2doParcialProg - 17/11/2020

Tuesday, November 17, 2020
M.Luana Jaimez

Hilos:

a. No es posible lanzar un hilo parametrizado.
b. Un proceso está compuesto de ‘n’ hilos. Si alguno queda abierto, el proceso seguirá vivo.
c. Si se quiere detener un hilo se utiliza el método Stop.
d. No es posible acceder a un control (WindowsForm) desde un hilo secundario.
e. Todas las anteriores.
f. Ninguna de las anteriores.
h. a, b y c.
i. a y c.

El método ‘Fill’ del SqlDataAdapter ¿ejecuta algún comando sobre la base de datos?

a. Sí, todos.
b. Algunos, según el RowState.
c. Sí, solo los que estén configurados.
d. Ninguno, es un método que no recibe parámetros.
e. Sí, el comando de actualización (updateCommand).
f. Ninguna de las anteriores.

Para conectarse con una base de datos SQL Server:

a. El objeto SqlCommand se conectará a la base de datos y ejecutará distintas consultas.
b. Podremos utilizar el SqlCommand para ejecutar consultas en distintos servidores (SQL, ORACLE, etc.), sólo cambiando la conexión.
c. SqlConnection administra la conexión con cualquier servidor (SQL, ORACLE, etc.).
d. Se pueden ejecutar sentencias al menos de dos formas: ExecuteNonQuery y ExecuteReader.
e. Todas las anteriores.
f. a y d
g. b, c y d.
h. c y d.
i. Ninguna de las anteriores.

¿Cuáles de las siguientes afirmaciones son verdaderas al hablar de métodos de extensión?

a. Permiten adicionar métodos y propiedades a tipos existentes sin crear un nuevo tipo derivado, recompilar o modificar de otra manera el tipo original.
b. Se pueden definir dentro de cualquier clase del sistema y para cualquier tipo de dato.
c. Por definición, la clase y el método deberán ser estáticos.
d. Su utilización será mediante un casteo de la clase extendida.
e. Permiten derivar clases selladas.
f. Todas las anteriores.
g. Ninguna de las anteriores.
h. a, c, d y e.
i. a, d y e.
j. a y d.

Indique cuáles de las siguientes afirmaciones sobre serialización son correctas:

a. Sólo se pueden serializar atributos y propiedades públicas, en formato XML.

b. Los objetos que se deseen serializar en formato binario no necesitan ningún agregado o característica en particular.
c. Los objetos que se deseen serializar en formato XML deben tener un constructor público con parámetros.
d. No es posible serializar en formato binario una gerarquía de clases (clase padre, clase hija) en una colección.
e. No es posible serializar en formato XML una gerarquía de clases (clase padre, clase hija) en una colección.
f. Todas las anteriores
g. Ninguna de las anteriores
h. a, c y d.
i. c y d.

El método 'Update' del SqlDataAdapter ¿ejecuta algún comando sobre la base de datos?

a. Sí, todos.
b. Algunos, según el RowState del SqlCommand.
c. Sí, solo los que estén configurados.
d. Sí, el comando de selección (selectCommand).
e. Sí, el comando de actualización (updateCommand).
f. Ninguna de las anteriores.

Interfaces:

a. Se implementan mediante el siguiente código: ClaseA : where(IMiInterfaz)
b. Podremos definir métodos, propiedades y atributos, sin indicar los modificadores de visibilidad.
c. Los métodos se definirán con la palabra reservada abstract ya que no definen código.
d. Debemos indicar la visibilidad de cada miembro de la interface.
e. Si los miembros de una interface se implementan de forma implícita, no se debe colocar el modificador de visibilidad.
f. Si quiero invocar a un miembro implementado explícitamente, debo invocarlo a través del nombre del método.
g. Al realizar la herencia de una clase que implementa una interfaz, podré implementar los métodos directamente en la clase que la hereda, sin deber implementarlos en la clase base.
h. a, b y f.
i. a y e.
j. a, e y f.
k. a y f.
l. Ninguna de las anteriores.

La serialización XML sirve para:

a. Generar archivos sólo con la extensión .bin o .dat.
b. Guardar en formato de bytes distintos tipos de datos.
c. Serializar clases.
d. Serializar clases, objetos e interfaces.
e. Todas las anteriores.
f. Serializar atributos públicos y privados.
g. Serializar métodos de instacia y métodos de clase.
h. a, b, c.
i. b, c, f, g.
j. b, d, f, g.
k. b, d, f.
l. Ninguna de las anteriores.

Cuando se realiza un 'test unitario':

a. Se puede verificar cómo se comporta una porción del código interpretando valores de variables, excepciones lanzadas en métodos de instancia.

b. La clase de la prueba hereda de UnitTest1.
c. A través de métodos de instancia de la clase Assert se puede informar el resultado de cada prueba.
d. Cada método llevará la etiqueta [WebMethod].
e. Todas las anteriores.
f. a, b y d.
g. a, c y d.
h. a y d.
i. Ninguna de las anteriores.

Delegados:

a. La propiedad de sólo lectura Method, devuelve una lista con todos los métodos de la lista de invocación.
b. Para agregar métodos a la lista de invocación, se utiliza el método Add, de la clase Delegate.
c. El método GetInvocationList, retorna el último método de la lista de invocación.
d. El método Invoke, invoca la ejecución del primer método que contiene el delegado.
e. Todas las anteriores.
f. Ninguna de las anteriores.

Los eventos:

a. Son un modo que tienen los objetos de proporcionar excepciones cuando ocurre algo en particular.
b. Se pueden referenciar a varios métodos (con distinta firma) para que controlen a un evento en particular.
c. El evento al ser lanzado conoce que método lo va a tratar.
d. Se implementan mediante delegados definidos por el framework.
e. El método manejador del evento debe estar implementado en la misma clase donde se definió el evento.
f. Todas las anteriores.
g. a, b y d
h. a, c, d y e.
i. a y d.
j. Ninguna de las anteriores.

Generics:

a. Se pueden utilizar en clases, métodos, atributos y propiedades, pero no al mismo tiempo.
b. Sólo se puede tener un tipo genérico por clase.
c. Al generar la clase, se reemplaza por un comodín nombrado cómo T.
d. Los métodos pueden recibir tipos genéricos, pero no retornarlos.
e. Las interfaces no pueden implementarlos.
f. a y b.
g. a y c.
h. a, b y c.
f. Ninguna de las anteriores.

Los parámetros de un comando (SqlCommand) deben nombrarse con '@' y según:

a. Los campos del DataTable
b. Los campos de la base de Datos.
c. Depende del tipo de comando.
d. Ninguna de las anteriores.

¿Qué son los test unitarios?

a. Son pruebas basadas en la ejecución, revisión y retroalimentación de las funcionalidades previamente diseñadas para el software.
b. Son aquellos que prueban que todos los elementos unitarios funcionan juntos correctamente probándolos en

grupo.
c. Son pruebas diseñadas para cada aplicación o proyecto, de forma que cada prueba sea independiente del resto.
d. Son pruebas individuales que se aplican sobre la aplicación, para probar el correcto funcionamiento de la misma.
e. Todas las anteriores.
f. Ninguna de las anteriores.

Dentro de una interfaz pueden definirse:

a. Métodos de instancia y métodos estáticos.
b. Métodos de instancia, propiedades y constructores de instancia.
c. Propiedades de instancia, atributos de instancia y métodos de instancia.
d. Constructores de clase.
e. a y c.
f. c y d.
g. Ninguna de las anteriores.

Para ejecutar un comando de eliminación de datos debo acceder al método:

a. ExecuteReader()
b. ExecuteScalar()
c. ExecuteNonQuery()
d. Todas las anterioes.
e. a y b.
f. a y c.
g. b y c.
h. Ninguna de las anteriores.

¿Cómo configuro la propiedad CommandText (de un SqlCommand) para obtener los registros (y campos) de la tabla paises, cuyo campo ID esté entre 25 y 30?

a. SELECT ALL FROM paises WHERE id > 24 < 31
b. SELECT * paises WHERE id >= 25 AND id <= 30
c. SELECT * FROM paises WHERE id > 24 AND id <= 30
c. SELECT * FROM ID WHERE paises.id > 24 AND paises.id <= 30
e. SELECT paises WHERE id > 25
f. Ninguna de las anteriores.
g. a y b.
h. a y c.
i. c y e.

Hilos. Suponiendo que el siguiente código se encuentra dentro de un Console Application, teniendo todas las referencias agregadas, ¿qué hará el siguiente código según las opciones dadas?

```
static void Main()
{
   Prueba p = new Prueba("hola");
}

class Prueba
{
   public Prueba(string mensaje)
   {
      Thread hilo = new Thread(new ParameterizedThreadStart(Prueba.Imprimir));
      hilo.Start(mensaje);
   }

   private static void Imprimir(object o)
   {
      Console.WriteLine(o);
   }
}
```

a. Error en tiempo de diseño.
b. Error en tiempo de ejecución.
c. Muestra, en un hilo secundario, el texto 'hola' por consola.
d. Muestra, en el hilo principal, el texto 'hola' por consola.
e. No se muestra nada.
f. Ninguna de las anteriores es correcta.
g. a y e.
h. b y e.

¿Qué haría el siguiente código, asumiendo que es el único código que está en el método Main, y que el archivo "C:\test.txt" existe?
1/1

```
using (StreamWriter lector = new StreamWriter("C:\\test.txt"))
{
   string linea = "";
   while (linea != null)
   {
     linea = lector.ReadLine();
     Console.WriteLine(linea);
   }
}
```

a. Lee el contenido del archivo caracter por caracter y lo muestra por consola.
b. Lee el contenido del archivo por línea (es decir, hasta encontrar el caracter 'enter') y muestra esa línea por consola.
c. La sintaxis es incorrecta.
d. Lee sólo la primera línea del archivo y la muestra por consola.
e. Arroja una excepción que no es controlada.
f. Ninguna de las anteriores.

Al utilizar manejo de excepciones:

a. Sólo deben utilizarse si se manipulan archivos (de texto, binarios o XML).
b. Se pueden definir 'n' bloques finally.
c. Obligatoriamente debe existir al menos un bloque try – catch - finally.
d. En el bloque catch no puede generarse ninguna excepción.
e. Los bloques catch van de lo general a lo particular.
f. Si se lanza una excepción propia, se debe utilizar el new Exception().
g. Ninguna de las anteriores.