**Вопросы к лабораторной работе №12:**

1. Какие классы содержаться в System.IO?

* **BinaryReader, BinaryWriter**

Эти классы позволяют сохранять и извлекать элементарные типы данных (целочисленные, булевские, строковые и т.п.) в двоичном виде.

* **BufferedStream**

Этот класс предоставляет временное хранилище для потока байтов, которые могут затем быть перенесены в постоянные хранилища.

* **Directory, DirectoryInfo**

Эти классы используются для манипуляций структурой каталогов машины. Тип Directory представляет функциональность, используя статические члены. Тип DirectoryInfo обеспечивает аналогичную функциональность через действительную объектную ссылку.

* **DriveInfo**

Этот класс предоставляет детальную информацию относительно дисковых устройств, используемых данной машиной.

* **File, FileInfo**

Эти классы служат для манипуляций множеством файлов данной машины. Тип File представляет функциональность через статические члены. Тип FileInfo обеспечивает аналогичную функциональность через действительную объектную ссылку.

* **FileStream**

Этот класс обеспечивает произвольный доступ к файлу (т.е. возможности поиска) с данными, представленными в виде потока байт.

* **FileSystemWatcher**

Этот класс позволяет отслеживать модификации внешних файлов в определенном каталоге.

* **MemoryStream**

Этот класс обеспечивает произвольный доступ к данным, хранящимся в памяти, а не в физическом файле.

* **Path**

Этот класс выполняет операции над типами System.String, содержащими информацию о пути к файлу или каталогу в независимой от платформы манере.

* **StreamWriter, StreamReader**

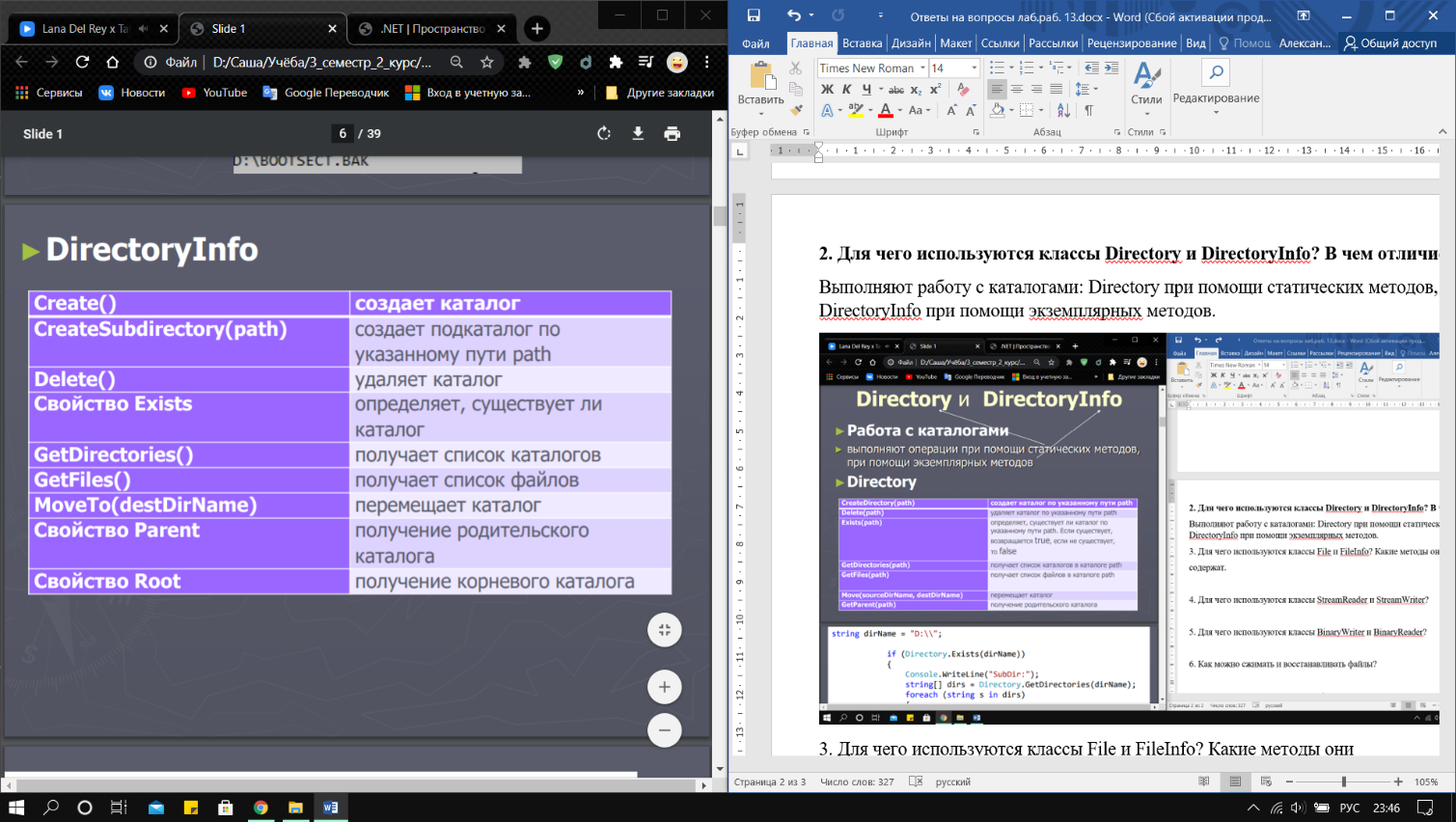
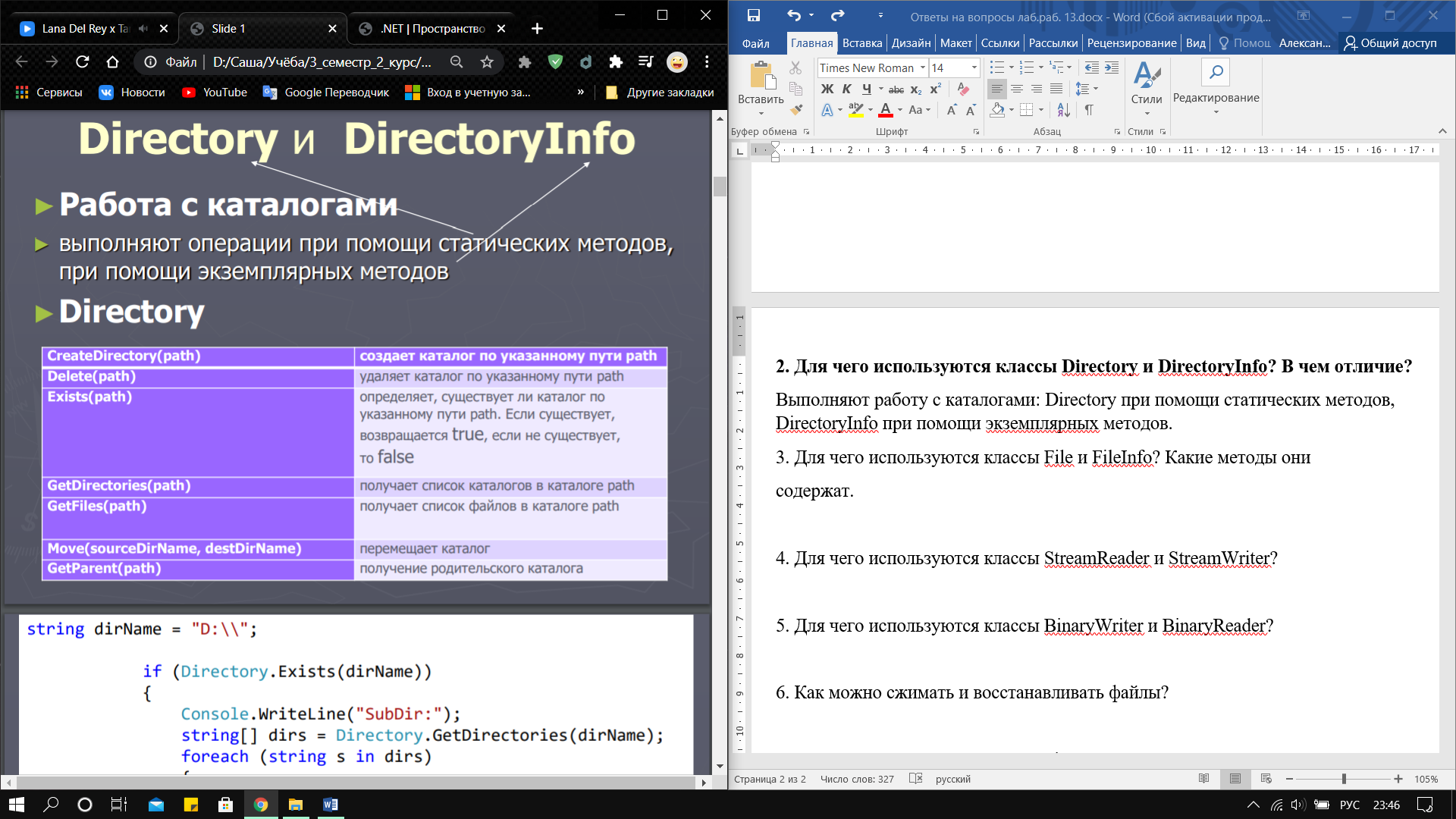
Эти классы используются для хранения (и извлечения) текстовой информации из файла. Эти классы не поддерживают произвольного доступа к файлу.

* **StringWriter, StringReader**

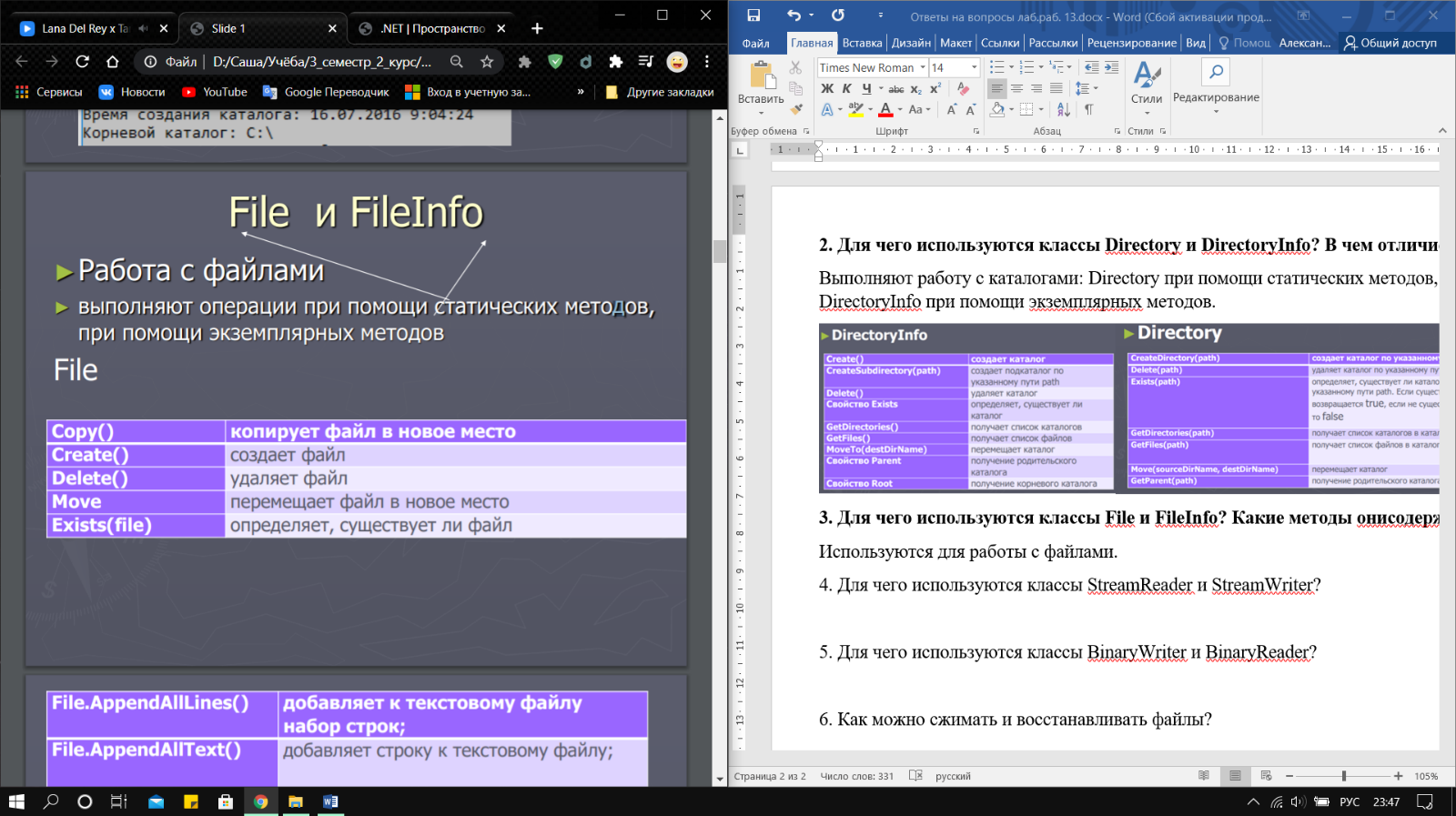
Подобно классам StreamWriter/StreamReader, эти классы также работают с текстовой информацией. Однако лежащим в основе хранилищем является строковый буфер, а не физический файл.

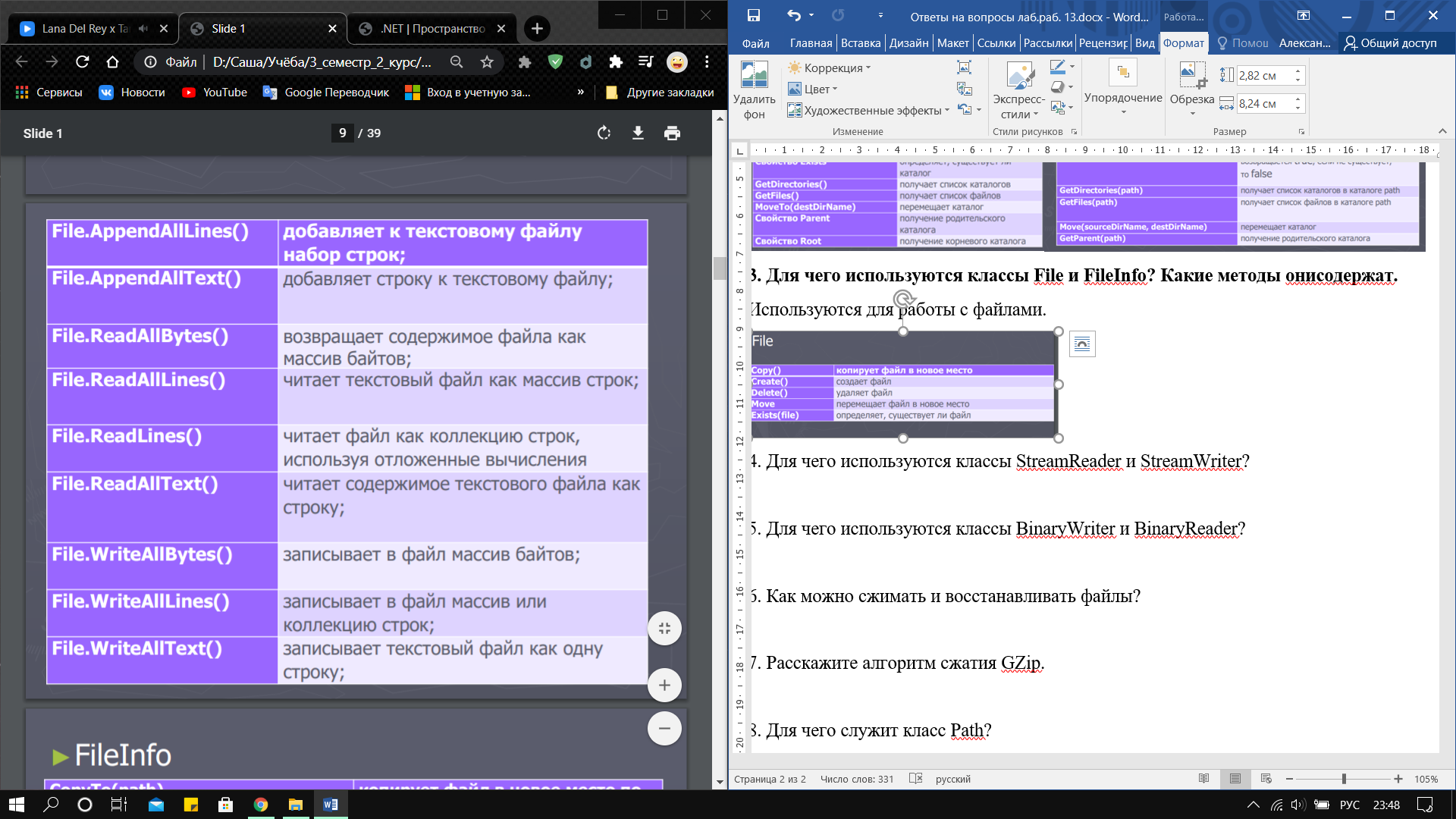
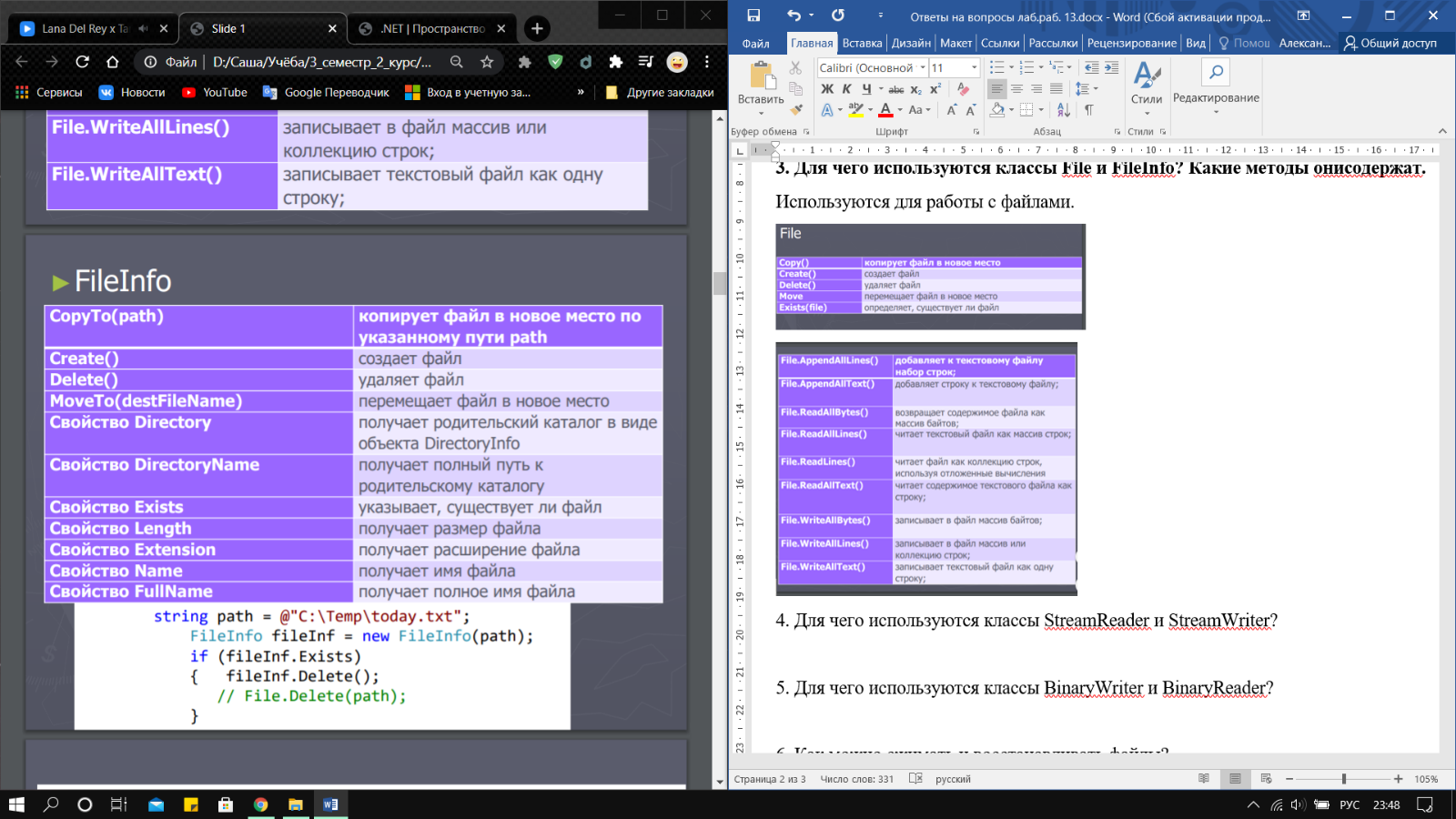
1. Для чего используются классы Directory и DirectoryInfo? В чем отличие?

Выполняют работу с каталогами: **Directory** при помощи статических методов, **DirectoryInfo** при помощи экземплярных методов.



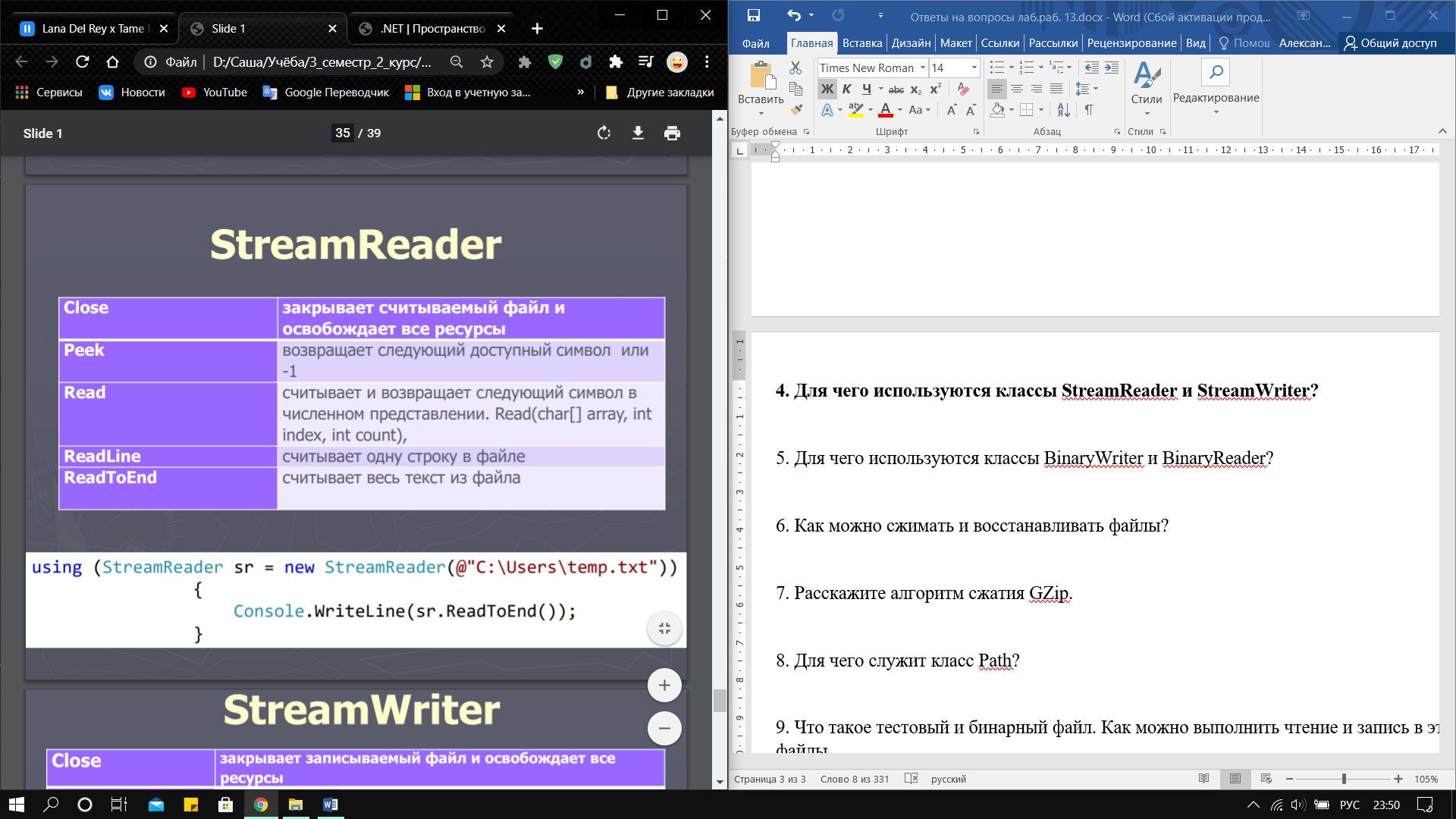
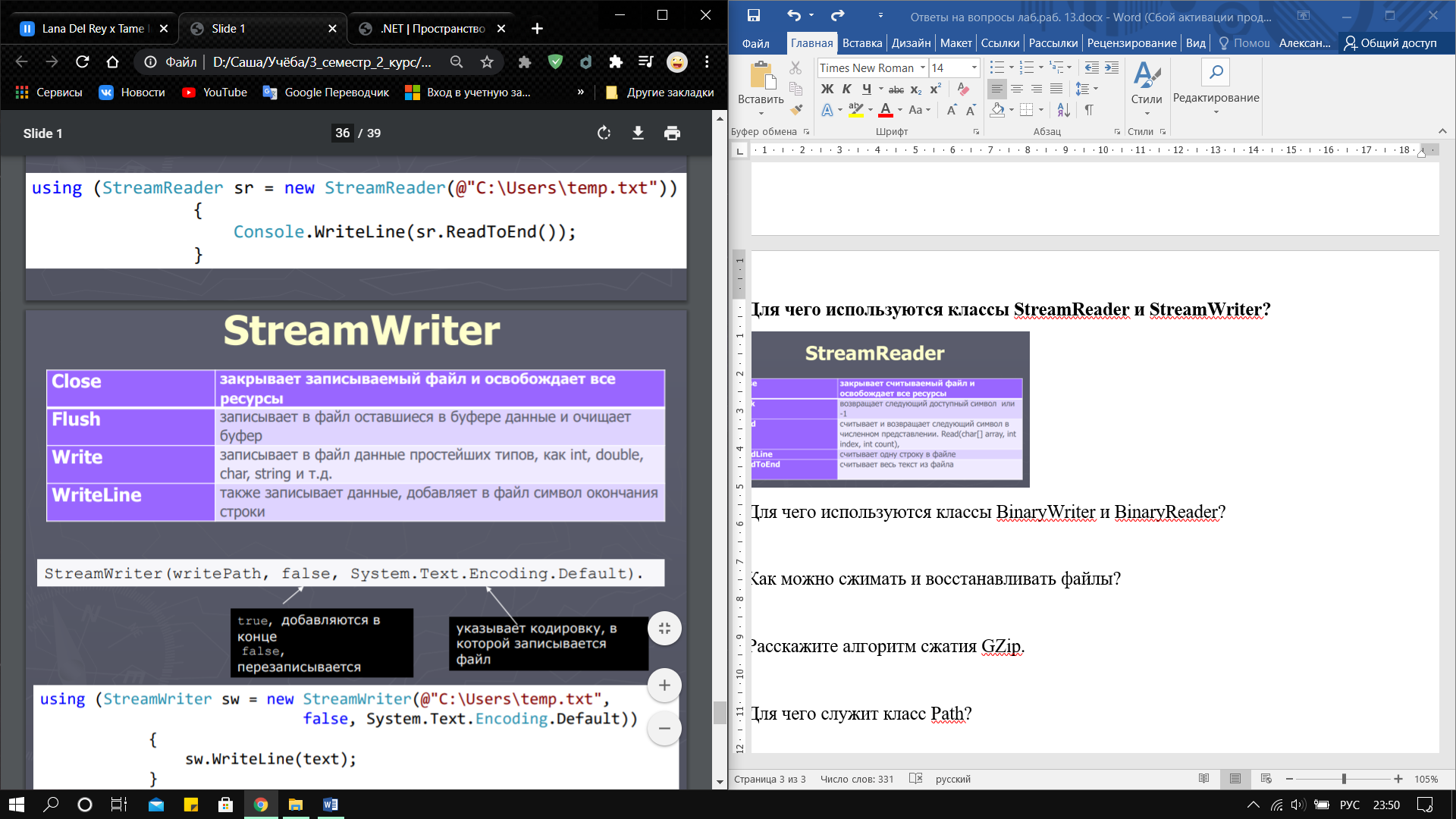
1. Для чего используются классы File и FileInfo? Какие методы они содержат.

Используются для работы с файлами.



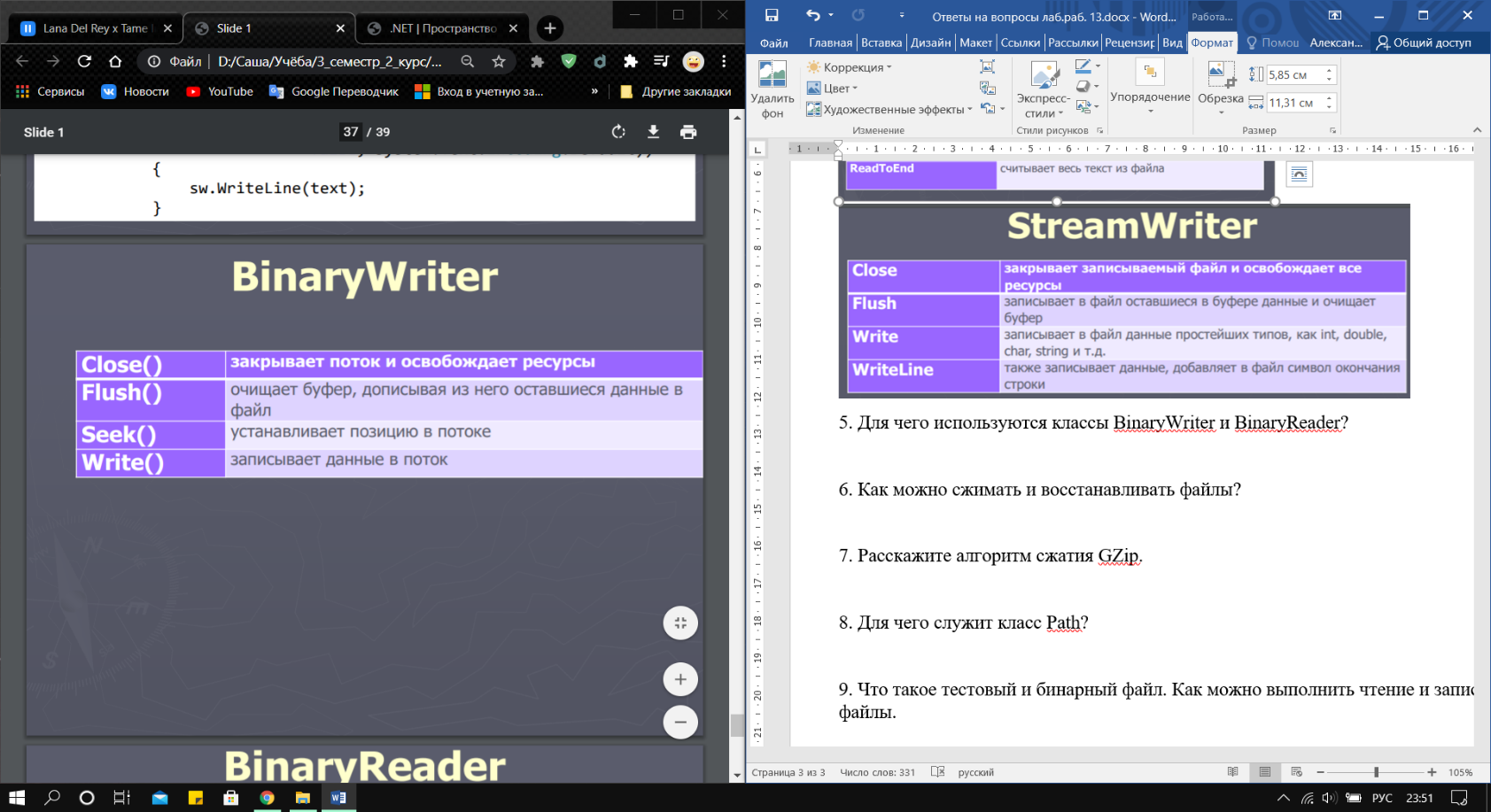
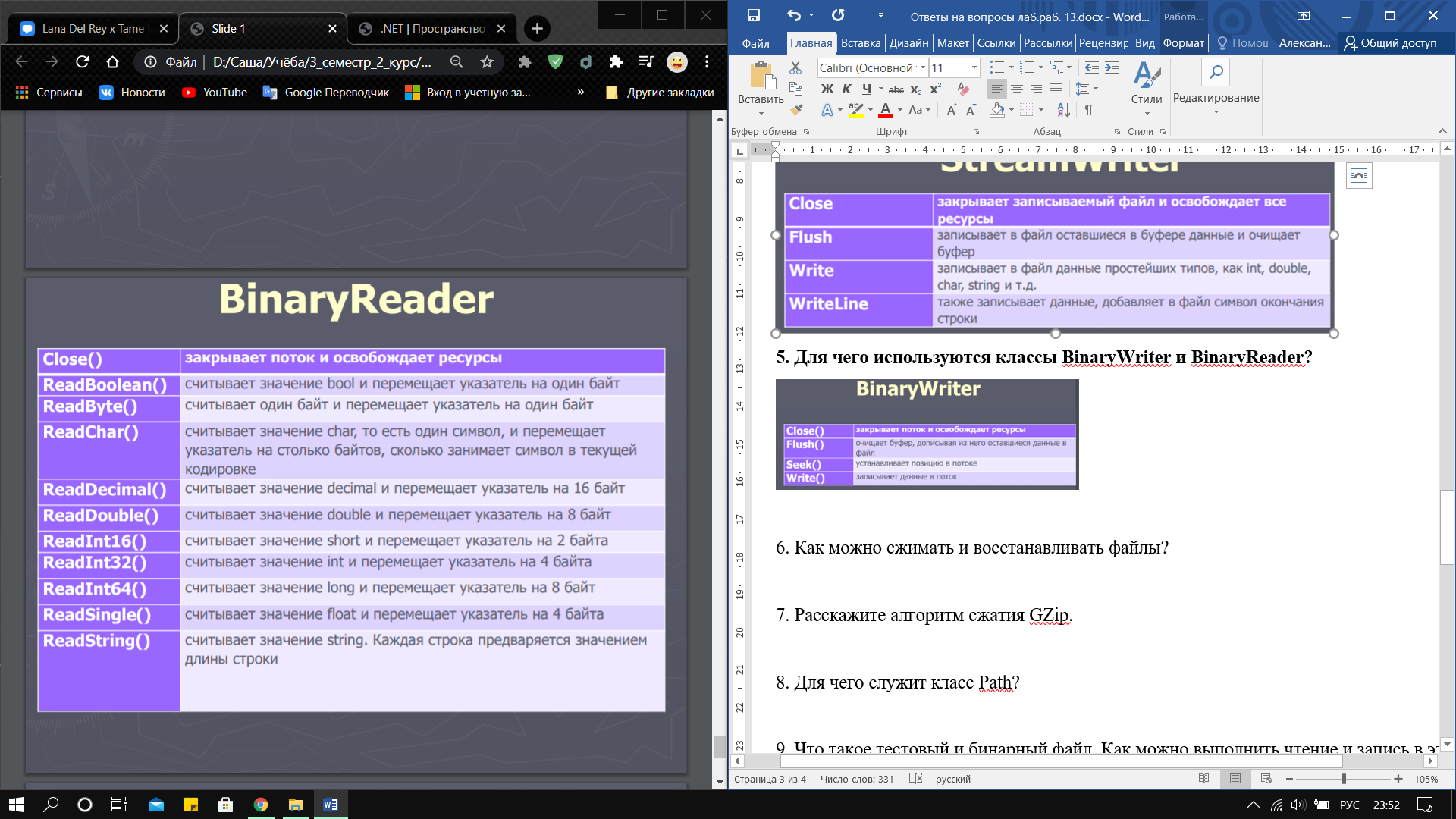
1. Для чего используются классы StreamReader и StreamWriter?

**StreamReader** и **StreamWriter** – классы для ввода и вывода информации в строковом представлении.



1. Для чего используются классы BinaryWriter и BinaryReader?

**BinaryReader** и **BinaryWriter** – классы для ввода и вывода примитивных типов в двоичном формате.



1. Как можно сжимать и восстанавливать файлы?

**ZipArchive и ZipFile –** архивация

**ZipFile.ExtractToDirectory(path2, path3)** – разархивация

**DeflateStream** и **GZipStream** – классы для потоков со сжатием данных (System.IO.Compression).

GZipStream(Stream stream, CompressionMode mode): mode указывает, будут ли данные сжиматься или, наоборот, восстанавливаться и может принимать два значения:

* CompressionMode.Compress: данные сжимаются
* CompressionMode.Decompress: данные восстанавливаться

1. Расскажите алгоритм сжатия GZip.

Компрессия выполняется в два этапа:

* Замена повторного вхождения строки ссылкой на одну из предыдущих позиций вхождения. (алгоритм LZ77);
* замена символов новыми символами, основываясь на частоте их использования (алгоритм Хаффмана).

1. Для чего служит класс Path?

Класс **Path** представляет строку, в которой заложены:

* путь к файлу;
* файловое имя;
* расширение файла.

Каждый из этих элементов формируется в процессе выполнения программы на основе правил используемых операционной системой.

1. Что такое тестовый и бинарный файл. Как можно выполнить чтение и запись в эти файлы.

* **Двоичный (бинарный) файл** — в широком смысле: последовательность произвольных байтов. Название связано с тем, что байты состоят из бит, то есть двоичных (англ. binary) цифр.
* **Текстовый файл** — это файл, в котором данные хранятся в виде символов ASCII, и обычно используется для хранения потока символов. StreamReader, StreamWriter.
* **Двоичный** **файл** — это файл, в котором данные хранятся в файле так же, как они хранятся в основной памяти для обработки. Он хранится в двоичном формате вместо символов ASCII.

В узком смысле слова двоичные файлы противопоставляются текстовым файлам. При этом с точки зрения технической реализации на уровне аппаратуры, текстовые файлы являются частным случаем двоичных файлов, и, таким образом, в широком значении слова под определение «двоичный файл» подходит любой файл.

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

Взять из файла нужную порцию информации или наоборот записать в определенном месте файла определенный набор байтов.

Произвольный доступ к файлам:

* С помощью метода Seek() мы можем управлять положением курсора потока, начиная с которого производится считывание или запись в файл. Этот метод принимает два параметра: offset (смещение) и позиция в файле. Позиция в файле описывается тремя значениями:

Seek(-5, SeekOrigin.End); // минус 5 символов с конца потока

* SeekOrigin.Begin: начало файла
* SeekOrigin.End: конец файла
* SeekOrigin.Current: текущая позиция в файле

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

Конструкция using оформляет блок кода и создает объект некоторого типа

