

Лекция 22. Чтение и запись данных в файл

Курс «Программирование»
ПрИнф & МОиАИС, 2 семестр
Щукин Александр
Валентинович

Варианты

1. Текстовый файл
 - а. Строковый
 - б. Со структурой (например, xml)
2. Бинарный файл
 - а. Сериализация - процесс перевода какой-либо структуры данных в последовательность бит
 - б. Ручное управление

Namespace System.IO

1. Файловые операции
2. Трансфер (чтение \ запись) данных из файла и в файл и из памяти и в память
3. Синхронно и асинхронно
4. Компрессия файлов
5. ...

Файлы (file) и потоки (stream)

Концепция потока в .Net не ограничена файловым вводом-выводом. Потоки могут предоставлять доступ к сетям, областям памяти, прочим абстракциям, связанными с потоками.

Классы

Класс	Назначение
BinaryReader BinaryWriter	Запись и чтение данных (числовые, булевские, даты...) в двоичном виде
FileInfo	Манипуляции с файлами, в том числе открытие для чтения или записи
FileStream	Произвольный доступ к файлу (с возможностью поиска) с данными представленными в виде потока байт
MemoryStream	Произвольный доступ к памяти
StreamReader StreamWriter	Чтение и запись текстовой информации из файла
StringReader StringWriter	Чтение и запись текстовой информации из файла с использованием строкового буфера

Класс FileInfo

Метод	Описание
Exist (свойство)	Проверяет существование файла
Open	Открытие файла с полным набором параметров. Возвращает FileStream
CreateText AppendText OpenText OpenRead OpenWrite	Возвращает StreamWriter или StreamReader для выполнения соответствующих операций с текстовым файлом
CopyTo Create Delete MoveTo	Выполняют операции с файлом (не с содержимым) в– скопировать, создать, удалить, переместить
OpenRead OpenWrite	Возвращает FileStream для выполнения соответствующих операций с содержимым файла
Name, Length, DirectoryName	Свойства с соответствующей информацией

Создание файла и работа с ним

1. Обработка ошибок.
2. При существовании файла он будет удален.
3. Метод Create возвращает ссылку на FileStream.

```
FileInfo fi = new FileInfo("d:/temp/log.txt");  
StreamWriter sw = fi.CreateText();
```

```
sw.WriteLine("Hello world!");  
sw.WriteLine("Another line!");
```

```
sw.Close();
```

```
Console.WriteLine("Done!");  
Console.ReadLine();
```

Альтернативный метод - FileInfo.Open()

1. FileMode – операции при создании объекта.
2. FileAccess – тип доступа для себя.
3. FileShare – тип доступа для других.

```
FileStream fs;  
fs = fi.Open(FileMode.OpenOrCreate, FileAccess.Write, FileShare.Read);
```

Перечисления

1. FileMode – Append, Create, CreateNew, Open, OpenOrCreate, Truncate
2. FileAccess – Read, ReadWrite, Write
3. FileShare – Delete, Inheritable, None, Read, ReadWrite, Write

Классы StreamWriter, StreamReader

1. Удобны при работе с текстовыми файлами
2. Кодировка текста по умолчанию Unicode. Но это можно изменить используя `System.Text.Encoding`
3. StreamWriter наследует основные члены от `TextWriter`, StreamReader от `TextReader`

Класс StreamReader

Метод	Описание
Read	Чтение символа, символов
ReadLine	Чтение строки
ReadToEnd	Чтение до конца файла
ReadBlock	Чтение блока символов
Peek	Взять символ не перемещая курсор в файле
ReadAsync ReadLineAsync ReadToEndAsync	Асинхронное чтение
Close	Закрывает SR и освобождает все ресурсы
EndOfStream (свойство)	Конец потока
CurrentEncoding (свойство)	Текущая кодировка

Чтение текстового файла построчно

OpenText() или OpenRead()

```
StreamReader sr;  
string strText;  
sr = fi.OpenText();  
  
while (sr.EndOfStream == false)  
{  
    strText = sr.ReadLine();  
    Console.WriteLine(strText);  
}
```

Запись в бинарный файл

1. Используется `BinaryWriter`
2. Тип записываемых данных и порядок записи важен и при чтении

```
int[] arrNums = { 2, 4, 6, 8 };  
// Входной аргумент конструктора - Stream  
BinaryWriter bw = new BinaryWriter(File.Open("d:/temp/log.bin", FileMode.Create));  
foreach (int x in arrNums)  
{  
    bw.Write(x);  
}  
bw.Close();
```

Класс File

1. Статические члены
2. Плюс к тем методам, аналоги которых есть в `FileInfo` есть уникальные методы
 1. `ReadAllBytes` – результат – массив байт
 2. `ReadAllLines` - результат – массив строк
 3. `ReadAllText` - результат – строка
 4. `WriteAllBytes`
 5. `WriteAllLines`
 6. `WriteAllText`