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 № 12 |
| ОТНОШЕНИЯ МЕЖДУ КЛАССАМИ И ОБЪЕКТАМИ |
| по курсу: |
| ОСНОВЫ ПРОГРАММИРОВАНИЯ |
|  |

РАБОТУ ВЫПОЛНИЛ

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

Санкт-Петербург 2024

СОДЕРЖАНИЕ

[1 Цель работы 3](#_Toc185551826)

[2 Задача 3](#_Toc185551827)

[3 Ключевые позиции 3](#_Toc185551828)

[3.1 Реализация методов 3](#_Toc185551829)

[3.2 Создание интерфейса 5](#_Toc185551830)

[4 Тестирование программы 6](#_Toc185551831)

[ВЫВОДЫ 9](#_Toc185551832)

1. Цель работы

Выполнение работы имело следующие цели: научиться проектировать взаимодействие между классами в объектно-ориентированном программировании с использованием интерфейсов, ассоциации, агрегации и композиции, а также реализовать эти механизмы в библиотеке классов для решения конкретных задач.

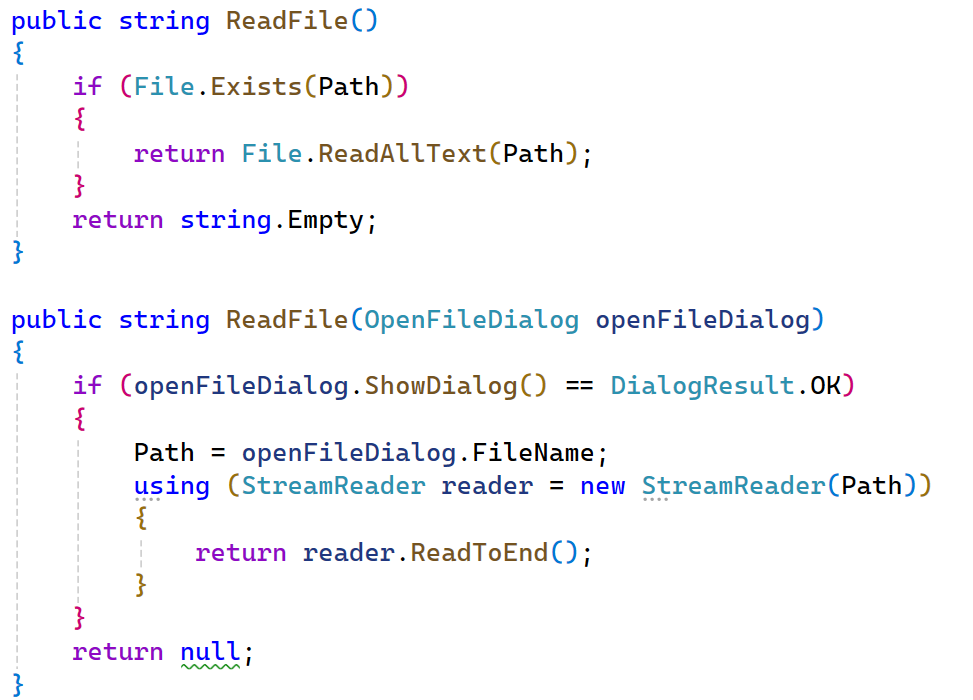
1. Задача

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 № 12 включают в себя:

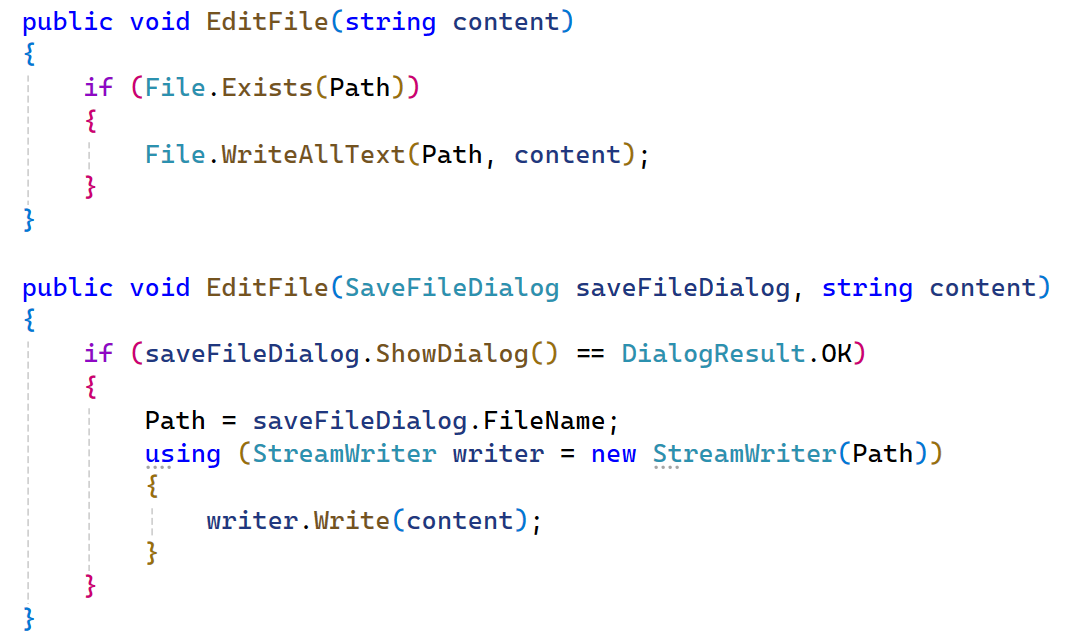
* Разработать интерфейс, обеспечивающий взаимодействие между объектами различных классов.
* Реализовать ассоциацию между классами, описав конкретный сценарий взаимодействия объектов.
* Выполнить агрегацию классов, продемонстрировав, как один класс использует другой в качестве независимого компонента.
* Реализовать композицию, где один класс управляет жизненным циклом другого, описав задачу взаимодействия в рамках этой связи.

1. Ключевые позиции
   1. Реализация методов

На рис. 1, 2 показаны.



Рисунок



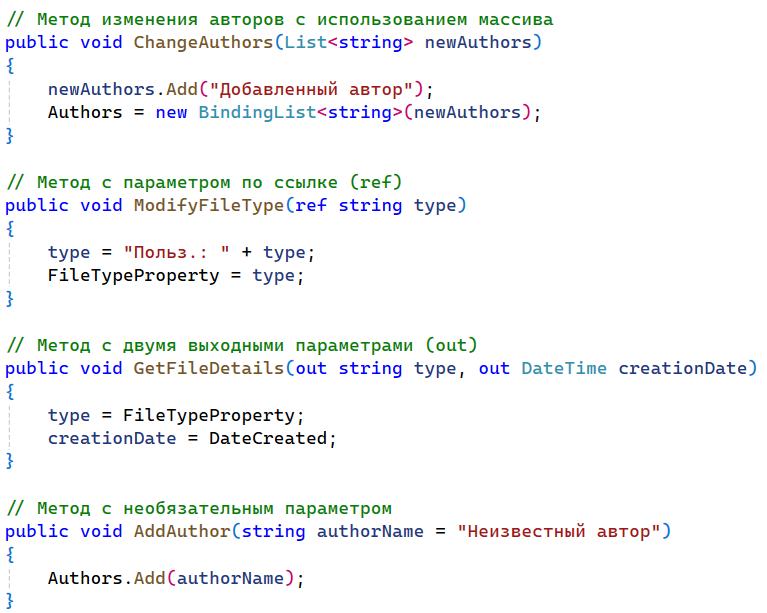
Рисунок

На рис. 3 изображены.



Рисунок

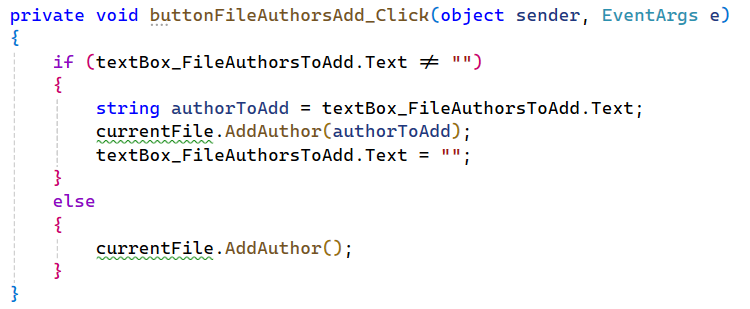
На рис. 4 показаны.



Рисунок

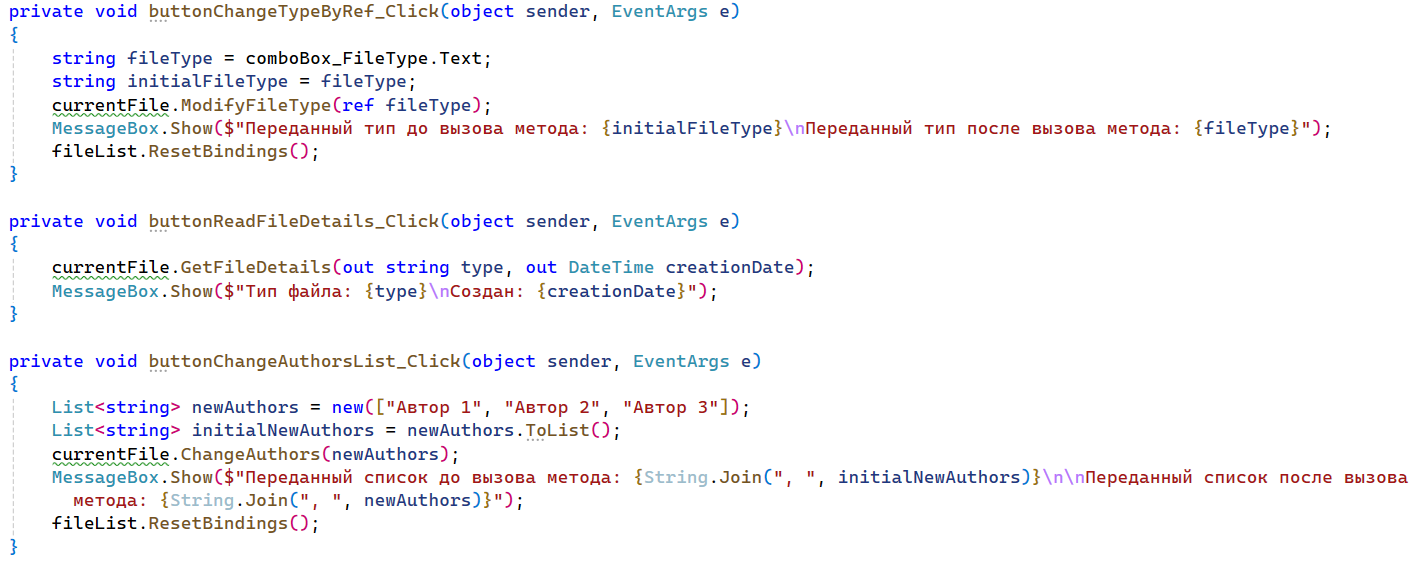
* 1. Создание интерфейса

На рис. 5 изображен.



Рисунок

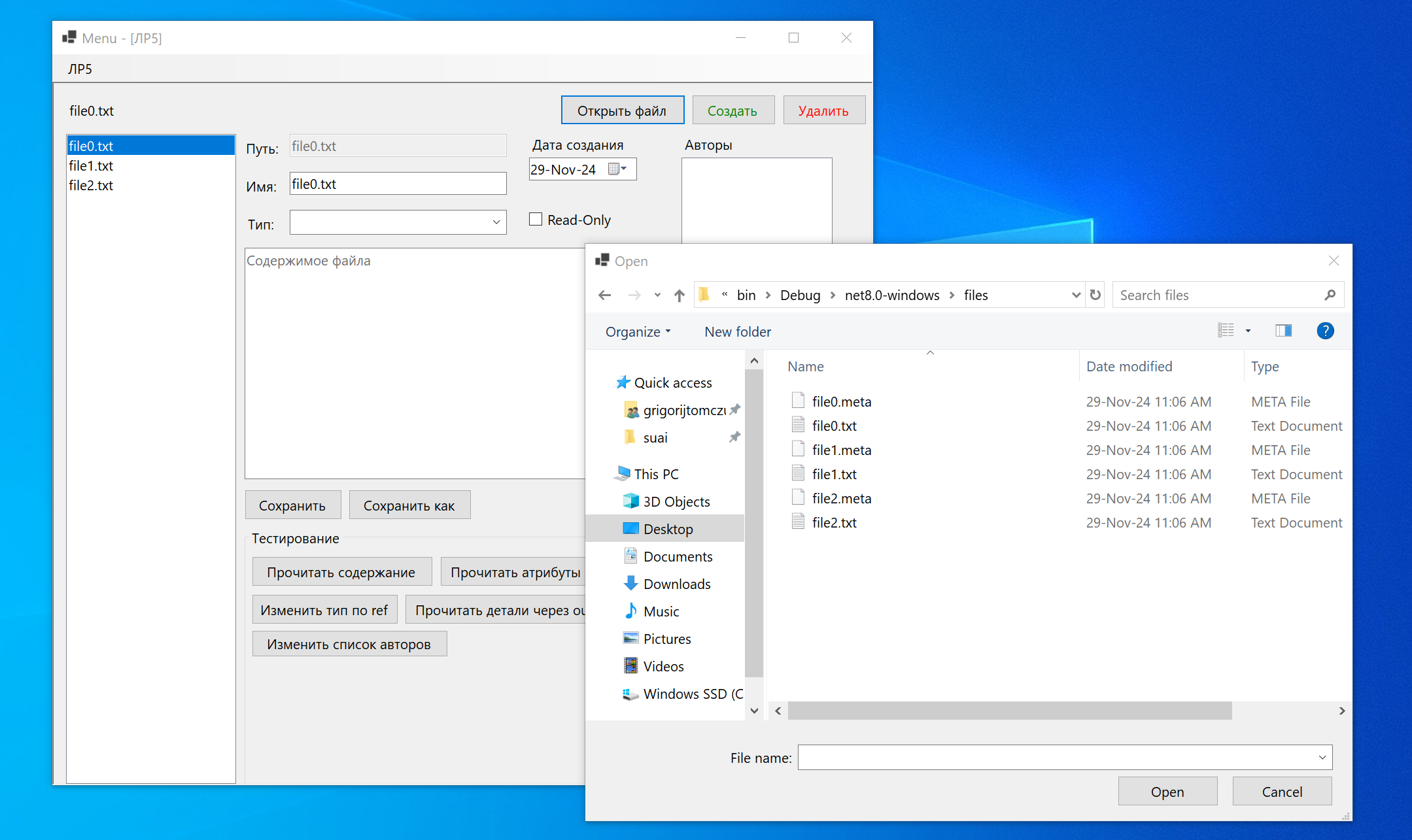
На рис. 6 показаны.



Рисунок

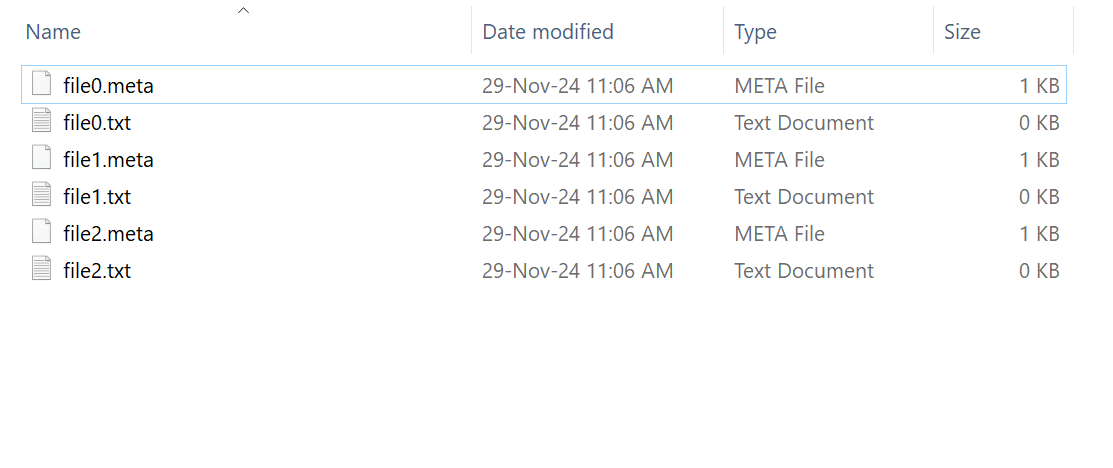
1. Тестирование программы

На рис. 7, 8, 9, 10, 11 представлены результаты тестирования программы. На рис. 7 изображено.



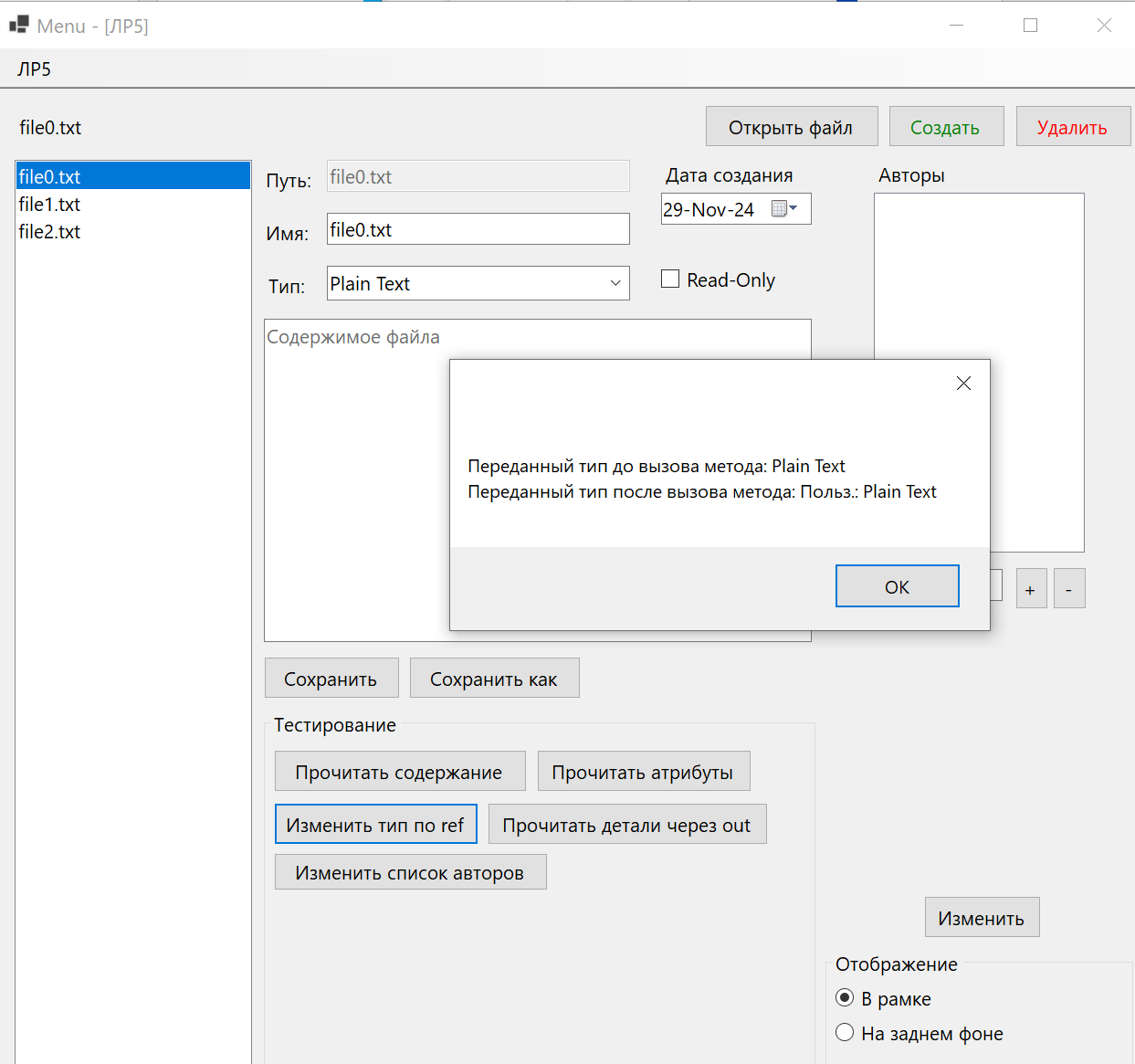
Рисунок

На рис. 8 показана.



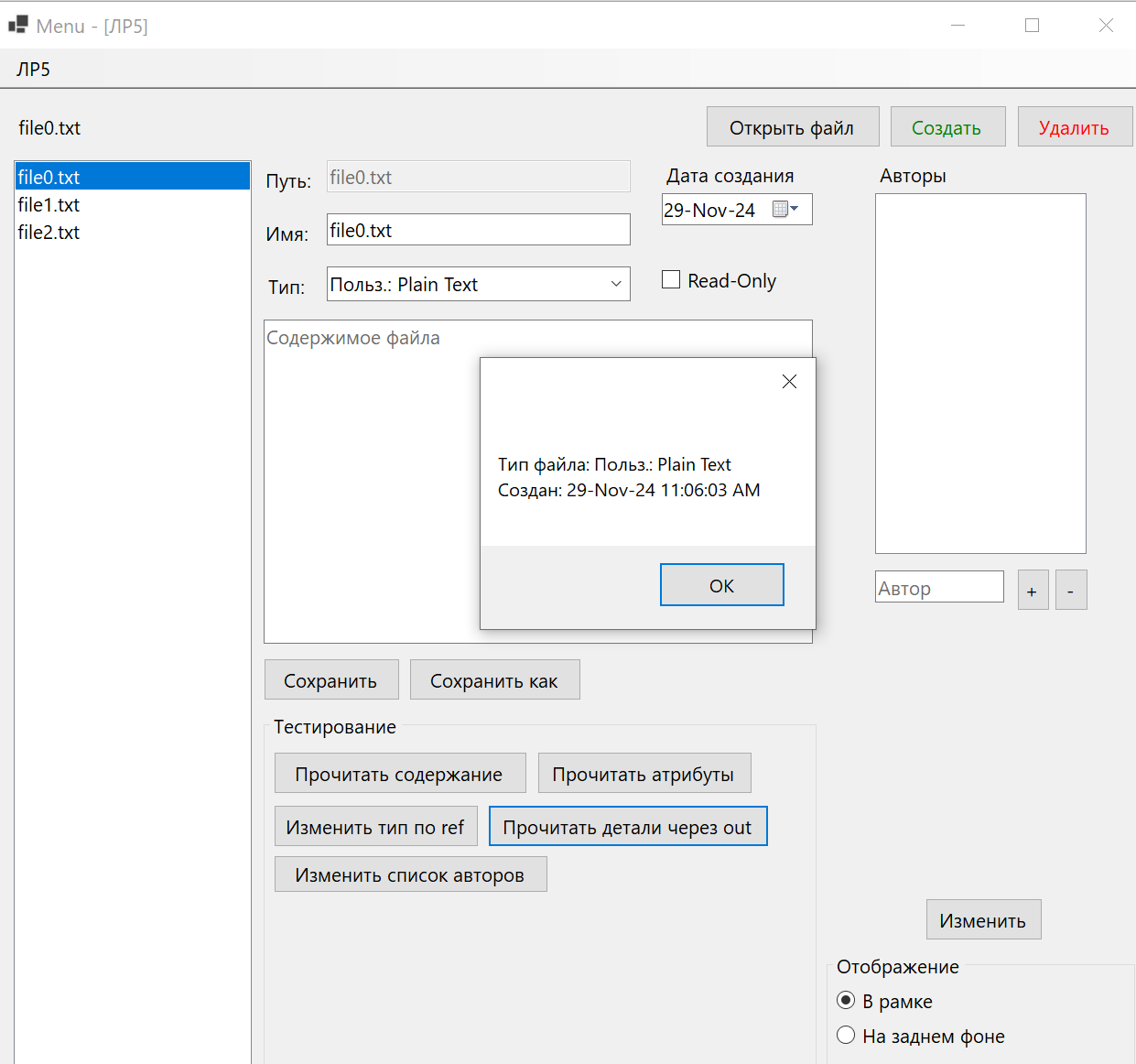
Рисунок

На рис. 9 показано.



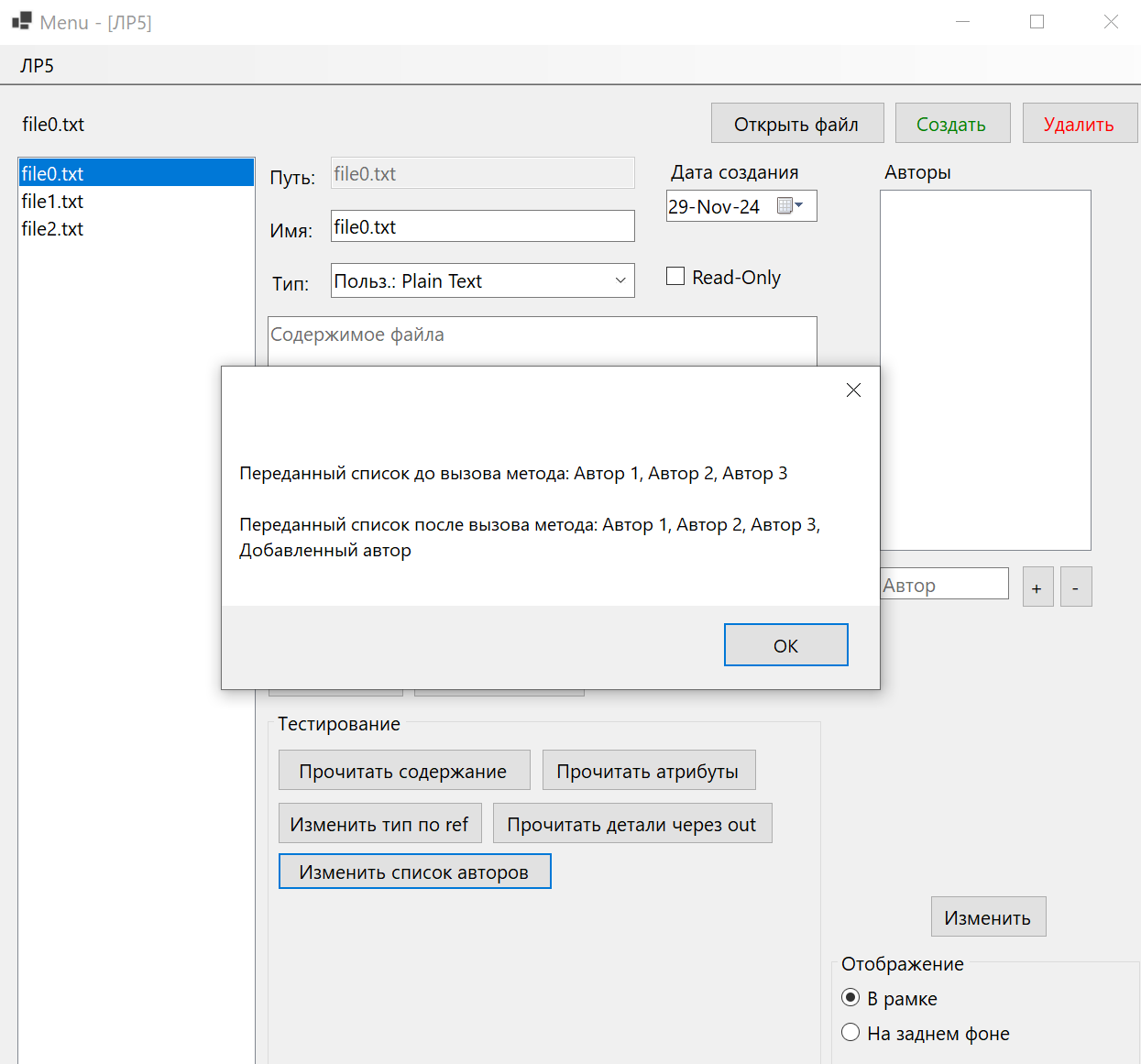
Рисунок

На рис. 10 показано.



Рисунок

На рис. 11 изображено.



Рисунок

ВЫВОДЫ

* Интерфейсы: Интерфейсы позволяют определять общие методы взаимодействия для различных классов, обеспечивая гибкость и совместимость при проектировании систем.
* Ассоциация: Ассоциация описывает слабую связь между объектами, где они могут взаимодействовать, но не зависят друг от друга напрямую. Это полезно для задач, где объекты временно обмениваются данными или выполняют совместные действия.
* Агрегация: Агрегация предполагает, что один объект содержит ссылки на другие объекты, которые могут существовать самостоятельно. Этот механизм удобен для случаев, когда компонент может быть использован в нескольких местах.
* Композиция: Композиция обеспечивает сильную связь, при которой один объект управляет жизненным циклом другого. Это подходит для задач, где дочерние объекты не должны существовать без родительского.
* Практическая значимость: Использование интерфейсов, ассоциации, агрегации и композиции позволяет проектировать гибкие, масштабируемые и устойчивые к изменениям системы. Эти механизмы облегчают разработку сложных программных решений, обеспечивая модульность и четкую структуру.