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 № 3 |
| СВОЙСТВА КЛАССА |
| 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:

* Понять, как реализовывать и использовать свойства в классах и структурах, а также как управлять доступом к данным через блоки get и set.
* Научиться применять модификаторы доступа для отдельных блоков get и set, обеспечивая контроль над доступом к данным.
* Изучить возможности автоматических свойств, включая их модификацию и использование свойств только для чтения.
* Научиться проверять и изменять значения свойств, например, изменяя первый символ строки с строчного на прописной.
* Применить свойства для полей различных типов данных (числовые, строки, дата-время) и протестировать их работу.

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 № 3 включают в себя:

* Реализовать и протестировать работу свойств в классе и структуре.
* Проверить утверждения о модификаторах доступа для блоков set и get, используя свойства с различными модификаторами.
* Создать интерфейс для проверки возможностей автоматических свойств, включая их модификацию и ограничение доступа.
* Разработать свойство для проверки первого символа строки, с автоматической заменой строчного символа на прописной.
* Добавить и протестировать свойства для полей с различными типами данных: числовыми, строковыми и датой-временем.

1. Ключевые позиции
   1. Проверка работы свойств класса

На рис. 1, 2 показано определение свойства IsReadOnly класса FileClass и присвоение значения данному свойству в перегруженном конструкторе.

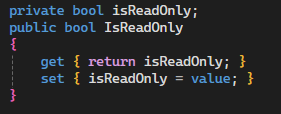


Рисунок 1



Рисунок

На рис. 3 изображены обработчики событий изменения типа файла, даты создания и значения "Read-Only" – три поля класса, заданные с помощью свойств.

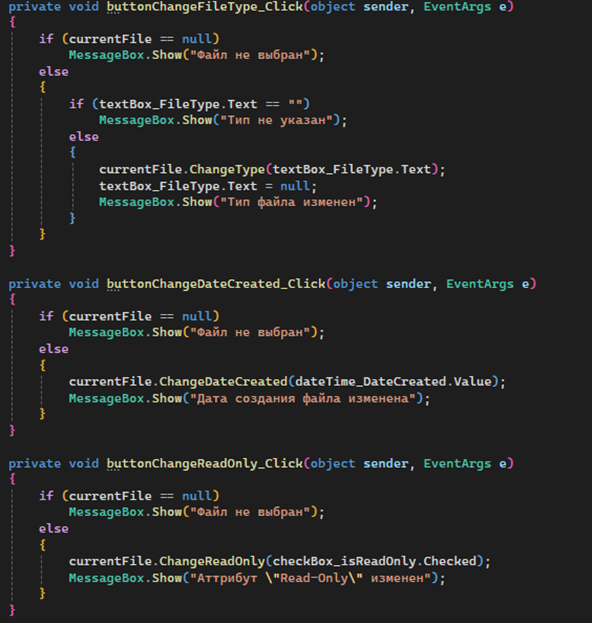


Рисунок 3

На рис. 4 показано свойство FileSize, имеющее модификатор доступа private на блоке get, что делает его недоступным для чтения вне класса FileClass.

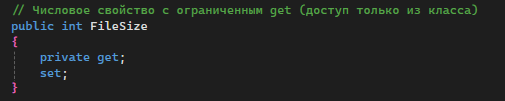


Рисунок 4

* 1. Автоматические свойства

На рис. 5 изображены два автоматических свойства Name и Path класса FileSystemItem. В данном случае компилятор сам создаст необходимые поля для свойств.

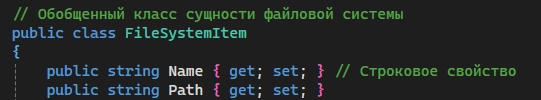


Рисунок 5

На рис. 6 показано свойство DateCreated, определенное внутри структуры fileMetadata, с типом данных DateTime.



Рисунок 6

На рис. 7 показана сокращенная запись свойства, эквивалентная полной записи, но без блока set.



Рисунок 7

* 1. Проверка на прописной символ

На рис. 8 изображено свойство, реализующее валидацию присваиваемого полю значения. При присвоении, блок set, в случае, если первый символ значения находится в нижнем регистре, заменяет его на символ в верхнем регистре и сохраняет поле с новым значением.

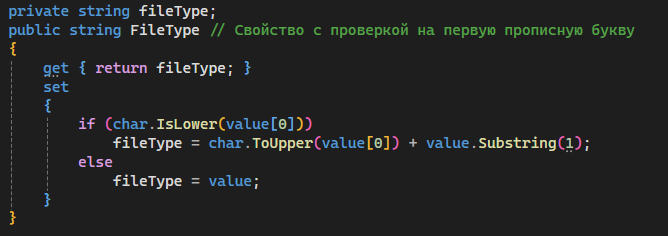


Рисунок 8

1. Тестирование программы

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

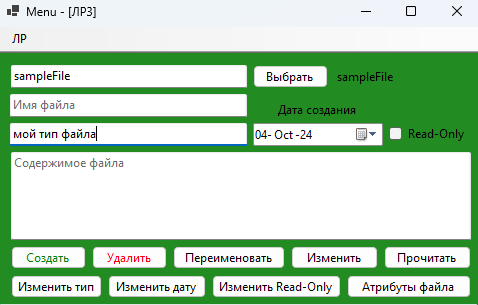


Рисунок 9

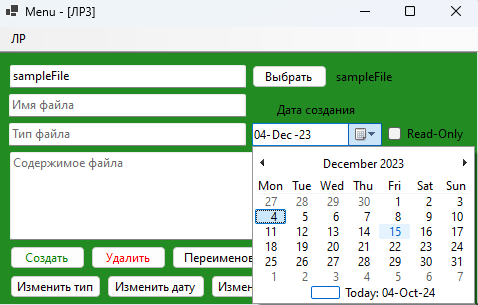


Рисунок 10

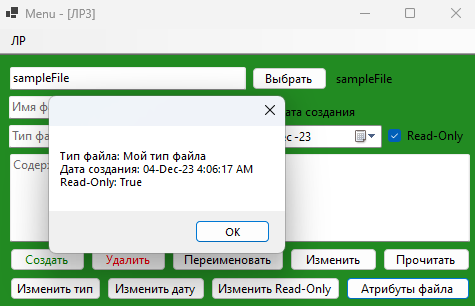


Рисунок 11

ВЫВОДЫ

* Свойства в C# представляют собой специальные члены класса или структуры, которые обеспечивают контроль доступа к полям через методы доступа get и set. Они позволяют скрывать внутреннюю реализацию полей и управлять доступом к ним, обеспечивая инкапсуляцию данных.
* Модификаторы доступа для блоков get и set позволяют гибко управлять доступом к свойствам, при этом они должны быть согласованы с модификатором самого свойства.
* Автоматические свойства упрощают код, но при этом могут иметь модификаторы доступа и быть только для чтения, что делает их универсальными и удобными для использования.
* Свойства позволяют реализовывать валидацию данных, например, проверку и корректировку введённых значений, что повышает надёжность приложения.
* Свойства эффективно работают с различными типами данных, позволяя управлять не только строками и числами, но и такими типами, как дата-время, что делает их универсальным инструментом в программировании на C#.