**Группа** ИС 3-1

**ФИО:** Мозыркий А.Д.

**Курс:** C# Professional

**Урок 3:** Программирование ввода-вывода

**Теория**

* Для перечисления объектов файловой системы и получения подробной информации об их свойствах можно использовать классы FileInfo, DirectoryInfo и DriveInfo.
* Класс Path позволяет получать подробную информацию о путях файловой системы, его следует использовать вместо ручного разбора путей.
* Для отслеживания изменений файловой системы, таких как добавление, удаление и переименование файлов и папок, можно использовать класс FileSystemWatсher.
* Класс File позволяет открывать, создавать, читать и записывать файлы целиком либо по частям.
* Класс FileStream представляет файл и позволяет выполнять чтение и запись.
* Чтобы упростить чтение-запись строк в потоки, используются классы StreamReader и StreamWriter.
* Класс MemoryStream – специализированный поток, поддерживающий создание памяти буфера чтения-записи и запись данных буферизированного потока в другие потоки.
* Классы потоков сжатия (GZipStream и DeflateStream) поддерживают сжатие декомпрессию данных объемом до 4-х Гб
* Классы потоков сжатия служат оболочками потоков, хранящих сжатые данные.
* Изолированное хранилище – это защищенная область для хранения данных, специфичных для сборки, пользователя или приложения. Для работы с изолированным хранилищем не требуется высоких привилегий, поэтому приложения смогут хранить в нем свои данные, даже не обладая разрешениями на доступ к системе пользователя.
* Для хранения данных сборок и пользователей в защищенных областях используется класс IsolatedStorageFile.
* Класс IsolatedStorageFileStream позволяет обмениваться данными с безопасными хранилищами.
* Поскольку класс IsolatedStorageFileStream – потомок FileStream, с созданными им файлами можно работать, как с любыми другими файлами файловой системы.
* Класс IsolatedStorageFilePermission гарантирует наличие у кода разрешений, необходимых для взаимодействия с изолированным хранилищем

**Практика**

**Задание 1**

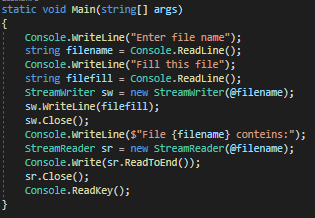


Рис.1.1 Реализация кода программы

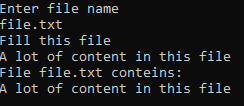


Рис.1.2 Результат работы программы

Результат работы программы соответствует ожидаемому.

**Задание 2**



Рис.2.1 Реализация кода программы

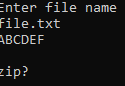




Рис.2.2 Результат работы программы

Результат работы программы соответствует ожидаемому.

**Задание 3**



Рис.3.1 Реализация кода программы





Рис.3.2 Результат работы программы

Результат работы программы соответствует ожидаемому.