1. Какие классы содержаться с System.IO?

Содержит типы, позволяющие осуществлять чтение и запись в файлы и потоки данных, а также типы для базовой поддержки файлов и папок.

|  |  |
| --- | --- |
| [BinaryReader](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.binaryreader?view=net-7.0) | Считывает примитивные типы данных как двоичные значения в заданной кодировке. |
| [BinaryWriter](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.binarywriter?view=net-7.0) | Записывает примитивные типы в двоичный поток и поддерживает запись строк в заданной кодировке. |
| [BufferedStream](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.bufferedstream?view=net-7.0) | Добавляет уровень буферизации в операциях чтения и записи в другие потоки. Этот класс не наследуется. |
| [Directory](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.directory?view=net-7.0) | Предоставляет статические методы для создания, перемещения и перечисления в каталогах и вложенных каталогах. Этот класс не наследуется. |
| [DirectoryInfo](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.directoryinfo?view=net-7.0) | Предоставляет методы экземпляра класса для создания, перемещения и перечисления в каталогах и подкаталогах. Этот класс не наследуется. |
| [DirectoryNotFoundException](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.directorynotfoundexception?view=net-7.0) | Исключение, которое создается, когда не удается найти часть файла или каталога. |
| [DriveInfo](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.driveinfo?view=net-7.0) | Предоставляет доступ к сведениям на диске. |
| [DriveNotFoundException](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.drivenotfoundexception?view=net-7.0) | Исключение вызывается при попытке доступа к недоступному диску или данным совместного использования. |
| [EndOfStreamException](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.endofstreamexception?view=net-7.0) | Исключение, которое выдается при попытке чтения за концом потока. |
| [EnumerationOptions](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.enumerationoptions?view=net-7.0) | Предоставляет параметры перечисления файлов и каталогов. |
| [ErrorEventArgs](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.erroreventargs?view=net-7.0) | Предоставляет данные для события [Error](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.filesystemwatcher.error?view=net-7.0). |
| [File](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.file?view=net-7.0) | Предоставляет статические методы для создания, копирования, удаления, перемещения и открытия одного файла, а также помогает при создании объектов [FileStream](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.filestream?view=net-7.0). |
| [FileInfo](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.fileinfo?view=net-7.0) | Предоставляет свойства и методы экземпляра для создания, копирования, удаления, перемещения и открытия файлов, а также позволяет создавать объекты [FileStream](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.filestream?view=net-7.0). Этот класс не наследуется. |
| [FileLoadException](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.fileloadexception?view=net-7.0) | Исключение, которое создается в случае, когда управляемая сборка найдена, но не может быть загружена. |
| [FileNotFoundException](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.filenotfoundexception?view=net-7.0) | Исключение, которое выдается при попытке получить доступ к файлу или каталогу, которых нет на диске. |
| [FileStream](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.filestream?view=net-7.0) | Предоставляет [Stream](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.stream?view=net-7.0) в файле, поддерживая синхронные и асинхронные операции чтения и записи. |
| [FileStreamOptions](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.filestreamoptions?view=net-7.0) | Определяет различные параметры конфигурации для [FileStream](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.filestream?view=net-7.0). |
| [FileSystemAclExtensions](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.filesystemaclextensions?view=net-7.0) | Предоставляет предназначенные для платформы Windows статические методы расширения, которые служат для работы с атрибутами безопасности списков управления доступом (ACL) для файлов и каталогов. |
| [FileSystemEventArgs](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.filesystemeventargs?view=net-7.0) | Предоставляет данные для событий каталога: [Changed](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.filesystemwatcher.changed?view=net-7.0), [Created](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.filesystemwatcher.created?view=net-7.0), [Deleted](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.filesystemwatcher.deleted?view=net-7.0). |
| [FileSystemInfo](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.filesysteminfo?view=net-7.0) | Предоставляет базовый класс для объектов [FileInfo](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.fileinfo?view=net-7.0) и [DirectoryInfo](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.directoryinfo?view=net-7.0). |
| [FileSystemWatcher](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.filesystemwatcher?view=net-7.0) | Ожидает уведомления файловой системы об изменениях и инициирует события при изменениях каталога или файла в каталоге. |
| [InternalBufferOverflowException](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.internalbufferoverflowexception?view=net-7.0) | Исключение, которое создается при переполнении внутреннего буфера. |
| [InvalidDataException](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.invaliddataexception?view=net-7.0) | Исключение, которое выдается, если поток данных имеет недопустимый формат. |
| [IOException](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.ioexception?view=net-7.0) | Исключение, которое выдается при возникновении ошибки ввода-вывода. |
| [MemoryStream](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.memorystream?view=net-7.0) | Создает поток, резервным хранилищем которого является память. |
| [Path](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.path?view=net-7.0) | Выполняет операции для экземпляров класса [String](https://learn.microsoft.com/ru-ru/dotnet/api/system.string?view=net-7.0), содержащих сведения о пути к файлу или каталогу. Эти операции выполняются межплатформенным способом. |
| [PathTooLongException](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.pathtoolongexception?view=net-7.0) | Исключение, которое создается, когда путь или полное имя файла длиннее, чем максимальная длина, определенная системой. |
| [RandomAccess](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.randomaccess?view=net-7.0) | Предоставляет API на основе смещения для чтения и записи файлов потокобезопасным способом. |
| [RenamedEventArgs](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.renamedeventargs?view=net-7.0) | Предоставляет данные для события [Renamed](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.filesystemwatcher.renamed?view=net-7.0). |
| [Stream](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.stream?view=net-7.0) | Предоставляет универсальное представление последовательности байтов. Этот класс является абстрактным. |
| [StreamReader](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.streamreader?view=net-7.0) | Реализует объект [TextReader](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.textreader?view=net-7.0), который считывает символы из потока байтов в определенной кодировке. |
| [StreamWriter](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.streamwriter?view=net-7.0) | Реализует [TextWriter](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.textwriter?view=net-7.0) для записи символов в поток в определенной кодировке. |
| [StringReader](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.stringreader?view=net-7.0) | Реализует класс [TextReader](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.textreader?view=net-7.0), осуществляющий чтение из строки. |
| [StringWriter](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.stringwriter?view=net-7.0) | Реализует объект [TextWriter](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.textwriter?view=net-7.0) для записи сведений в строку. Сведения хранятся в базовом [StringBuilder](https://learn.microsoft.com/ru-ru/dotnet/api/system.text.stringbuilder?view=net-7.0). |
| [TextReader](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.textreader?view=net-7.0) | Представляет средство чтения, позволяющее считывать последовательные наборы символов. |
| [TextWriter](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.textwriter?view=net-7.0) | Представляет модуль записи, который может записывать последовательные наборы символов. Это абстрактный класс. |
| [UnmanagedMemoryAccessor](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.unmanagedmemoryaccessor?view=net-7.0) | Предоставляет произвольный доступ к неуправляемым блокам памяти из управляемого кода. |
| [UnmanagedMemoryStream](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.unmanagedmemorystream?view=net-7.0) | Предоставляет доступ к неуправляемым блокам памяти из управляемого кода. |

2. Для чего используются классы Directory и DirectoryInfo? В чем

отличие?

|  |  |
| --- | --- |
| [Directory](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.directory?view=net-7.0) | Предоставляет статические методы для создания, перемещения и перечисления в каталогах и вложенных каталогах. Этот класс не наследуется. |
| [DirectoryInfo](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.directoryinfo?view=net-7.0) | Предоставляет методы экземпляра класса для создания, перемещения и перечисления в каталогах и подкаталогах. Этот класс не наследуется. |

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

содержат.

|  |  |
| --- | --- |
| [File](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.file?view=net-7.0) | Предоставляет статические методы для создания, копирования, удаления, перемещения и открытия одного файла, а также помогает при создании объектов [FileStream](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.filestream?view=net-7.0). |
| [FileInfo](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.fileinfo?view=net-7.0) | Предоставляет свойства и методы экземпляра для создания, копирования, удаления, перемещения и открытия файлов, а также позволяет создавать объекты [FileStream](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.filestream?view=net-7.0). Этот класс не наследуется. |

1. Для чего используются классы StreamReader и StreamWriter?

|  |  |
| --- | --- |
| [StreamReader](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.streamreader?view=net-7.0) | Реализует объект [TextReader](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.textreader?view=net-7.0), который считывает символы из потока байтов в определенной кодировке. |
| [StreamWriter](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.streamwriter?view=net-7.0) | Реализует [TextWriter](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.textwriter?view=net-7.0) для записи символов в поток в определенной кодировке. |

1. Для чего используются классы BinaryWriter и BinaryReader?

|  |  |
| --- | --- |
| [BinaryReader](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.binaryreader?view=net-7.0) | Считывает примитивные типы данных как двоичные значения в заданной кодировке. |
| [BinaryWriter](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.binarywriter?view=net-7.0) | Записывает примитивные типы в двоичный поток и поддерживает запись строк в заданной кодировке. |

1. Как можно сжимать и восстанавливать файлы?

string sourceFolder = "D://test/"; // исходная папка

string zipFile = "D://test.zip"; // сжатый файл

string targetFolder = "D://newtest"; // папка, куда распаковывается файл

ZipFile.CreateFromDirectory(sourceFolder, zipFile);

Console.WriteLine($"Папка {sourceFolder} архивирована в файл {zipFile}");

ZipFile.ExtractToDirectory(zipFile, targetFolder);

GZIP обеспечивает сжатие без потерь, иными словами, исходные данные можно полностью восстановить при распаковке. Он основан на алгоритме DEFLATE, который использует комбинацию алгоритма LZ77 и алгоритма Хаффмана.

Алгоритм LZ77 заменяет повторные вхождения данных на «ссылки».

1. Расскажите алгоритм сжатия GZip.

https://habr.com/ru/post/221849/

async Task CompressAsync(string sourceFile, string compressedFile)

{

// поток для чтения исходного файла

using FileStream sourceStream = new FileStream(sourceFile, FileMode.OpenOrCreate);

// поток для записи сжатого файла

using FileStream targetStream = File.Create(compressedFile);

// поток архивации

using GZipStream compressionStream = new GZipStream(targetStream, CompressionMode.Compress);

await sourceStream.CopyToAsync(compressionStream); // копируем байты из одного потока в другой

Console.WriteLine($"Сжатие файла {sourceFile} завершено.");

Console.WriteLine($"Исходный размер: {sourceStream.Length} сжатый размер: {targetStream.Length}");

}

async Task DecompressAsync(string compressedFile, string targetFile)

{

// поток для чтения из сжатого файла

using FileStream sourceStream = new FileStream(compressedFile, FileMode.OpenOrCreate);

// поток для записи восстановленного файла

using FileStream targetStream = File.Create(targetFile);

// поток разархивации

using GZipStream decompressionStream = new GZipStream(sourceStream, CompressionMode.Decompress);

await decompressionStream.CopyToAsync(targetStream);

Console.WriteLine($"Восстановлен файл: {targetFile}");

}

1. Для чего служит класс Path?

|  |  |
| --- | --- |
| [Path](https://learn.microsoft.com/ru-ru/dotnet/api/system.io.path?view=net-7.0) | Выполняет операции для экземпляров класса [String](https://learn.microsoft.com/ru-ru/dotnet/api/system.string?view=net-7.0), содержащих сведения о пути к файлу или каталогу. Эти операции выполняются межплатформенным способом. |

9. Что такое тестовый и бинарный файл. Как можно выполнить чтение

и запись в эти файлы.

**BinaryWriter** и **BinaryReader**.

using (BinaryWriter writer = new BinaryWriter(File.Open(path, FileMode.OpenOrCreate)))

{

    // записываем в файл строку

    writer.Write("Tom");

    // записываем в файл число int

    writer.Write(37);

}

using (BinaryReader reader = new BinaryReader(File.Open("person.dat", FileMode.Open)))

{

    // считываем из файла строку

    string name = reader.ReadString();

    // считываем из файла число

    int age = reader.ReadInt32();

}

StreamReader и StreamWriter

using (StreamWriter writer = new StreamWriter(path, false))

{

    await writer.WriteLineAsync(text);

}

using (StreamReader reader = new StreamReader(path))

{

    string text = await reader.ReadToEndAsync();

    Console.WriteLine(text);

}

1. Что такое произвольный доступ к файлу? Приведите пример.

11.Как применяется конструкция using (не директива) при работе с

файловыми потоками? Для чего ее используют.