**StrStreamReader**

Implementa un oggetto TextReader che legge i caratteri da un flusso di byte in una particolare codifica.

**Spazio dei nomi: System.IO:** using System.IO;

# Costruttori pubblici

|  |  |
| --- | --- |
| **Nome** | **Descrizione** |
| StreamReader | Sottoposto a overload. Inizializza una nuova istanza della classe StreamReader per il flusso specificato. |

# Campi pubblici

|  |  |
| --- | --- |
| **Nome** | **Descrizione** |
| Null | Oggetto **StreamReader** che include un flusso vuoto. |

# Proprietà pubbliche

|  |  |
| --- | --- |
| **Nome** | **Descrizione** |
| EndOfStream | Ottiene un valore che indica se la posizione corrente del flusso è alla fine del flusso. |

# Metodi pubblici

|  |  |
| --- | --- |
| **Nome** | **Descrizione** |
| Close | Sottoposto a override. Chiude l'oggetto **StreamReader** e il flusso sottostante e libera tutte le risorse di sistema associate al visualizzatore. |
| Peek | Sottoposto a override. Restituisce il successivo carattere disponibile senza utilizzarlo. |
| Read | Sottoposto a overload. Sottoposto a override. Legge il carattere successivo dal flusso di input. Il carattere è rappresentato come oggetto Int32 oppure -1, se non sono presenti altri caratteri disponibili. |
| Read (Char[], Int32, Int32) | Legge un numero massimo specificato di caratteri dal flusso corrente e scrive i dati in un buffer, iniziando dall'indice specifico. Valore restituito: Numero di caratteri letti oppure 0 se alla fine del flusso non è stato letto alcun dato. Il numero sarà minore o uguale al parametro count, a seconda che i dati siano disponibili o meno all'interno del flusso. |
| ReadLine | Sottoposto a override. Legge una riga di caratteri dal flusso corrente e restituisce i dati come stringa. |
| ReadToEnd | Sottoposto a override. Legge il flusso dalla posizione corrente fino alla fine. |

* **Costruttore StreamReader**
* StreamReader (String)

Inizializza una nuova istanza della classe StreamReader per il nome file specificato.

* StreamReader (Stream)

Inizializza una nuova istanza della classe StreamReader per il flusso specificato.

**StreamWriter**

Implementa un oggetto TextWriter che scrive i caratteri in un flusso con una particolare codifica.

# Costruttori pubblici

|  |  |
| --- | --- |
| **Nome** | **Descrizione** |
| StreamWriter | Sottoposto a overload. Inizializza una nuova istanza della classe StreamWriter. |

# Campi pubblici

|  |  |
| --- | --- |
| **Nome** | **Descrizione** |
| Null | Fornisce un oggetto **StreamWriter** privo di archivio di backup, in cui è possibile scrivere ma dal quale non è possibile leggere. |

# Proprietà pubbliche

|  |  |
| --- | --- |
| **Nome** | **Descrizione** |
| AutoFlush | Ottiene o imposta un valore che indica se **StreamWriter** invia i dati contenuti nel proprio buffer nel flusso sottostante dopo ogni chiamata a StreamWriter.Write. |

# Metodi pubblici

|  |  |
| --- | --- |
| **Nome** | **Descrizione** |
| Close | Sottoposto a override. Chiude l'oggetto **StreamWriter** corrente e il flusso sottostante. |
| Flush | Sottoposto a override. Cancella tutti i buffer relativi al writer corrente e consente la scrittura dei dati memorizzati nel buffer nel flusso sottostante. |
| Write | Sottoposto a overload. Scrive nel flusso. |
| WriteLine | Sottoposto a overload. Scrive alcuni dati, come specificato dai parametri di overload, seguiti da un terminatore di riga. (Ereditato da TextWriter) |

* **Costruttore StreamWriter**
* StreamWriter (String)

Inizializza una nuova istanza della classe StreamWriter per il file specificato nel percorso specificato, utilizzando le dimensioni di buffer e la codifica predefinite.

* StreamWriter (String, Boolean)

Inizializza una nuova istanza della classe StreamWriter per il file specificato nel percorso specificato, utilizzando le dimensioni di buffer e la codifica predefinite. Se il file esiste, è possibile sovrascriverlo o aggiungervi informazioni. Se il file non esiste, il costruttore crea un nuovo file.

Il valore Boolean determina se i dati devono essere aggiunti al file. Se il file esiste e append è false, il file viene sovrascritto. Se il file esiste e append è true, i dati vengono aggiunti. In caso contrario, viene creato un nuovo file.

**Esempio**

string mioFile = @"C:\TestFile.txt";

bool Aggiungo = false;

StreamWriter sw = new StreamWriter(mioFile, Aggiungo);

// Scrittura su file.

sw.Write("Titolo ");

sw.WriteLine("del file.");

sw.WriteLine("----------------");

sw.WriteLine();

sw.Write("Data e ora: ");

sw.WriteLine(DateTime.Now);

sw.WriteLine();

for (int i = 0; i < 10; i++)

sw.WriteLine("Riga n. {0}", i);

sw.Close();

// Lettura da File

Console.WriteLine("Lettura utilizzando null");

Console.WriteLine();

StreamReader sr = new StreamReader(mioFile);

String Riga;

while ((Riga = sr.ReadLine()) != null)

Console.WriteLine(Riga);

sr.Close();

// Lettura da File con EndOfStream

Console.WriteLine("Lettura utilizzando EndOfStream");

Console.WriteLine();

StreamReader sr = new StreamReader(mioFile);

String Riga = "";

while (! **sr.EndOfStream**)

{

Riga = sr.ReadLine();

Console.WriteLine(Riga);

}

sr.Close();

// Lettura da File con RigaToEnd

Console.WriteLine("Lettura con ReadToEnd");

Console.WriteLine();

StreamReader srToEnd = new StreamReader(mioFile);

String RigaToEnd;

RigaToEnd = srToEnd.ReadToEnd();

Console.WriteLine(RigaToEnd);

srToEnd.Close();

// Lettura da File con Read

Console.WriteLine("Lettura utilizzando Read");

Console.WriteLine();

StreamReader sr = new StreamReader(mioFile);

char Carattere;

while (!sr.EndOfStream)

{

Carattere = (char)sr.Read();

Console.WriteLine(Carattere);

}

sr.Close();

// Lettura da File con Read (Char[], Int32, Int32)

Console.WriteLine("Lettura utilizzando Read (Char[], Int32, Int32)");

Console.WriteLine();

StreamReader sr = new StreamReader(mioFile);

char [] Carattere = new char[5];

while (!sr.EndOfStream)

{

//Legge 5 caratteri alla volta

sr.Read(Carattere, 0, Carattere.Length);

Console.WriteLine(Carattere);

}

sr.Close();

**N.B. Per riutilizzare gli oggetti StreamReader e StreamWriter:**

//Per usare i caratteri accentati, speciali: System.Text.Encoding.Default

sr = new StreamReader(mioFile);

//Per usare i caratteri accentati, speciali: System.Text.Encoding.Default

sw = new StreamWriter(mioFile, Aggiungo);