Segundo Parcial

Que realiza el conectionString

* Realiza la conexión a la base
* Realizar una consulta en la BD
* Crea un enlace con la base de datos
* Es un string que contiene la ruta y login a la base

Que garantiza una interface

* Que la clase que la implementa contiene todas los atributos que contiene
* Que la clase que la hereda contiene todos los métodos que contiene
* Que la clase que la implementa contiene todas las propiedades que contiene
* Que la clase que la implementa contiene todos los métodos que contiene

Cuando se termina un hilo.

* Cuando se ejecuta el método Abort.
* Cuando termina de ejecutar el método
* Cuando finaiiza el hilo padre
* Cuando ocurre un error

Que Clase explicita el resultado del Test

* TestMethod
* IsValid
* Assert
* Test

Como se debe implementar una interface en una clase hija de Persona

* public class Alumno : IAlumno, Persona
* public class Alumno <IAlumno: Persona>
* public class Alumno : Persona, IAlumno
* public class Alumno <Persona> , <IAlumno>

Con que método del SqlCommand ejecutamos una consulta Insert

* Read
* ExecuteScalar
* ExecuteNonQuery
* ExecuteReader

Como se Ejecutan las pruebas de un test unitario

* Desde el explorador de pruebas
* Desde la consola del proyecto
* Desde el método que quiere revisarse.
* Corren automáticamente cuando se debuguea el proyecto

Que puedo ingresar en el parámetro H de la siguiente clase: public class Auto<H> where H: Vehiculo , new()

* Una instancia de Vehículo
* Error en tiempo de diseño
* La clase Vehículo si tiene constructor por defecto
* Una instancia de un hijo de la Clase Vehículo que tengan constructor por defecto

Cual/es elemento/s pueden utilizar generics

* Interface
* Método
* Propiedad
* Clase

La serializacion Binaria

* Exporta todos los métodos y propiedades públicos
* Exporta todos los métodos y propiedades
* Exporta todas las propiedades publicas
* Exporta todas las propiedades

Marque la/s Interfaz/es válida/s

* public interface miInterface { string Nombre { get; set; } }
* public abstract interface miInterface { virtual string Nombre{ get; set; } }
* public interface miInterface { virtual string Nombre{ get; set; } }
* public abstract interface miInterface { string Nombre{ get; set; } }

Que realiza la clase BinaryFormatter

* Serializa objetos en formato binario.
* Serializa objetos en formato XML.
* Establece la estructura que tendrá el archivo XML.
* Serializa Clases en formato binario.

Cuantas Interfaces pueden implementarse en una Clase

* Una
* Las interfaces no se implementan.
* Todas las que hereden del mismo padre.
* Todas que se requiera.

Cual de los siguiente metodos acepta el constructor de ParameterizedThreadStart \*

* public string metodo4(object o)
* private string metodo3(object o)
* public void metodo1(object o)
* private void metodo2(object o, EventArgs e)

Que código es correcto para verificar la existencia de un arhivo.

* File.Exists(path);
* Path.Exists(path);
* File A = new File(); A.Exists(path);
* Path A = new Path(); A.Exists(path);

Como se crean los test unitarios

* Creando un proyecto de Test
* Creando un entorno de Test
* Creando una clase de Test
* Realizando la conexión sql a un servidor de Desarrollo

Como capturo la excepción

* Con un objeto Exception
* Con un método try catch finally
* Con un bloque try y catch
* Con un Bloque try. No es necesario el catch

Qué pasa si una excepción no es manejada

* Sale un mensaje de error y continua el flujo
* Se abre un nuevo hilo de trabajo
* Se detiene el flujo y el programa deja de funcionar
* El programa no compila

En que nameSpace se encuentra SqlConnection

* Microsoft.System.Data;
* System.Data;
* Sql.Data;
* System.Data.SqlClient;

Como Identifico la excepción que se lanzo.

* Por el código de error
* Por el catch
* Por el mensaje de error
* Por la clase de la excepción

Un método genérico es

* Un método parametrizado
* Un método no sobrescribible
* Un método de clase
* Un método creado con “extends”

Los delegados

* Son una lista de punteros a métodos
* Son métodos ejecutables
* Son Colecciones de métodos
* Son un puntero a función

La serializacion XML

* Exporta todos los métodos y propiedades públicos
* Exporta todas las propiedades
* Exporta todos los métodos y propiedades
* Exporta todas las propiedades publicas

¿Cuándo se lanza una excepción?

* Cuando hay un error en tiempo de diseño
* Cuando el programador lo desee
* Cuando hay un error de ejecución
* Cuando se utilizar Try catch

Que propiedad de SqlCommand tiene la consulta SQL

* CommandText;
* Parameters;
* El consutructor;
* CommandType;

Los eventos

* Son la forma que tienen las clases para comunicarse
* Son acciones del usuario
* Son Metodos que se ejecutan asincrónicamente
* Son punteros a funciones

Marque la/s opción/es correcta/s

* Todas las interfaces son públicas.
* Las interfaces pueden heredar de otra interfaz.
* Las interfaces contienen atributos, propiedades y métodos abstractos.
* Todos sus métodos deben ser virtuales.

Que es un hilo

* Una serie de métodos ordenados
* Un proceso o rutina
* Una secuencia de tareas encadenadas
* La ejecución de un método

Que retorna el siguiente método:

string buscar(string connStr)

{

SqlConnection conexion = new SqlConnection(connStr);

SqlCommand comando =new SqlCommand();

comando.CommandText="SELECT nombre FROM Personas";

try

{

conexion.Open();

SqlDataReader oDr= comando.ExecuteReader();

string aux;

while (oDr.Read())

{

aux = oDr["nombre"].ToString();

}

}

catch(SqlException e)

{

return "Error en la sentencia SQL"

}

return aux;

}

* El ultimo nombre de la tabla persona
* Error en tiempo de ejecucion
* Error en la sentencia SQL
* Los nombres de la tabla persona

Cual/es de las siguientes declaraciones de clases es correcta

* public class Alumno<T, J> where T : new() where J : new() {}
* public class Alumno<T> where T: new() {}
* public class Alumno<T,J> where T,J : new() {}
* public class Alumno<T> where T is new() {}

Que hace la palabra where en un método parametrizado

* Realiza un filtro sobre los datos
* Es el nombre del párametro
* Ubica la clase donde se generó el método
* Restringe el parámetro que puede utilizarse

Que es un método de extensión

* Un método no sobrescribible
* Un método de clase
* Un método creado con “extends”
* Un método parametrizado

Como se lanza un evento

* Con el método Invoke
* Llamando al metodo por defecto de la insatancia
* Con el método Event
* Con el +=

Que retorna el siguiente código cuando no se encuentra el archivo.

public void Leer(string path)

{

try

{

File.OpenRead(path);

}

catch (Exception e)

{

MessageBox.Show("Salida 1");

throw new FileNotFoundException(path, e);

}

catch (FileNotFoundException e)

{

MessageBox.show("Salida 2");

throw new FileNotFoundException(path, e);

}

catch (IOException e)

{

MessageBox.show("Salida 3");

throw new FileNotFoundException(path, e);

}

}

* Error en tiempo de Diseño
* “salida 3 “ y Excepcion
* “salida 2 “ y Excepcion
* “salida 1 “ y Excepcion

Para que se utiliza la etiquita Serializable

* Para establecer que una clase se pude serializar Binaria
* Para establecer que una propiedad privada se pude serializar Binaria
* Para establecer que una propiedad privada se pude serializar XML
* Para establecer que una clase privada se pude serializar XML