Evaluación Teórica .NET

1. Del siguiente código que compila y se ejecuta identificar y corregir el error.

**Tiempo: 5min**

using System;

public class MathUtils

{

public static double Average(int a, int b)

{

return a + b / 2;

}

public static void Main(string[] args)

{

Console.WriteLine(Average(2, 1));

}

}

1. Del siguiente código, generar :
   * Una nueva interface **IAlertDAO**, el cual contenga los mismos métodos que la clase **AlertDAO** (AddAlert y GetAlert)
   * La clase **AlertDAO** implemente la interface **IAlertDAO**
   * La clase **AlertService** reciba en el constructor la interface **IAlertDAO**
   * Los métodos **RaiseAlert** y **GetAlertTime** use el objeto pasado en el constructor.

**Tiempo: 15min**

using System.Collections.Generic;

using System;

public class AlertService

{

private readonly AlertDAO storage = new AlertDAO();

public Guid RaiseAlert()

{

return this.storage.AddAlert(DateTime.Now);

}

public DateTime GetAlertTime(Guid id)

{

return this.storage.GetAlert(id);

}

}

public class AlertDAO

{

private readonly Dictionary<Guid, DateTime> alerts = new Dictionary<Guid, DateTime>();

public Guid AddAlert(DateTime time)

{

Guid id = Guid.NewGuid();

this.alerts.Add(id, time);

return id;

}

public DateTime GetAlert(Guid id)

{

return this.alerts[id];

}

}

1. Implementar el método **UniqueNames**, cuando se pase dos arreglos con nombres, el método deberá de regresar solo un arreglo que contenga los nombres de ambos arreglos pero sin repetirlos.

Ejemplo: al invocar MergeNames.UniqueNames(new string[]{"Ava", "Emma", "Olivia"}, new string[] {"Olivia", "Sophia", "Emma"} ) el método debe de regresar un resultado Ava, Emma,Olivia y Sophia.

**Tiempo: 15min**

using System;

public class MergeNames

{

public static string[] UniqueNames(string[] names1, string[] names2)

{

throw new NotImplementedException();

}

public static void Main(string[] args)

{

string[] names1 = new string[] {"Ava", "Emma", "Olivia"};

string[] names2 = new string[] {"Olivia", "Sophia", "Emma"};

Console.WriteLine(string.Join(", ", MergeNames.UniqueNames(names1, names2))); // should print Ava, Emma, Olivia, Sophia

}

}

1. Del siguiente código, generar dos controles de entrada: **TextInput** que acepte todos los caracteres y **NumericInput** que solo acepte dígitos.
   * Implementar la clase **TextInput** que contenga:
     + Método publico **void Add(char c)**.- agregar el carácter indicado al valor actual
     + Método publico **string GetValue()**.-regrese el valor actual
   * Implementar la clase **NumericInput** donde:
     + Que herede de **TextInput**
     + Sobre escribir el método **Add** el cual ignore los valores que no son numéricos

Por ejemplo dentro del método MAIN, el resultado es:

1 0

**Tiempo: 15min**

public class TextInput { }

public class NumericInput { }

public class UserInput

{

public static void Main(string[] args)

{

//TextInput input = new NumericInput();

//input.Add('1');

//input.Add('a');

//input.Add('0');

//Console.WriteLine(input.GetValue());

}

}

**Tiempo máximo: 50 min.**