



Работа с файлами в C#

Мухортова Н.Н.



Цель

Познакомиться с библиотекой ввода-вывода System.IO

Освоить способы работы с файлами



System.IO

Всю работу с файлами можно разделить на 2 группы:

- 1) это работа с файлами, как с элементами файловой системы, например, найти файл, получить список файлов в директории, узнать дату изменения файла, различные его атрибуты, расширение, скопировать или удалить файл, создать новый.
- 2) это работа с содержимым файла: прочитав файл или записать в него что-нибудь.



System.IO

Существует 2 класса `File` и `FileInfo`, которые предназначены для работы с файлом, как с частью файловой системы.

Здесь также есть несколько методов, которые позволяют работать с содержимым файла целиком.

В простых случаях (когда размеры файла небольшие, скажем до 1МБ) файл можно полностью прочитать и сохранить в память.



Некоторые методы по работе с файлами

`ReadAllText()`. Самая простая операция — это прочитать файл целиком. Данный метод является статическим. Здесь в качестве параметра можно указать полный путь к файлу. Когда указываем путь к файлу, не забываем экранировать символ "\".

`ReadAllLines()`. Этот статический метод возвращает массив строк, т.е. читает файл построчно.

`WriteAllText()`. Здесь в качестве параметра передаём строку, и эта строка целиком добавляется в файл с перезаписью содержимого файла.

`AppendAllText()`. Здесь содержимое добавляется поверх существующего файла, без его перезаписи.



File

Предоставляет статические методы для создания, копирования, удаления, перемещения и открытия одного файла, а также помогает при создании объектов `FileStream`.

`AppendAllLines(String, IEnumerable<String>, Encoding)`

Добавляет строки в файл, используя заданную кодировку, затем закрывает файл. Если указанный файл не существует, этот метод создает файл, записывает в него указанные строки и затем закрывает файл.

`AppendText(String)`

Создает объект `StreamWriter`, добавляющий текст с кодировкой UTF-8 в существующий файл, или в новый файл, если указанный файл не существует.



File

`Copy(String, String, Boolean)`

Копирует существующий файл в новый файл. Перезапись файла с тем же именем разрешена.

`CreateText(String)`

Создается или открывается файл для записи текста в кодировке UTF-8. Если файл уже существует, его содержимое перезаписывается.

`Move(String, String)`

Перемещает заданный файл в новое местоположение и разрешает переименование файла.



FileInfo

Класс предоставляет следующие свойства, позволяющие получить сведения о файле. Пример использования каждого свойства см. в разделе страницы свойств.

`Directory` Свойство получает объект, представляющий родительский каталог файла.

`DirectoryName` Свойство получает полный путь к родительскому каталогу файла.

`Exists` Свойство проверяет наличие файла перед его обработкой.

`IsReadOnly` Свойство извлекает или задает значение, указывающее, можно ли изменить файл.

`Length` Возвращает размер файла.

`Name` Возвращает имя файла.



FileInfo

`AppendText()`

Создает `StreamWriter`, который добавляет текст в файл, представленный этим экземпляром `FileInfo`.

`CopyTo(String)`

Копирует существующий файл в новый файл и запрещает перезапись существующего файла.

`CopyTo(String, Boolean)`

Копирует существующий файл в новый файл и разрешает перезапись существующего файла.



FileInfo

Create()

Создает файл.

CreateAsSymbolicLink(String)

Создает символическую ссылку, расположенную в FullName , указывающую на указанный объект pathToTarget .

(Унаследовано от FileSystemInfo)



Выводы

То, что можно сделать с помощью класса "File", можно сделать и с помощью класса "FileInfo", и наоборот.

Методы класса "File" — статические, а методы класса "FileInfo" являются методами объекта. При работе с методами класса "File" первым параметром всегда указывается путь до файла.

В случае "FileInfo" — это имя указывается один раз в конструкторе при создании объекта. Если нужно выполнить разовую операцию, то лучше использовать класс "File". В случае работы со многими файлами лучше использовать "FileInfo".