|  |  |  |
| --- | --- | --- |
| Дата | Содержание выполняемой работы | Подпись  руководителя |
| 05.04.2020 | **Практическая работа №13**  Тема: Работа с файлами. Потоки символов. Работа с каталогами. Сохранение объектов.  Наименование работы: Потоки байтов. Потоки символов. Консольный ввод-вывод. Работа с каталогами и  файлами.  Цель: Сформировать навыки создания программ по работе с потоками символов. Сформировать навыки создания программ по чтению/записи данных с файла.  Задание 1.  Изучил предложенную теорию. Кратко законспектировал.  Пространство имен System.IO.  Пространство имен System.IO в .NET — это область библиотек базовых классов, посвященная службам файлового ввода-вывода, а также ввода-вывода из памяти.  Классы File и FileInfo.  Класс FileInfo позволяет получать подробности относительно существующих файлов на жестком диске и предназначен для создания, копирования, перемещения и удаления файлов.  Тип File предоставляет функциональность, почти идентичную типу FileInfo, с помощью нескольких статических методов.  Классы Directory и DirectoryInfo.  DirectoryInfo содержит набор членов, используемых для создания, перемещения, удаления и перечисления каталогов и подкаталогов.  По большей части статические члены Directory повторяют функциональность, предоставленную членами уровня экземпляра, которые определены в DirectoryInfo. Однако DriveInfo предоставляет множество дополнительных деталей.  Класс Path.  Для класса Path создавать экземпляры не нужно. Он предоставляет статические методы, которые упрощают выполнение операций с путевыми именами.  Класс Stream.  Основным для потоков является класс System.IO.Stream. Он представляет байтовый поток и является базовым для всех остальных классов потоков.  Класс FileStream.  Класс FileStream предоставляет реализацию абстрактного члена Stream в манере, подходящей для потоковой работы с файлами.  Классы StreamWriter и StreamReader.  Классы StreamWriter и StreamReader удобны во всех случаях, когда нужно читать или записывать символьные данные. Оба типа работают по умолчанию с символами Unicode; однако это можно изменить предоставлением правильно сконфигурированной ссылки на объект System.Text.Encoding.  Классы BinaryWriter и BinaryReader.  Для чтения и записи двоичных значений встроенных в C# типов данных служат классы потоков BinaryReader и BinaryWriter.  BinaryWriter(Stream output) где output обозначает поток, в который выводятся записываемые данные. Для записи в выходной файл в качестве параметра output может быть указан объект, создаваемый средствами класса FileStream.  BinaryReader(Stream input) где input обозначает поток, из которого вводятся считываемые данные. Для чтения из входного файла в качестве параметра input может быть указан объект, создаваемый средствами класса FileStream.  Задание 2.  Создал программы по образцу. Листинг кода переписал в Приложение 6.  Задание 2.1  Создание приложения по сохранению объектов класса. Листинг кода переписал в Приложение 12. Результат выполнения программы представлен на рисунке 27    Рисунок 27 – результат выполнения программы  Задание 2.2  Создание приложения по чтению/записи текстовых файлов. Листинг кода переписал в Приложение 12. Результат выполнения программы представлен на рисунке 28.    Рисунок 28 – результат выполнения программы    Рисунок 29 – результат выполнения программы  **Контрольные вопросы:**   1. Укажите, какие классы работают с потоками байтов. Перечислите их методы и свойства.   Ответ: Классы: Stream, BinaryWriter(Stream output), BinaryReader(Stream input).  Методы и свойства: CanRead, CanWrite, CanSeek, Close(), Fluch(), Length, Position, Read(), ReadByte(), Seek(), SetLength(), Write(), WriteByte().   1. Укажите, какие классы работают с потоками символов. Перечислите их методы и свойства.   Ответ: Классы: StreamWriter, StreamReader.  Методы и свойства: Close(), Flush(), NewLine, Write(), WriteLine().   1. Укажите, какие классы работают с каталогами и файлами. Перечислите их методы и свойства.   Ответ: Классы: File, FileInfo, Directory, DirectoryInfo, Path, FileStream.  Методы и свойства: AppendText(), CopyTo(), ReadAllBytes(), ReadAllLines(), Create(), CreateSubdirectory(), AltDirectorySeparatorChar, DirectorySeparatorChar.   1. Укажите последовательность действий при работе с файлами.   Ответ: Экземпляр FileStream применяется для чтения и записи данных в любой файл. Для создания экземпляра FileStream потребуется указать следующие фрагменты информации:   * Файл, к которому должен получаться доступ. * Режим открытия файла. * Вид доступа к файлу. * Общий доступ, который показывает, должен ли доступ к файлу быть эксклюзивным или же должна быть возможность доступа со стороны других потоков одновременно. |  |