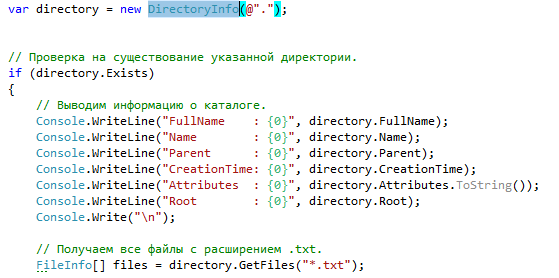
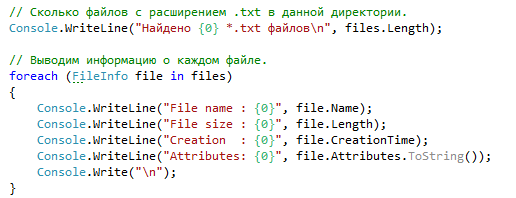
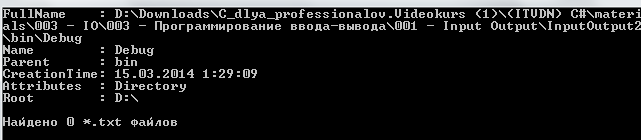


Если Parent отсутствует, свойство возвращает пустую строку

В данном примере получаем список файлов \*txt и выводим информацию о них

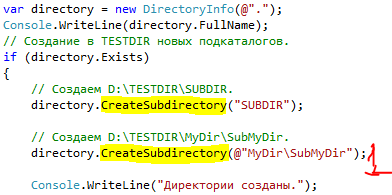






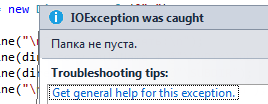
Создание директории CreateSubdirectory

1.создаст две папки (если папка существует то создана будет только вложенная)

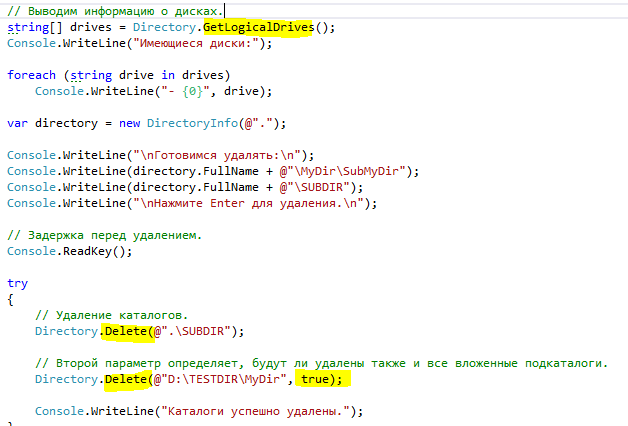


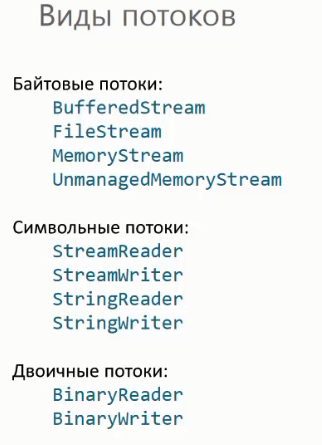
Удаление директории

Если в папке имеются подпапки или файлы то она удалена не будет и удаление вызовет ошибку



В связи с этим вторым параметром следует передать true - Directory.Delete(@".\SUBDIR",true); (для удаления всего находящегося в папке)

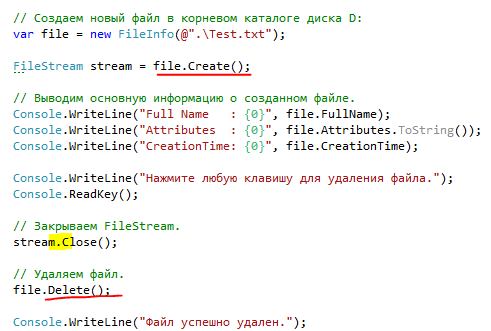




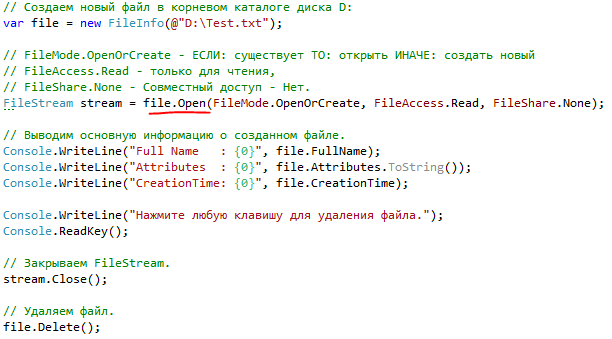
FileInfo и FileStream- байтовый поток (непосредственный ввод ввывод в физический файл на диске)

FileStream stream = file.Create(); - создает файл

FileStream – байтовый поток



Другой способ создания файла

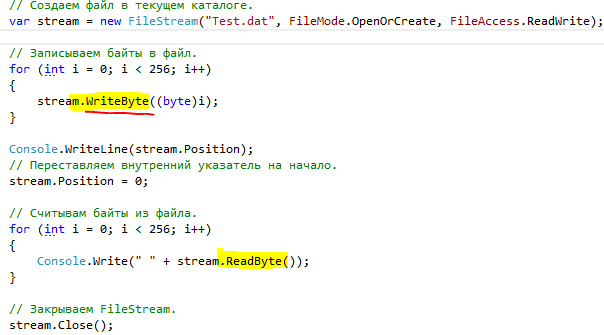


Запись и Чтение (байт) в файл

stream.WriteByte((byte)i); - запись

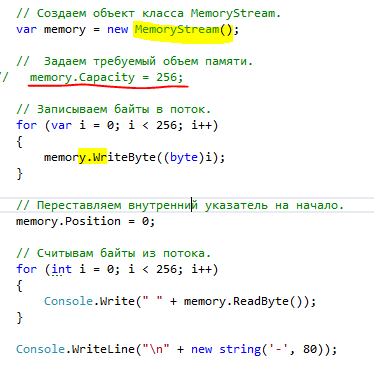
stream.ReadByte() – чтение (возвращает -1 если читать нечего)

stream.Position = 0; - переставить внутренний указатель на начало файла

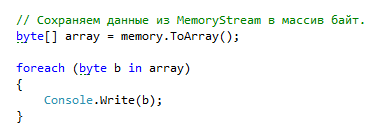


MemoryStream – байтовый поток

(создает некую область в управляемой куче куда будет осуществлять запись и чтение данных)

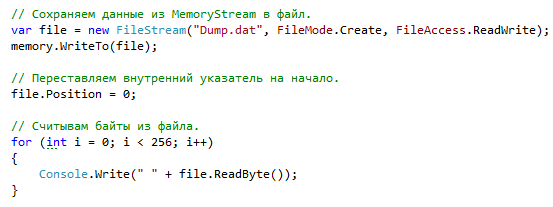


memory.ToArray();-Сохраняет из MemoryStream в байтовый массив



memory.WriteTo(file); - принимает любой поток наследуемый от Stream

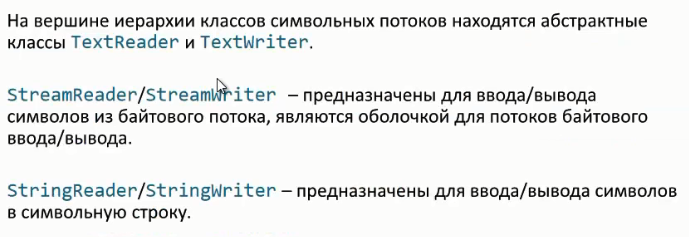
Сохраняет из MemoryStream данные в файл



ОБЯЗАТЕЛЬНО НУЖНО ЗАКРЫВАТЬ КАК MemoryStream так FileStream

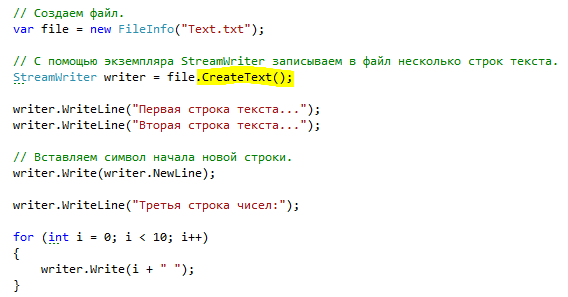


СИМВОЛЬНЫЕ ПОТОКИ

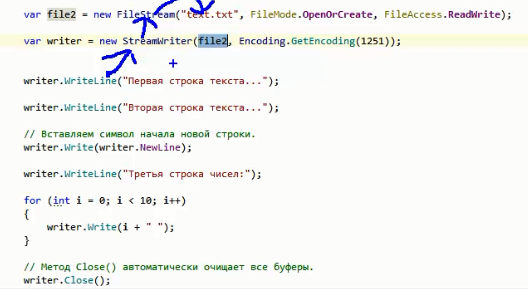


StreamWriter – символьный поток

file.CreateText(); - создает экземпляр класса StreamWriter



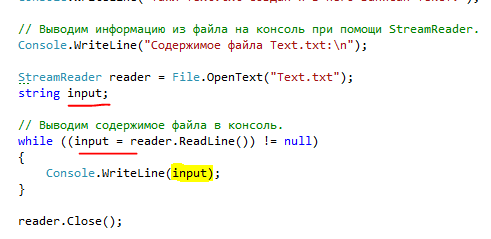
StreamWriter перекодирует в байтовый поток

var writer = new StreamWriter(file2, Encoding.GetEncoding(1251)); - первым аргументом передается байтовый поток, вторым аргументом кодировка (стандартная windows 1251)

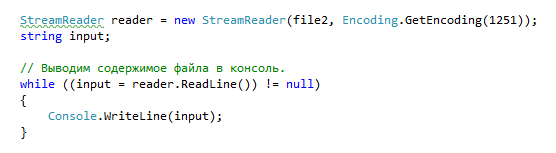
StreamReader

Вывод содержимого из файла

reader.ReadLine() – считывает построчно пока не вернуть null



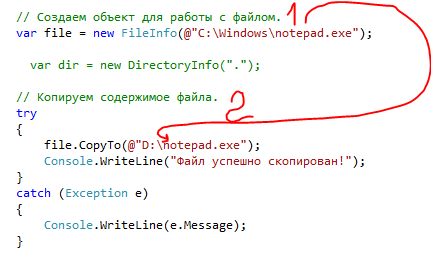
Если при записи была указана кодировка то ее следует указать при считывании



КОПИРОВАНИЕ ФАЙЛА FileInfo

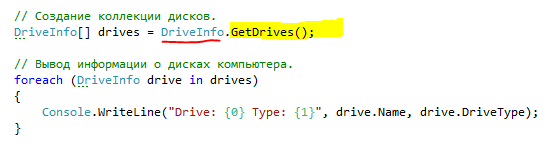
Копирование из 1 – го во 2 –й файл

filecreate.CopyTo(@"D:\new1.txt",true); -если файла нет то создаст новый, если есть будет исключение (передав вторым параметром true файл будет полностью перезаписан)

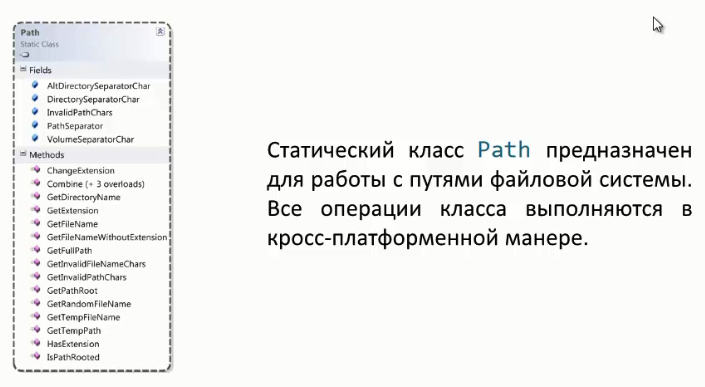


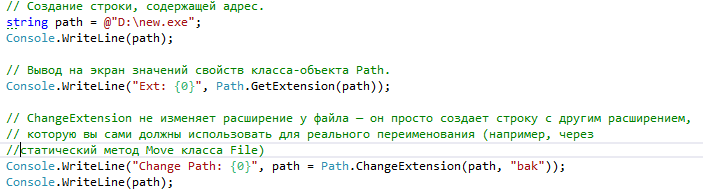
DriveInfo

DriveInfo.GetDrives(); - возвращает массив DriveInfo[] где описывается каждый диск

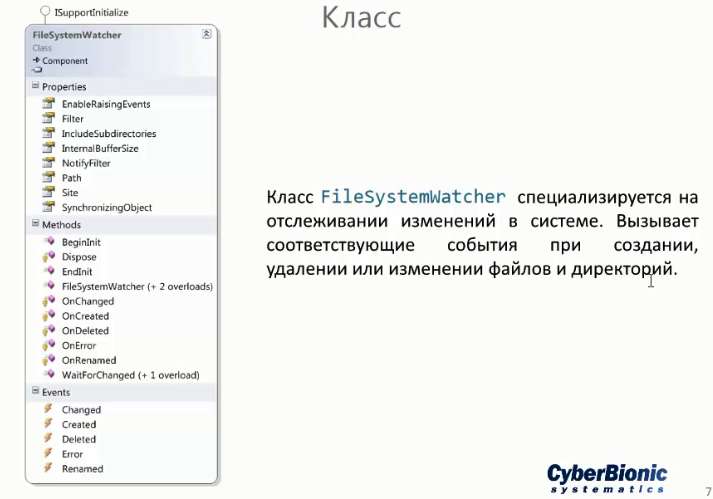


Path









FileSystemWatcher

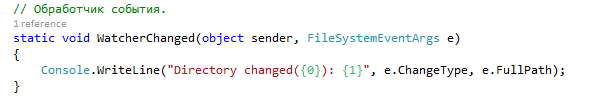
Позволяет отслеживать изменений в директории (при создании файла, папки…)

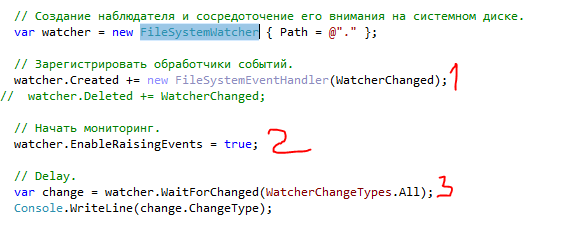
1. Метод который будет подписан на событие
2. Включить слежение за указанной директории
3. Останавливает поток до момента пока не будет создано что либо в директории

atcher.Created += - вызывается событие при создании файла

watcher.Deleted += - вызывается событие при удалении файла

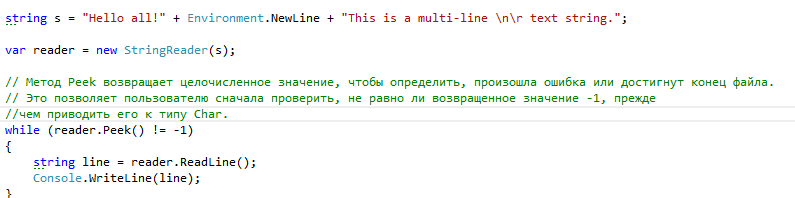
watcher.Renamed += - - вызывается событие при изменении имени файла



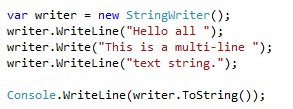


StringReader

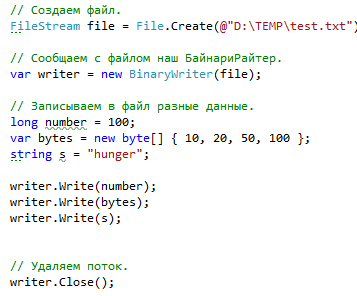
В случае если имеется большая строка и ее необходимо считать



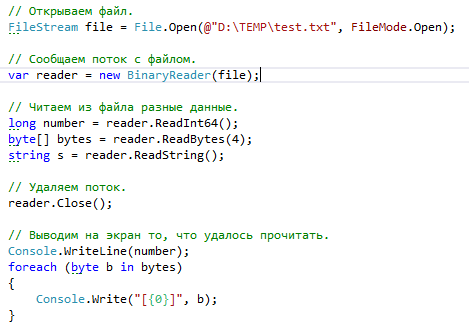
StringWriter - конкатенирует все строки в одну строку



BinaryWriter



BinaryReader

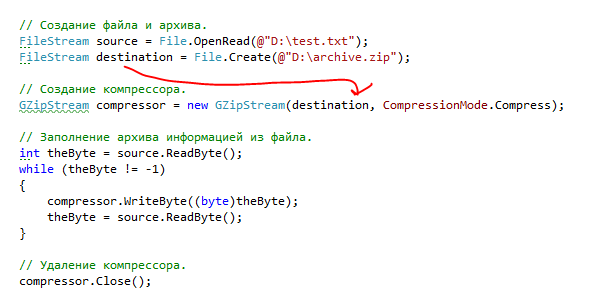


GZipStream то что и DeflateStream (архивирует и разархивирует данные)

using System.IO.Compression;

// Заполнение архива информацией из файла.

int theByte = source.ReadByte();



Распаковка

