Programación II C # Clase 11 Maximiliano Neiner

Temas a Tratar Archivos de Texto Serialización Serialización XML Serialización Binaria

Temas a Tratar Archivos de Texto StreamWriter StreamReader Excepciones Serialización Serialización XML Serialización Binaria

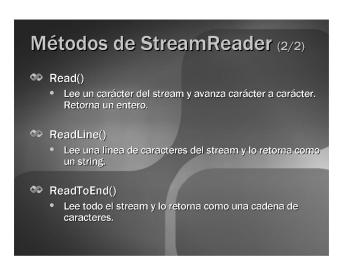
Archivos de Texto Control La clase StreamWriter escribe caracteres en archivos de texto. Control La clase StreamReader lee desde un archivo de texto. Control La clase StreamReader lee desde un archivo de texto. Control La clase StreamReader lee desde un archivo de texto. Control La clase StreamReader lee desde un archivo de texto.

Métodos de StreamWriter (1/2) StreamWriter (string path) Inicializa una nueva instancia de la clase StreamWriter, en un path específico. Si el archivo existe, se sobrescribirá, sino se creará. StreamWriter (string path, bool append) Ídem anterior, si append es true, se agregarán datos al archivo existente. Caso contrario, se sobrescribirá el archivo. StreamWriter (string path, bool append, Encoding e) Ídem anterior, dónde se le puede especificar el tipo de codificación que se utilizará al escribir en el archivo.

Métodos de StreamWriter (2/2) Write (string value) Escribe una cadena en el archivo sin provocar salto de línea. WriteLine(string value) Escribe una cadena en un archivo provocando salto de línea. Close() Cierra el objeto StreamWriter.

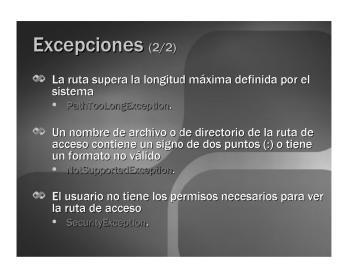
Temas a Tratar Archivos de Texto StreamWriter StreamReader Excepciones Serialización Serialización XML Serialización Binaria

Métodos de StreamReader (1/2) StreamReader (string path) Inicializa una nueva instancia de la clase StreamReader. El path especifica de donde se leerán los datos. StreamReader (string path, Encoding e) Ídem anterior, dónde se le especifica el tipo de codificación que se utilizará para leer el archivo. Close() Cierra el objeto StreamReader.





Excepciones (1/2) La ruta de acceso no es válida porque... Es una cadena de longitud cero, contiene sólo espacios en blanco, o contiene caracteres no válidos. (ArgumentException). La ruta de acceso es Null (ArgumentNullException). File señala a una ruta de acceso que no existe. FileNotFoundException DirectoryNotFoundException. El archivo está en uso por otro proceso o hay un error de E/S IOException.



Temas a Tratar

- Archivos de Texto
- Serialización
 - Introducción.
- Serialización XML
- Serialización Binaria

Serialización

- 🗪 ¿Qué es?
 - Es el proceso de convertir un objeto en memoria en una secuencia lineal de bytes.
- ¿Para qué sirve?
 - · Para pasarlo a otro proceso.
 - Para pasarlo a otra máquina.
 - Para grabarlo en disco.
 - Para grabarlo en una base de datos.

Los Formatters

- Controlan el formato de la serialización.
- Serialización a XML
 - Por defecto incluye sólo las propiedades y atributos públicos.
- Serialización Binaria
 - Por defecto incluyen todos los atributos y propiedades, ya sean públicas o privadas.
- 🗪 ¿Y después?
 - Se reconstruye el objeto mediante Deserialización proceso inverso en el mismo proceso o no, en la misma máquina o no.

Temas a Tratar

- Archivos de Texto
- Serialización
- Serialización XML
 - Introducción
 - Métodos de XmlSerializer
 - XmlTextWriter
 - XmlTextReader
- Serialización Binaria

Serialización XML (1/3)

- La serialización XML sólo serializa los atributos públicos y los valores de propiedad de un objeto en una secuencia XML.
- La serialización XML no convierte los métodos, indexadores, atributos privados ni propiedades de sólo lectura (salvo colecciones de sólo lectura).
- La clase central de la serialización XML es XmlSerializer y sus métodos más importantes son Serialize y Deserialize.

Serialización XML (2/3)

- La secuencia XML que genera XmlSerializer cumple con la recomendación 1.0 del W3C (www.w3.org) acerca del lenguaje de definición de esquemas XML (XSD).
- Además, los tipos de datos generados cumplen las especificaciones enumeradas en el documento titulado "XML Schema Part 2: Datatypes".
- Al crear una aplicación que utiliza la clase XmlSerializer, debe tener en cuenta los siguientes elementos y sus implicaciones:

Serialización XML (3/3)

- La clase XmlSerializer crea archivos C# (.cs) y los compila en archivos .dll en el directorio especificado por la variable de entorno TEMP; la serialización se produce con esos archivos DLL.
- Una clase debe tener un constructor por defecto para que XmlSerializer pueda serializarla.
- Sólo se pueden serializar los atributos y propiedades públicas.
- Los métodos no se pueden serializar.

Temas a Tratar

- Archivos de Texto
- ⇔ Serialización
- Serialización XML
 - Introducción
 - Métodos de XmlSerializer
 - XmlTextWriter
 - XmlTextReader
- Serialización Binaria

Métodos de XmlSerializer

- XmlSerializer (System.Type type)
 - Inicializa una nueva instancia de la clase XmlSerializer la cual puede serializar objetos del tipo especificado en el parámetro type.
- Serialize (System.IO.Stream stream, Object o)
 - Serializa el objeto especificado y escribe en un documento Xml usando el Stream especificado.
- Deserialize (System.IO.Stream stream)
 - Deserializa el documento Xml contenido por el Stream especificado.

Temas a Tratar

- Archivos de Texto
- Serialización
- Serialización XML
 - Introducción
 - Métodos de XmlSerializer
 - XmlTextWriter
 - XmlTextReader
- Serialización Binaria

La Clase XmlTextWriter

- Provee una manera de generar archivos con contenido de datos XML que cumple con la recomendación 1.0 del W3C (www.w3.org) acerca del lenguaje de definición de esquemas XML (XSD).
- Métodos:
 - XmlTextWriter (string filename, System.Text.Encoding encoding)
 - Crea una instancia de XmlTextWriter.
 - El filename indica en que archivo se escribirá.
 - Con encoding se indicará cual será la codificación.

Ejemplo - Serialización XML

Dato p = new Dato(); //Objeto a serializar. XmlTextWriter writer; //Objeto que escribirá en XML.

XmlSerializer ser; //Objeto que serializará.

writer = new XmlTextWriter(ArchivoXml, Codificación);

//Se indica ubicación del archivo XML y su codificación. ser = new XmlSerializer(typeof(Dato));

//Se indica el tipo de objeto ha serializar.

ser.Serialize(writer, p);

//Serializa el objeto p en el archivo contenido en writer.

writer.Close();

//Se cierra el objeto writer.

Temas a Tratar Archivos de Texto Serialización Serialización XML Introducción Métodos de XmlSerializer XmlTextWriter XmlTextReader Serialización Binaria



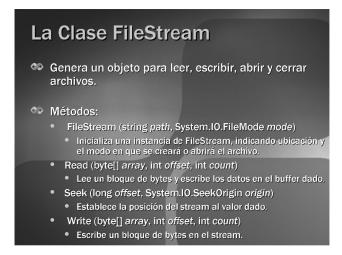




La Clase BinaryFormatter Serializa y Deserializa objetos en formato binario. Se encuentra en el espacio de nombres System.Runtime.Serialization.Formatters.Binary Puede serializar atributos públicos y privados. Una clase debe tener un constructor por defecto para que BinaryFormatter pueda serializarla. Los métodos más importantes de la clase BinaryFormatter son: Serialize Deserialize

Métodos de BinaryFormatter BinaryFormatter() Inicializa una nueva instancia de la clase BinaryFormatter. Serialize(System.IO.FileStream seralizationStream, Object graph) Serializa el objeto especificado y escribe en un archivo binario usando el serializationStream especificado. Deserialize(System.IO.FileStream serializationStream) Deserializa el archivo binario contenido por el serializationStream especificado.





Ejemplo - Serialización Binaria Dato p = new Dato(); //Objeto a serializar. FileStream fs; //Objeto que escribirá en binario. BinaryFormatter ser; //Objeto que serializará. fs = new FileStream(ArchivoBinario, FileMode.Create); //Se indica ubicación del archivo binario y el modo. ser = new BinaryFormatter(); //Se crea el objeto serializador. ser.Serialize(fs, p); //Serializa el objeto p en el archivo contenido en fs. fs.Close(); //Se cierra el objeto fs.



