

Files/FileSystem

Потоки данных. Работа с файлами

В .NET Framework - существует механизм обработки данных, не зависящий от источника. Этот механизм основан на понятии потока (stream), поскольку все данные могут быть представлены в виде потока байтов. Поток - это сущность, допускающая операции чтения и записи, например файлы, данные, получаемые по сети, данные в области памяти.

Класс Stream

Большинство операций ввода-вывода реализованы в пространстве имен SystemIO. Для выполнения этих операций используется абстрактный класс Stream.

Основные классы, производные от Stream

Класс

Описание

14. Файлы - ITYourself! - компьютерная документация, книги, мануалы, it-тематика

Автор: Администратор
27.02.2009 14:44

FileStream

Доступ к файлам

MemoryStream

Блок памяти (часто используется при работе с буферами)

NetworkStream

Данные, полученные в виде потока по сетевому соединению

Основные свойства и методы класса Stream:

Метод / свойство

Описание

CanRead

Признак поддержки чтения

CanWrite

Признак поддержки записи

CanSeek

Признак поддержки перемещения в потоке

Read

Чтение заданного количества байтов, начиная с текущей позиции. Возвращает количество успешных байтов.

ReadByte

Чтение отдельного байта. Если текущая позиция находится в конце потока, возвращает -1.

Write

Запись заданного количества байтов начиная с текущей позиции

14. Файлы - ITYourself! - компьютерная документация, книги, мануалы, it-тематика

Автор: Администратор
27.02.2009 14:44

WriteByte

Запись отдельного байта

Length

Длина потока в байтах

Position

Позиция в текущем потоке

Flush

Запись данных и очищение всех буферов, используемых потоком

Close

Закрывает поток и освобождает используемые ресурсы

Класс FileStream

Файл - основная единица файловой системы, предназначенная для хранения информации. При открытии для чтения/записи файл становится потоком.

Класс FileStream используется для открытия, чтения, записи и закрытия файлов. Для открытия файла или создания нового нужно создать объект класса FileStream

```
FileStream(ByVal strFileName As String, ByVal fm As FileMode, ByVal fa As FileAccess, ByVal fs As FileShare)
```

где, strFileName определяет имя файла;

FileMode определяет операцию с файлом;

Перечисление FileMode

Значение

Описание

Автор: Администратор
27.02.2009 14:44

CreateNew

Создать файл. Ошибка, если файл существует.

Create

Создать файл. Если файл существует, его содержимое удаляется

Open

Открыть файл. Ошибка, если файл не существует

OpenOrCreate

Открыть файл. Если файла не существует, создать

Append

Поместить указатель в конец файла. Создать файл если не существует

FileAccess указывает режим доступа к файлу;

Перечисление FileAccess

Значение

Описание

Read

Открыть для чтения

Write

Открыть для записи

ReadWrite

Открыть для чтения и записи (по умолчанию)

FileShare позволяет указать режим совместного доступа к файлу.

14. Файлы - ITYourself! - компьютерная документация, книги, мануалы, it-тематика

Автор: Администратор
27.02.2009 14:44

Перечисление FileShare

Значение

Описание

None

Запретить доступ других процессов к файлу (по умолчанию)

Read

Разрешить другим процессам чтение из файла

Write

Разрешить другим процессам запись в файл

ReadWrite

Полный доступ к файлу

Основные методы и свойства класса FileStream

Метод / свойство

Описание

Handle

Файловый манипулятор операционной системы для файла

Position

Текущая позиция в файле. Нумерация позиций начинается с нуля. Перейти в конец файла мож

14. Файлы - ITYourself! - компьютерная документация, книги, мануалы, it-тематика

Автор: Администратор
27.02.2009 14:44

Lock

Блокирует доступ к файлу или его части со стороны других процессов

Seek

Устанавливает указатель текущей позиции в заданное положение. Точка отсчета определяется

Unlock

Снимает блокировку с ранее заблокированной части файла

Свойство `Length` и методы `Write`, `WriteByte`, `Read`, `ReadByte`, `Close` аналогичны методам класса `Stream`. Свойство `CanSeek` всегда равно `True` для открытых файлов.

Класс `Environment`

Класс `Environment` позволяет получать значения переменных среды. Методы `GetLogicalDrives` и `GetFolderPath` возвращают информацию о файловой системе.

Свойство `CurrentDirectory` позволяет выяснить или установить текущий диск и каталог для приложения. При определении текущего каталога можно задавать относительный путь, включая строку «`..`», указывающую на родительский каталог.

Класс `Path`

Класс `Path` предназначен для обработки имен файлов

Основные методы и свойства класса Path

Свойство / метод

Описание

DirectorySeparatorChar

Символ-разделитель путей для текущей платформы

InvalidPathChars

Массив символов, недопустимых в имени файла

Автор: Администратор
27.02.2009 14:44

ChangeExtension

Изменяет расширение файла и возвращает новое имя

GetDirectoryName

Возвращает путь к каталогу, в котором находится файл

GetFileName

Возвращает имя файла

GetExtension

Возвращает расширение файла

GetFileNameWithoutExtension

Возвращает имя файла без расширения

Combine

Объединяет путь с путем и/или именем файла

Работа с каталогами и файлами

В VB .NET существуют два класса для работы с каталогами: Directory и DirectoryInfo и два класса для работы с файлами: File и FileInfo.

Классы `Directory` и `File` являются производными от `Object`, а классы `DirectoryInfo` и `FileInfo` от класса `FileSystemInfo` и содержат обычные методы, поэтому обращение к их членам происходит через конкретные экземпляры.

Члены классов `Directory` и `File` возвращают строки с описанием каталогов или файлов, а члены классов `DirectoryInfo` и `FileInfo` экземпляры своих классов.

Основные методы класса `Directory`

Метод

Описание

`CreateDirectory`

Создает каталог с заданным именем и возвращает объект `DirectoryInfo` для созданного каталога

14. Файлы - ITYourself! - компьютерная документация, книги, мануалы, it-тематика

Автор: Администратор
27.02.2009 14:44

Delete

Удаляет пустой каталог. Для удаления папки, содержащей файлы или вложенные папки, след

Exists

Возвращает признак существования каталога

GetCreationTime

Возвращает дату и время создания каталога

GetCurrentDirectory

Автор: Администратор
27.02.2009 14:44

Возвращает полный путь к текущему каталогу

GetDirectories

Возвращает массив строк с описанием подкаталогов.

GetFiles

Возвращает массив строк с описаниями файлов каталога.

GetLastAccessTime

Возвращает время последнего обращения к каталогу

GetLastWriteTime

14. Файлы - ITYourself! - компьютерная документация, книги, мануалы, it-тематика

Автор: Администратор
27.02.2009 14:44

Возвращает время последней записи в каталог

GetLogicalDrives

Возвращает строковый массив с именами логических дисков в формате «диск:\» (например, C:\)

Move

Перемещает каталог

SetCurrentDirectory

Задает текущий каталог

Основные методы класса File

Метод

Описание

Сору

Копирование файла

GetAttributes

Возвращает установленные атрибуты файла (архивный, скрытый и т.д.)

Автор: Администратор
27.02.2009 14:44

SetAttributes

Задаёт атрибуты файла

Методы Exists, Delete, GetCreationTime, GetLastAccessTime, GetLastWriteTime, Move аналогичны методам класса Directory

Класс FileSystemInfo является базовым для классов DirectoryInfo и FileInfo и содержит большую часть их общей функциональности.

Основные методы и свойства класса FileSystemInfo

Метод / свойство

Описание

14. Файлы - ITYourself! - компьютерная документация, книги, мануалы, it-тематика

Автор: Администратор
27.02.2009 14:44

Attributes

Атрибуты объекта

CreationTime

Время создания объекта

Extension

Расширение файла

FullName

Автор: Администратор
27.02.2009 14:44

Как правило, при создании экземпляров классов DirectoryInfo и FileInfo указывается имя каталога или файла. Пример:

```
Dim myDirectory As DirectoryInfo
```

```
myDirectory = New DirectoryInfo("C:TestDirectory")
```

```
Dim currentDir As New DirectoryInfo(".")
```

Основные методы и свойства класса DirectoryInfo

14. Файлы - ITYourself! - компьютерная документация, книги, мануалы, it-тематика

Автор: Администратор
27.02.2009 14:44

Метод / свойство

Описание

Exists

Логический признак существования каталога

Name

Имя каталога

Parent

Автор: Администратор
27.02.2009 14:44

Объект DirectoryInfo для родительского каталога (для корневых каталогов возвращается Nothing)

Create

Создает каталог

Delete

Удаляет пустой каталог

GetDirectories

Возвращает массив объектов DirectoryInfo для подкаталогов

GetFiles

Возвращает массив объектов FileInfo для файлов текущего каталога

Основные методы и свойства класса FileInfo

Метод / свойство

Описание

Directory

Объект DirectoryInfo для каталога, в котором находится файл

DirectoryName

Полный путь к файлу в строковом виде

CopyTo

Копирует файл и возвращает объект FileInfo для копии

Create

Создает файл по имени

MoveTo

Перемещает файл

Типы файлов

Файлы бывают трех типов:

1. Текстовые файлы (файлы последовательного доступа). Состоят из строк разной длины, отделенных друг от друга двумя специальными символами - возврат каретки и перевод строки.
2. Типизированные (двоичные) файлы (файлы произвольного (прямого) доступа). Состоят из набора записей одинаковой длины.
3. Легализированные файлы. Представлены в виде потока байтов, не объединенных ни в какие логические структуры.

Работа с текстовыми файлами

В первую очередь, файл необходимо открыть, то есть связать с каким-то целым числом - файловым номером при помощи процедуры FileOpen:

```
FileOpen(filename, filename, workmode)
```

где, filename - это имя файла на диске,

filename - целый положительный файловый номер,

workmode - режим работы (OpenMode.Input (для чтения), OpenMode.Output (для записи), OpenMode.Append (для добавления)).

Чтение из файла:

Функция LineInput(filename) считывает очередную строку из входного файла.

Процедура `Input(filenumber, variable)` считывает значения, разделенные запятыми.

Запись в файл:

Процедуры `Print(filenumber, var)` и `PrintLine(filenumber, var)` означают запись строки из переменной `var` в файл с номером `filenumber`. `PrintLine` отличается от `Print` тем, что вставляет после значения `var` возврат каретки и перевод строки.

Для закрытия файла используется процедура `FileClose(filenumber)`.

Функция `EOF(filenumber)` возвращает `True`, если достигнут конец файла.

В пространстве имен `SystemIO` определены специальные классы для чтения и записи текстовых файлов: `StreamReader` и `StreamWriter`.

Files/BinaryFiles

Работа с типизированными файлами

Типизированные файлы или файлы прямого доступа состоят из набора однотипных записей, что упрощает поиск и перемещение по файлу. Можно прочитать, изменить, добавить любую запись. Например в файле может содержаться такая информация о сотрудниках предприятия как ФИО, дата рождения, зарплата.

Чтобы открыть типизированный файл необходимо в процедуре FileOpen указать режим работы OpenMode.Random (чтение и запись) и задать значение параметра RecordLength (размер записи в байтах):

```
FileOpen(filename, filename, workmode , , RecordLength)
```

Чтение из файла: процедура FileGetObject(filename,var,recnumber)

Запись в файл: процедура FilePutObject(filename,var,recnumber)

где resnumber - номер записи. Нумерация начинается с 1, по умолчанию операция производится с текущей записью.

Функция Space(K) возвращает строку, состоящую из K пробелов. Чтобы дополнить строку S до длины N, используется выражение $S + \text{Space}(N - \text{Len}(S))$

Функция LOF возвращает длину файла в байтах. Разделив полученное значение на размер записи можно получить количество записей.

Для чтения и записи двоичных файлов определены специальные классы BinaryReader и BinaryWriter.

Работа с нетипизированными файлами

Работа с нетипизированными файлами очень похожа на работу с файлами прямого доступа. В качестве режима работы при открытии файла указывается `OpenMode.Binary`, а чтение и запись производится при помощи процедур `FileGetObject` и `FilePutObject`. При этом значение параметра `recordnumber` означает не номер записи, а смещение в байтах от начала файла. Самая первая запись имеет смещение 1. Чтобы прочесть какую-либо запись, необходимо прочесть все записи, предшествующие требуемой.

Все файлы можно рассматривать как нетипизированные. Следует использовать данный тип файлов в том случае, если время доступа не имеет большого значения, а расход места на диске существенен.