ГУАП

КАФЕДРА № 42

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| канд. техн. наук, доцент |  |  |  | Н. В. Богословская |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 5 |
| ИСТОЧНИКИ ДАННЫХ. ЧТЕНИЕ И ЗАПИСЬ ТЕКСТОВЫХ ФАЙЛОВ: КЛАССЫ STREAMREADER И STREAMWRITER |
| по курсу: |
| ОСНОВЫ ПРОГРАММИРОВАНИЯ |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ гр. № | 4326 |  |  |  | Г. С. Томчук |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2024

СОДЕРЖАНИЕ

[1 Цель работы 3](#_Toc178912612)

[2 Задача 3](#_Toc178912613)

[3 Ключевые позиции 4](#_Toc178912614)

[3.1 Проверка работы свойств класса 4](#_Toc178912615)

[3.2 Автоматические свойства 6](#_Toc178912616)

[3.3 Проверка на прописной символ 6](#_Toc178912617)

[4 Тестирование программы 7](#_Toc178912618)

[ВЫВОДЫ 8](#_Toc178912619)

1. Цель работы

Выполнение работы имело следующие цели:

* Изучить и закрепить навыки работы с текстовыми файлами в C# с помощью классов StreamReader и StreamWriter, а также научиться использовать диалоговые окна для выбора файлов.
* Освоить работу с перегруженными методами и параметрами различных типов, включая ссылочные (ref), выходные (out) и необязательные параметры.
* Ознакомиться с особенностями передачи массивов как параметров и влиянием этого на их изменение.

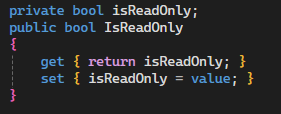
1. Задача

11. В приложении пользователь может создать объект класса Текстовый файл, используя классы Файл, Директория. Методы для работы с файлами должны полностью обеспечить пользователя возможностями создания, удаления, изменения, переименования файлов. Подзадачи лабораторной работы № 5 включают в себя:

* Разработать и протестировать перегруженные методы для чтения и записи текстовых файлов (EditFile() и ReadFile()) с использованием объектов классов OpenFileDialog и SaveFileDialog для выбора файлов.
* Реализовать чтение и сохранение метаинформации о файле, включая авторов, тип файла, дату создания и атрибут доступности (IsReadOnly), используя текстовые файлы.
* Создать метод ChangeAuthors для изменения массива авторов и продемонстрировать, что изменения сохраняются после выполнения метода, подтверждая передачу массива как ссылочного типа.
* Передать параметр по ссылке (ref) в одном из методов и убедиться, что изменения, сделанные в методе, остаются после его выполнения.
* Добавить метод с двумя выходными параметрами (out), протестировать его и продемонстрировать использование выходных параметров.
* Добавить метод с необязательным параметром и проверить его работу, показав, как необязательные параметры влияют на вызовы метода.
* Разработать пользовательский интерфейс для тестирования методов, используя различные элементы управления, подходящие для отображаемых данных, и задействовать разнообразные обработчики событий, помимо кнопок, для повышения удобства и современности интерфейса.

1. Ключевые позиции

На рис. 1, 2 показано

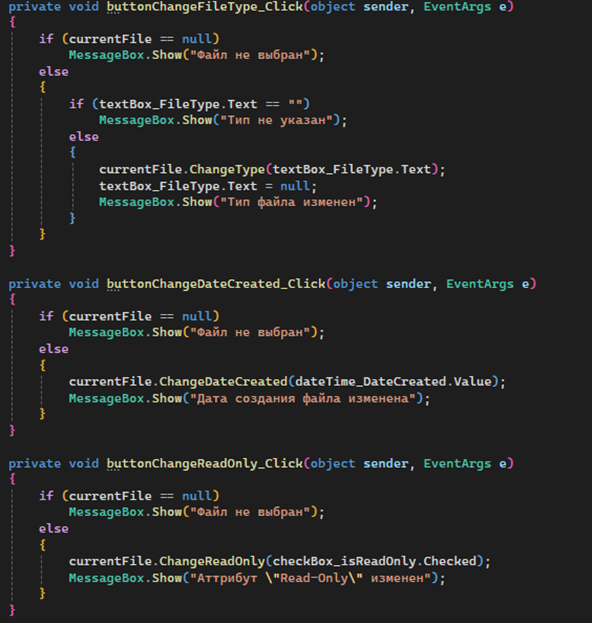


Рисунок



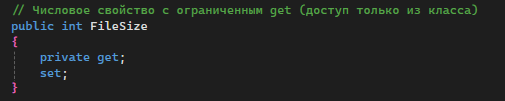
Рисунок 2

На рис. 3 изображены



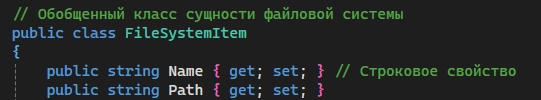
Рисунок

На рис. 4 показано



Рисунок

На рис. 5 изображены



Рисунок

На рис. 6 показано



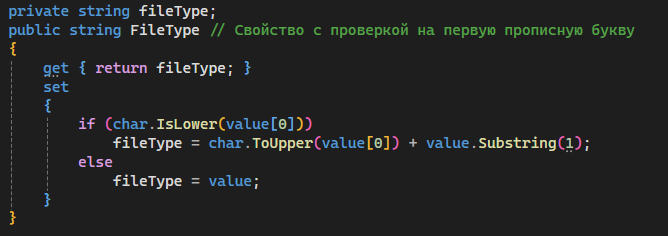
Рисунок

На рис. 7 показана



Рисунок

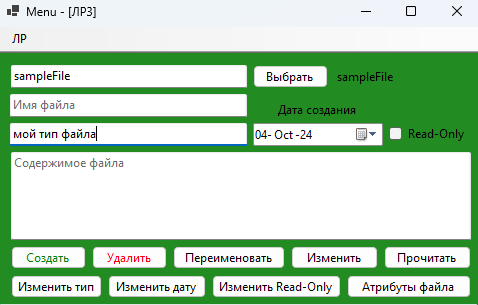
На рис. 8 изображено



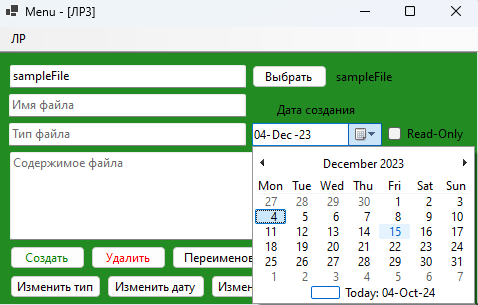
Рисунок

1. Тестирование программы

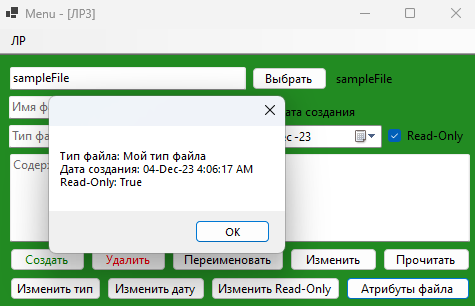
На рис. 9, 10, 11 представлены результаты тестирования программы.



Рисунок



Рисунок



Рисунок

ВЫВОДЫ

* Работа с текстовыми файлами с помощью StreamReader и StreamWriter позволяет легко реализовывать операции чтения и записи данных в файлы, а также предоставляет гибкость при обработке содержимого файлов.
* Работа с массивами как ссылочными типами продемонстрировала важность правильного понимания передачи параметров в методы. Изменения, сделанные в массиве, переданном в метод, сохраняются, что может быть полезно в ряде задач, но требует внимательного подхода, чтобы избежать нежелательных изменений данных.
* Передача параметров по ссылке (ref) и использование выходных параметров (out) обеспечивают дополнительную гибкость при разработке методов, позволяя сохранять изменения в параметрах или возвращать несколько значений из метода. Это особенно полезно, когда требуется возвращать не одно, а несколько значений.
* Использование необязательных параметров упрощает вызов методов и делает код более лаконичным, позволяя вызывать методы с минимально необходимыми аргументами или задавать стандартные значения для некоторых параметров.