1. Какие классы содержатся в System.IO?

**DriveInfo:** инкапсулирует инфу о диске

**GetDrives()** – получ массив DriveInfo  
**AvailableFreeSpace** – объем доступного места на диске [б]  
**DriveFormat** – имя файл. системы  
**DriveType** – тип диска  
**IsReady** – готов ли диск  
**Name** – имя диска  
**TotalFreeSpace** – общий объем своб. места на диске [б]  
**TotalSize** – размер диска [б]

2. Для чего исп. классы Directory и DirectoryInfo? В чем отличие?

DirectoryInfo содержит набор используемых для создания перемещения, удаления и перечисления каталогов и подкаталогов.

**Directory:**

**CreateDirectory(path)** – созд. каталог по указ. пути  
**Delete(path)** – удал. каталог  
**Exists(path)** – сущ. ли каталог (t/f)  
**GetDirectories(path)** – получ. список каталогов  
**GetFiles(path)** – получ. список файлов  
**Move(sourceDirName, destDirName)** – перемещ. каталог  
**GetParent(path)** – получ. родит. каталог

**DirectoryInfo:**

**Create()** – созд. каталог  
**CreateSubdirectory(path)** – созд. подкаталог  
**Delete()** – удал. каталог  
**Св-во Exists** – сущ. ли каталог (t/f)  
**GetDirectories()** – получ. список каталогов  
**GetFiles()** – получ. список файлов  
**Move(destDirName)** – перемещ. каталог  
**Св-во Parent** – получ. родит. каталог  
**Св-во Root** – получ. корневой каталог

3. Для чего исп. классы File и FileInfo? Какие методы они содержат.

Работа с файлами  
вып. операции при пом. стат. методов (File),  
при пом. экземплярных методов (FileInfo)

**File:**

**Copy()** – копирует файл в новое место  
**Create()** – созд. файл  
**Delete()** – удал. файл  
**Move** – перемещ. файл в новое место   
**Exists(file)** – опр., сущ. ли файл

**FileInfo:**

**CopyTo(path)** – копир. файл в новое место по указ. пути path  
**Create()** – созд. файл   
**Delete()** – удал. файл  
**MoveTo(destFileName)** – перемещ. файл в новое место  
**Свойство Directory** – получ. род. каталог в виде объекта DirectoryInfo  
**Свойство DirectoryName** – получ. полный путь к род. каталогу  
**Свойство Exists** – указ., сущ ли файл  
**Свойство Length** – получ. размер файла  
**Свойство Extension** – получ. расширение файла  
**Св-во Name** – получ. имя файла  
**Св-во FullName** – получ. полное имя файла

4. Для чего используются классы StreamReader, StreamWriter?

классы для ввода/вывода инфы в строковом представлении

**StreamReader:**

**Close** – закр. считываемый файл и освобожд. все ресурсы  
**Peek** – возвр. след. доступный символ или -1  
**Read** – счит. и возвр. след. символ в численном предст. Read(char[] array, int index, int count)  
**ReadLine** – счит. одну строку в файле   
**ReadToEnd** – счит. весь текст из файла

**StreamWriter:**

**Close** – закр. записываемый файл и освобожд. все ресурсы  
**Flush** – запис. в файл оставшиеся в буфере д-е и очищ. буфер  
**Write** – запис. в файл д-е простейших типов, как int, double, char, string…  
**WriteLine** – также запис. д-е, добав. в файл символ окончания строки

5. Для чего используются классы BinaryWriter и BinaryReader?

классы ввода/вывода примитивных видов в двоичном формате

**BinaryWriter:**

**Close()** – закр. поток и освобожд. ресурсы  
**Flush()**  – очищ. буфер, дописывая из него ост. д-е в файл  
**Seek()** – устан. позицию в потоке  
**Write()** – запис. д-е в поток

**BinaryReader:**

**Close()** – закр. поток и освобожд. ресурсы  
**ReadBoolean()** – счит. значение Boolean и перемещ. указ. на 1 байт  
**ReadByte()** – счит. 1 байт, перем. указ. на 1 байт  
**ReadChar()** – байт ск. в тек. кодировке  
**ReadDecimal()** – 16  
**ReadDouble()** – 8  
**ReadInt16()** – 2  
**ReadInt32()** – 4  
**ReadInt64()** – 8  
**ReadSingle()** – 4  
**ReadString()** –

6. Как можно сжимать и восстанавливать файлы?

DeflateStream и GZipStream

7. Расскажите алгоритм сжатия GZip

Этот класс представляет формат данных Gzip, использующий промышленный алгоритм для сжатия и распаковки файлов без потерь. Этот формат включает значение циклической проверки избыточности для обнаружения повреждений данных.

Функция сжатия в [DeflateStream](https://docs.microsoft.com/ru-ru/dotnet/api/system.io.compression.deflatestream?view=netframework-4.8) и [GZipStream](https://docs.microsoft.com/ru-ru/dotnet/api/system.io.compression.gzipstream?view=netframework-4.8) предоставляется в виде потока. Данные считываются на основе байтов, поэтому невозможно выполнить несколько проходов, чтобы определить лучший метод сжатия целых файлов или больших блоков данных. Классы [DeflateStream](https://docs.microsoft.com/ru-ru/dotnet/api/system.io.compression.deflatestream?view=netframework-4.8) и [GZipStream](https://docs.microsoft.com/ru-ru/dotnet/api/system.io.compression.gzipstream?view=netframework-4.8) лучше использовать в несжатых источниках данных. Если исходные данные уже сжаты, использование этих классов может фактически

8. Для чего нужен класс Path

► предназначен для работы с именами файлов и путями в файловой системе Назначение ► выделить имя файла из полного пути ► скомбинировать для получения пути имя файла и имя каталога ► сгенерировать имя для временного файла или каталога

9. Что такое текстовый и бинарный файл? Как можно выполнить чтение и запись в эти файлы.

Текстовые -только текст

Бинарные-могут содержать любые символы кодовой таблицы

Сама процедура **чтения** или **записи** **бинарного** **файла** состоит из тех же шагов, что для **текстового**: **файл** связывается с файловой переменной оператором assign, затем открывается в режиме reset или rewrite, **чтение** **и** **запись** выполняются операторами read и write соответственно, по завершении работы с **файлом** его следует закрыть оператором close.

10. Что такое произвольный доступ к файлу? Приведите пример

**Файлы с произвольным доступом**[[1]](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB%D1%8B_%D1%81_%D0%BF%D1%80%D0%BE%D0%B8%D0%B7%D0%B2%D0%BE%D0%BB%D1%8C%D0%BD%D1%8B%D0%BC_%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BE%D0%BC#cite_note-1) — [файлы](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB), хранящие информацию в структурированном (для [поиска](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%B8%D1%81%D0%BA) и обращения) виде. Поиск в таких файлах осуществляется в области адресов (ключей) и завершается обращением непосредственно к искомому участку.

Seek()  
SeekOrigin.Begin – начало файла  
SeekOrigin.End  
SeekOrigin.Current

11. Как применяется конструкция using (Не директива) при работе с файловыми потоками? Для чего ее используют

using (получение-ресурса)  
 вложенный оператор

получение-ресурса:  
\* об. и иниц. лок. пер-ной, ее тип реализ. IDisposable, только для чт  
\* вырж, знач. которого имеет тип, реализ. IDisposable