Визуальное программирование

ЛЕКЦИЯ 9

Содержание лекции

01 Работа с дисками

02 Работа с каталогами

03 Работа с файлами

04 Чтение и запись

РАБОТА С ДИСКАМИ

Класс DriveInfo

Для работы с дисками существует класс DriveInfo. Он содержится в пространстве имен System.IO.

Этот класс содержит несколько важных методов, при помощи которых можно получать множество полезной информации о дисках. DriveInfo можно использовать для того, чтобы определить доступность дисков, их емкость и свободное место.

Класс DriveInfo

Свойства класса DriveInfo:

- AvailableFreeSpace - Указывает объем доступного свободного места на диске в байтах
- DriveFormat - Получает имя файловой системы, например NTFS или FAT32
- DriveType - Получает тип диска, такой как компакт-диск, съемный, сетевой или жесткий
- IsReady - Получает значение, указывающее, готов ли диск
- Name - Возвращает имя диска, например C:\
- RootDirectory - Возвращает корневой каталог диска

Класс DriveInfo

Свойства класса DriveInfo:

- TotalFreeSpace - Возвращает общий объем свободного места, доступного на диске, в байтах
- TotalSize - Получает общий размер места для хранения на диске в байтах
- VolumeLabel - Получает или задает метку тома диска

Методы класса DriveInfo:

- GetDrives() - Возвращает имена всех логических дисков на компьютере

Класс Environment

Класс Environment позволяет получить различную информацию, относящуюся к операционной системе.

- CurrentDirectory - Возвращает или задает полный путь к текущей рабочей папке
- OSVersion - Возвращает объект OperatingSystem, который содержит идентификатор текущей платформы и номер версии
- SystemDirectory - Возвращает полный путь к системному каталогу
- UserName - Возвращает имя пользователя, сопоставленное с текущим потоком
- MachineName - Возвращает имя NetBIOS данного локального компьютера

РАБОТА С КАТАЛОГАМИ

Классы Directory и DirectoryInfo

Классы Directory и DirectoryInfo содержат в себе методы, позволяющие создавать, копировать, получать важную информацию, а так же удалять каталоги.

Их различие в том, что Directory – статический класс, предоставляющий статические методы для работы с директориями. DirectoryInfo – требует создания объекта и предоставляет информацию о каком-либо конкретном каталоге.

Класс Directory

Методы класса Directory:

- CreateDirectory(String) - Создает все каталоги и подкаталоги по указанному пути, если они еще не существуют
- Delete(String) - Удаляет пустой каталог по заданному пути
- Exists(String) - Определяет, указывает ли заданный путь на существующий каталог на диске
- GetCurrentDirectory() - Получает текущий рабочий каталог приложения

Класс Directory

Методы класса Directory:

- GetDirectories(String) - Возвращает имена подкаталогов (включая пути) в указанном каталоге
- GetFiles(String) - Возвращает имена файлов (с указанием пути к ним) в указанном каталоге
- Move(String, String) - Перемещает файл или каталог со всем его содержимым в новое местоположение

Класс DirectoryInfo

Методы и свойства класса DirectoryInfo:

- Create() - Создает каталог
- Delete() - Удаляет каталог, если он пуст
- Exists - позволяет узнать, существует ли каталог
- GetDirectories() - Возвращает подкаталоги текущего каталога

Класс DirectoryInfo

Методы и свойства класса DirectoryInfo:

- GetFiles() - Возвращает список файлов текущего каталога
- MoveTo(String) - Перемещает экземпляр DirectoryInfo и его содержимое в местоположение, на которое указывает новый путь
- Parent - Получает родительский каталог

РАБОТА С ФАЙЛАМИ

Класс File

Класс File можно использовать для стандартных операций копирования, перемещения, переименования, создания, открытия, удаления и добавления в один файл за раз.

Класс File предоставляет статические методы для создания, копирования, удаления, перемещения и открытия одного файла.

Класс File

Методы класса File:

- Copy(String, String) - Копирует существующий файл в новый файл. Перезапись файла с тем же именем не разрешена
- Create(String) - Создает или перезаписывает файл в указанном пути
- Exists(String) - Определяет, существует ли заданный файл
- Move(String, String) - Перемещает заданный файл в новое местоположение и разрешает переименование файла

Класс FileInfo

Класс FileInfo предоставляет свойства и методы экземпляра для создания, копирования, удаления, перемещения и открытия файлов.

Его методы и свойства так же очень похожи на класс File.

Класс FileInfo

Свойства класса FileInfo:

- Length - Получает размер текущего файла в байтах
- Name - Возвращает имя файла
- DirectoryName - Получает строку, представляющую полный путь к каталогу
- Exists - Получает значение, показывающее, существует ли файл
- Extension - Получает строку, содержащую расширение файла

Класс FileInfo

Методы класса FileInfo:

- Create() - Создает файл
- Delete() - Удаляет файл без возможности восстановления
- MoveTo(String) - Перемещает заданный файл в новое местоположение и разрешает переименование файла
- CopyTo(String) - Копирует существующий файл в новый файл и запрещает перезапись существующего файла

ЧТЕНИЕ И ЗАПИСЬ

Чтение и запись

В .NET существует целый набор классов предназначенных для чтения и записи в файлы. Все они имеют немного отличающиеся друг от друга цели и логику работы.

Эти классы позволяют работать как с текстовыми данными, так и с бинарным форматом.

Класс FileStream

Класс FileStream предоставляет возможность чтения и записи как в текстовые так и в бинарные файлы.

Экземпляры этого класса можно создавать как напрямую через конструкторы, так и через некоторые методы класса File, например, File.OpenRead(filename).

Класс FileStream

Методы класса FileStream:

- Append – Открывает существующий, или создает. Если файл существует, то помещает новый текст в конец файла.
- Create – Создает новый файл или перезаписывает старые
- CreateNew – создает новый файл, если такой файл уже существует, то выбрасывает IOException
- Open – открывает существующий файл. Если файл не существует – вызывается исключение
- OpenOrCreate – открывает файл или создает новый
- Truncate – открывает существующий файл, и очищает его. Новый не создается

Класс FileStream

При создании объекта FIleStream при помощи класса File нужно указывать параметр FileMode, который отвечает за режим открытия.

```
string filename = @"D:\Users\MyFile.txt";
FileStream fs = File.Open(filename, FileMode.OpenOrCreate);
```

Чтение из файла

Read(Byte[] array, Int32 offset, Int32 count) - Выполняет чтение блока байтов из файла в массив байт.

- array Byte[] - массив в который будут помещены считанные байты
- offset Int32 - Смещение в байтах в массиве array, в который будут помещены считанные байты
- count Int32 - Максимальное число байтов, которые будут считаны

Запись в файл

Write (byte[] array, int offset, int count) - записывает в файл данные из массива байтов.

- array Byte[] - Массив, предназначенный для записи
- offset Int32 - Смещение байтов (начиная с нуля) массива array, с которого начинается копирование байтов в поток
- count Int32 - Максимальное число байтов для записи

Класс StreamReader

Класс StreamReader удобно использовать для чтения из текстовых файлов. Он считывает символы из потока байтов в определенной кодировке.

Класс StreamReader

Методы класса StreamReader:

- Close() - Закрывает объект StreamReader и основной поток и освобождает все системные ресурсы, связанные с устройством чтения
- Peek() - Возвращает следующий доступный символ, но не использует его
- Read() - Выполняет чтение следующего символа из входного потока и перемещает положение символа на одну позицию вперед
- ReadLine() - Выполняет чтение строки символов из текущего потока и возвращает данные в виде строки
- ReadToEnd() - Считывает все символы, начиная с текущей позиции до конца потока

Класс StreamWriter

Класс StreamWriter используется для записи в текстовые файлы.

Этот класс предназначен для вывода символов в определенной кодировке, тогда как классы, производные от Stream, предназначены для ввода и вывода байтов.

Класс StreamWriter

Методы класса StreamWriter:

- Close() - Закрывает текущий объект StreamWriter и базовый поток
- Write(String) - Записывает в поток строку
- WriteLine(Char[]) - Записывает в текстовый поток массив символов, за которыми следует признак конца строки

Класс BinaryWriter

Класс BinaryWriter записывает примитивные типы в двоичный поток и поддерживает запись строк в заданной кодировке. Он предоставляет методы, упрощающие запись примитивных типов данных в поток.

Например, можно использовать метод Write для записи логического значения в поток в виде однобайтового значения.

Класс включает методы записи, поддерживающие различные типы данных.

Класс BinaryWriter

Методы класса BinaryWriter:

- Close() - Закрывает текущий BinaryWriter и базовый поток
- Write(String) - Записывает в текущий поток строку, предваряемую ее длиной, используя текущую кодировку BinaryWriter, и перемещает позицию в потоке вперед в соответствии с используемой кодировкой и количеством записанных в поток символов

Класс BinaryReader

Класс BinaryReader нужен для чтения данных в бинарном формате.

Класс BinaryReader предоставляет методы, упрощающие чтение примитивных типов данных из потока.

Например, можно использовать метод ReadBoolean для считывания следующего байта в качестве логического значения.

Класс BinaryReader

Методы класса BinaryReader:

- Close() - Закрывает текущий поток чтения и связанный с ним базовый поток
- ReadBoolean() - Считывает значение Boolean из текущего потока и перемещает текущую позицию в потоке на один байт вперед
- ReadByte() - Считывает из текущего потока следующий байт и перемещает текущую позицию в потоке на один байт вперед

Класс BinaryReader

Методы класса BinaryReader:

- ReadInt32() - Считывает целое число со знаком длиной 4 байта из текущего потока и перемещает текущую позицию в потоке на четыре байта вперед
- ReadDouble() - Считывает число с плавающей запятой длиной 8 байт из текущего потока и перемещает текущую позицию в потоке на восемь байт вперед
- ReadString() - Считывает строку из текущего потока. Строка предваряется значением длины строки, которое закодировано как целое число блоками по семь битов

ВОПРОСЫ ПО ЛЕКЦИИ