Add(101);

MyDelegate myDelegate = new MyDelegate((List<Int32> y) =>

{

foreach (var item in y)

{

if (item % 3 == 0)

{

Console.WriteLine(item.ToString());

}

}

});

myDelegate(a);

Console.ReadKey();

}

}

}

**Output**:

3

9

78

1. Extend the functionality of the System.String class with function IsEmail() that will check the user input.

Solution:

**Extension Method**:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

namespace Extensions

{

public static class Extension

{

public static bool IsEmail(this string s)

{

bool isEmail = Regex.IsMatch(s, @"\A(?:[a-z0-9!#$%&'\*+/=?^\_`{|}~-]+(?:\.[a-z0-9!#$%&'\*+/=?^\_`{|}~-]+)\*@(?:[a-z0-9](?:[a-z0-9-]\*[a-z0-9])?\.)+[a-z0-9](?:[a-z0-9-]\*[a-z0-9])?)\Z", RegexOptions.IgnoreCase);

return isEmail;

}

}

}

**Main**:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Extensions;

namespace ExtendingStrings

{

class Program

{

static void Main(string[] args)

{

string inputStr = Console.ReadLine();

if (inputStr.IsEmail())

{

Console.WriteLine(true);

}

else

{

Console.WriteLine(false);

}

Console.ReadKey();

}

}

}