Họ và tên: Đỗ Thành Đạt

Mã SV : 21it268

Lớp: 21SE1

Ex1

    public delegate int MyDelegate(int numOne, int numTwo);

    class Program

    {

        static int Add(int numOne, int numTwo) => numOne + numTwo;

        static int Subtract(int numOne, int numTwo) => numOne - numTwo;

        static void Main(string[] args)

        {

      int n1 = 30;

            int n2 = 25;

            int result;

            // #2. Set target method

            MyDelegate obj1 = new MyDelegate(Add);

            // #3. Invoke method

            result = obj1(n1, n2);

            Console.WriteLine($"{n1} + {n2} = {result}");

            // Set target method

            MyDelegate obj2 = Subtract;

            // Invoke method

            result = obj2.Invoke(n1, n2);

            Console.WriteLine($"{n1} - {n2} = {result}");

            Console.ReadLine();

        }

    }

A black background with numbers and equal lines

Description automatically generated with medium confidence

Ex2

namespace Ex2

{

// declaring a delegate

    public delegate void MyDelegate(string msg);

    class MyClass {

        public static void Print(string message) =>

            Console.WriteLine($"{message.ToUpper()}");

        public static void Show(string message) =>

            Console.WriteLine($"{message.ToLower()}");

    }

    class Program {

        // MyDelegate type parameter

        static void InvokeDelegate(MyDelegate dele, string msg) => dele(msg);

        static void Main(string[] args) {

            string msg = "Passing Delegate As A Parameter";

            InvokeDelegate(MyClass.Print, msg);

            InvokeDelegate(MyClass.Show, msg);

            Console.ReadLine();

        }

    }

}

A black background with white letters

Description automatically generated

Ex3

// declaring a delegate

public delegate void MyDelegate(string msg);

class MyClass

{

    public static void Print(string message) =>

        Console.WriteLine($"{message.ToUpper()}");

    public static void Show(string message) =>

        Console.WriteLine($"{message.ToLower()}");

    public static void Display(string message) =>

        Console.WriteLine($"{message}");

}

class Program

{

    static void Main(string[] args)

    {

        string msg = "Multicast Delegate";

        MyDelegate MyDele01 = MyClass.Print;

        MyDelegate MyDele02 = MyClass.Show;

        Console.WriteLine("\*\*\*Combines MyDele01 + MyDele02\*\*\*");

        MyDelegate MyDele = MyDele01 + MyDele02;

        MyDele(msg);

        Console.WriteLine("\*\*\*Combines MyDele01 + MyDele02 + MyDele03\*\*\*");

        MyDelegate MyDele03 = MyClass.Display;

        MyDele += MyDele03;

        MyDele(msg);

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

        Console.WriteLine("\*\*\*Remove MyDele02\*\*\*");

        MyDele -= MyDele02;

        MyDele("Hello World !");

        Console.ReadLine();

    }

}

A screenshot of a computer

Description automatically generated

Ex4

public delegate void MyDele(int value);

class Program

{

    static void Main(string[] args)

    {

        MyDele printValue = delegate (int val)

        {

            Console.WriteLine("Inside Anonymous method. Value: {0}", val);

        };

        printValue += delegate

        {

            Console.WriteLine("This is Anonymous Method.");

        };

        printValue(100);

    }

}

A screen shot of a computer

Description automatically generated

Ex5

public delegate void PrintDetails(string msg);

class Program

{

    // Declaring an event

    event PrintDetails Print;

    void Show(string msg) => Console.WriteLine(msg.ToUpper());

    static void Main(string[] args)

    {

        Program p = new Program();

        // Register with an event

        p.Print += new PrintDetails(p.Show);

        // Raise "Print" event

        p.Print("Hello World.");

        Console.ReadLine();

    }

}



Ex6

class Program

{

    static void Main(string[] args)

    {

        int n1 = 12;

        int n2 = 16;

        int result;

        // Using lambda expression to add two numbers

        Func<int, int, int> addNumber = (a, b) => a + b;

        result = addNumber(n1, n2);

        Console.WriteLine($"{n1} + {n2} = {result}");

        Console.ReadLine();

    }

}



Ex 7

    class Program

    {

        static void Main(string[] args)

        {

            // Declare and initialize an array of strings

            string[] names = { "David", "Jane", "Peter", "John", "Mark" };

            var items = from word in names

                        where word.Contains("a")

                        select word;

            foreach (string s in items)

            {

                Console.WriteLine(s);

            }

            Console.ReadLine();

        }

    }

A screenshot of a computer

Description automatically generated

Ex8

    class Program

    {

        static void Main(string[] args)

        {

            // Declare and initialize an array of strings

            string[] names = { "David", "Jane", "Peter", "John", "Mark" };

            foreach (string item in names.OrderBy(s => s))

            {

                Console.WriteLine(item);

            }

            Console.ReadLine();

        }

    }

A screenshot of a computer

Description automatically generated