Sử dụng TextReader, TextWriter trong C#

*TextReader và TextWriter là các lớp cơ sở. StreamReader và StringReader được kế thừa từ TextReader. Tương tự, StreamWriter và StringWriter được kế thừa từ TextWriter.*  
  
**StreamWriters và StreamReaders**:  
- StreamWriter kế thừa từ lớp cơ sở TextWriter. Lớp này định nghĩa các thành viên cho phép thu được loại dữ liệu để ghi văn bản cho một dòng ký tự nhất định.  
- Xét một vài hàm thành viên chính của lớp trừu tượng TextWriter.

* *Close() - Đóng Writer và giải phóng tài nguyên liên quan.*
* *Write() - Ghi một dòng vào luồng dữ liệu (text stream), không kèm theo ký tự xuống dòng.*
* *WriteLine() - Ghih một dòng vòa luồng dữ liệu, có ký tự xuống dòng ở cuối dòng '\n'.*
* *Flush() - Xóa tất cả các vùng đệm mà Writer đang nắm giữ.*

**Ghi một file văn bản:  
-**Bây giờ chúng ta thực hiện một ví dụ ghi một file văn bản Sử dụng StreamWriter.  
- Để đưa được dữ liệu vào văn bản, ta cần sử dụng lớp "FileInfo". Lớp này được khai báo trong không gian tên "System.IO". Chức năng của FileInfo là tóm lược một số chi tiết về tập tin đã có trên ổ cứng của bạn (kích thước, thuộc tính, thời gian được tạo ra, w) cũng như các trợ giúp trong việc tạo ra hoặc xóa bỏ các tập tin mới.  
- Trong ví dụ này, trong lớp WriteTextFile tôi tạo một tập tin có tên "vidu.txt" sử dụng lớp FileInfo. Sau khi tạo tập tin văn bản bằng cách sử dụng phương thức CreateText(), hàm này trả về kiểu StreamWriter và ta sẽ ghi một số thông tin lên nó bằng phương thức Write();

public class WriteTextFile

{

public static void Main(string[] args)

{

FileInfo t = new FileInfo("vidu.txt");

StreamWriter Tex =t.CreateText();

Tex.WriteLine("chào mừng bạn đến với Vubka Blog");

Tex.WriteLine("cùng nhau chia sẻ kiến thức chung");

Tex.Write(Tex.NewLine);

Tex.close();

Console.WriteLine("Dữ liêu đã được ghi xuống file vidu.txt");

}

}

Dữ liệu đã được ghi xuống file "vidu.txt";  
  
**Đọc file văn bản:**  
- Bây giờ, làm thế nào để đọc được dữ liệu văn bản sử dụng StreamReader.  
- Hãy xem các hàm thành viên chính trong lớp trừu tượng TextReader.

* Read() - Đọc dữ liệu từ luồng dữ liệu vào (input stream).
* ReadLine() - Đọc chuỗi ký tự từ luồng hiện tại và trả dữ liệu về kiểu string.
* ReadToEnd() - Đọc tất cả các chuỗi ký tự cho đến cuối của TextReader và trả chúng về là một chuỗi kiểu string.

public class ReadTextFile

{

public static int Main(string[] args)

{

StreamReader re = File.OpenText("vidu.txt");

string input = null;

while ((input = re.ReadLine()) != null)

{

Console.WriteLine(input);

}

re.close;

return 0;

}

}

Trong lớp ReadTextFile trên, tôi đã mở file văn bản "vidu.txt" và đọc nội dung của nó sử dụng phương thức ReadLine(). Trong cả hai StreamReader và StreamWriter đều có liên quan với việc di chuyển dữ liệu văn bản đến và đi từ một tập tin chỉ định.  
  
**StringWriters và StringReaders:**- Sử dụng StringReaders và StringWriters có thể xử lý thông tin văn bản như một dòng các chuỗi ký tự trong bộ nhớ. Trong này chúng ta có thể chèn hoặc loại bỏ các chuỗi giữa một khối dữ liệu văn bản. Xem ví dụ dưới đây, trong đó có thêm và xóa một số chuỗi giữa một khối văn bản.

public class StringWrite

{

public static int Main(string[] args)

{

StringWriter wr = new StringWriter();

wr.WriteLine("Friendship is not a two way road.");

wr.WriteLine("It is a one way road travelled by two people.");

wr.Write(Writer.NewLine);

wr.close();

Console.WriteLine("Data: {0}",wr.ToString());

return0;

}

}

Chạy chương trình trên ta có được đoạn văn bản dữ liệu như trong giao diện console dưới đây:

Friendship is not a two way road.

It is a one way road travelled by two people.

Press any key to continue

Bây giờ thử chèn và xóa một số chuỗi giữa các dữ liệu văn bản.

using System.Text;

public class StringWrite

{

public static int Main(string[] args)

{

StringWriter wr = new StringWriter();

wr.WriteLine("Friendship is not a two way road.");

wr.WriteLine("It is a one way road travelled by two people.");

wr.Write(Writer.NewLine);

wr.close();

StringBuilder bu = wr.GetStringBuilder();

string entiredata = bu.ToString();

Console.WriteLine("The data:{\n0}",entiredata);

// Thêm.

bu.Insert(45,"together-hand in hand");

entiredata = bu.ToString();

Console.WriteLine("The modified data:\n{0}",entiredata);

// Xóa.

bu.Remove(45."together-hand in hand".Length);

entiredata = bu.ToString

Console.WriteLine("The original data:\n{0}",entiredata);

return 0;

}

}

Đoạn chương trình trên ghi một vài dữ liệu tới StringWriter và chỉnh sử dữ liệu bằng cách chèn một chuỗi tại vị trí 45. Sau đó lại remove đoạn dữ liệu vừa thêm vào, kết quả chương trình như sau:

Friendship is not a two way road.

It is a one way road travelled by two people.

The modified data:

Friendship is not a two way road.

It is a one way road travelled by two people together-hand in hand.

The original data:

Friendship is not a two way road.

It is a one way road travelled by two people.

Bây giờ hãy dùng StringReader để đọc khối dữ liệu chứ không phải là toàn bộ tập tin.

StringReader re = new StringReader(wr.ToString());

string input = null;

while(( input = re.ReadLine()) != null)

{

Console.WriteLine(input);

}

re.close();

Giống như cách ở trên bằng cách sử dụng StringReader ta có thể đọc dữ liệu từ một tập tin.  
  
Hy vọng qua bài này, bạn có được một số thông tin về cách tạo, viết và đọc dữ liệu văn bản bằng cách sử dụng TextReader và TextWriter trong C#.

0 Comments: