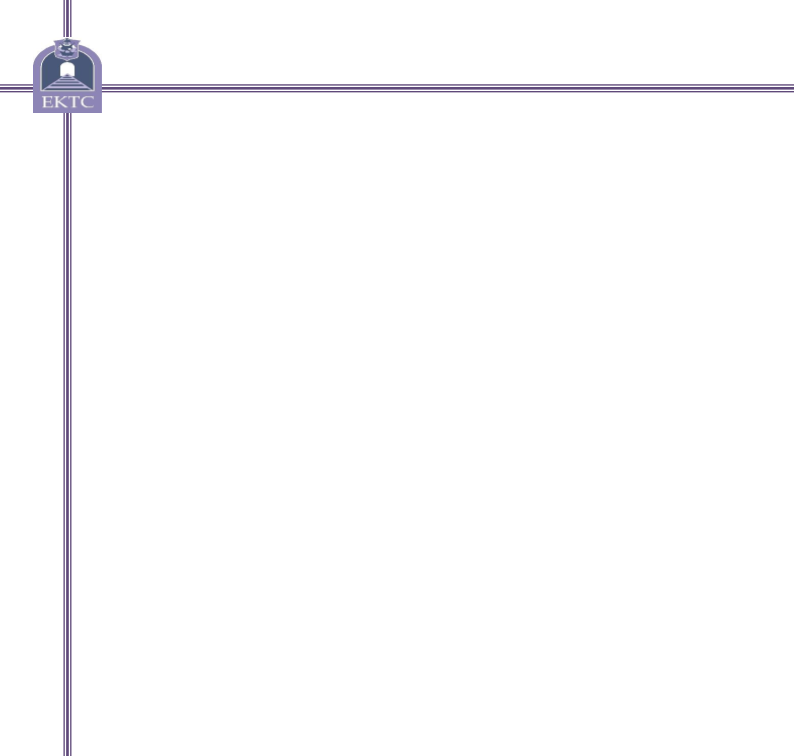
Министерство образования и молодежной политики Свердловской области ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчет по учебной практике УП 01.01



2024 г.

Выполнила: Мезенцева П.С.

Группа: ПР-21 Преподаватель: Ватолина Т.А

**Оглавление**

1. [Отчет по теме: «Текстовые файлы» 3](#_TOC_250004)
2. [Отчет по теме: «Разработка тестирующей программы» 6](#_TOC_250003)
3. [Отчёт по теме: «Разработка игры ‘Парные картинки’». 10](#_TOC_250002)
4. [Отчет по теме: «Разработка игры ‘Пятнашки’». 14](#_TOC_250001)
5. [Отчёт по теме: «Разработка документации». 18](#_TOC_250000)
6. Отчёт по теме: «Работа с базами данных. Индивидуальное задание» 23

## Задание:

# Отчет по теме: «Текстовые файлы»

Сделать так, чтобы после каждого слова стоял только один пробел или знак препинания.

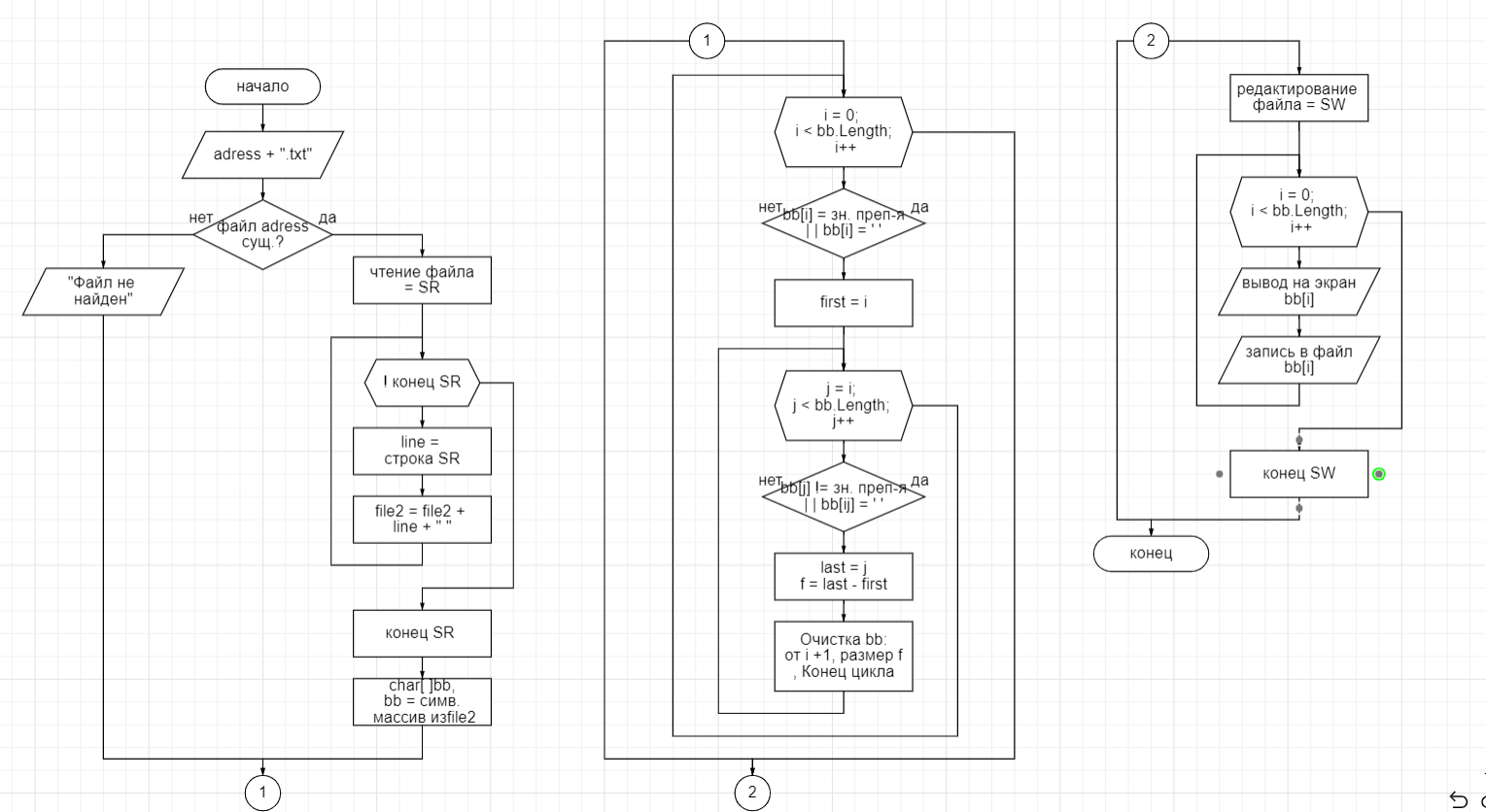
## Входные данные:

adress: string, вводится с клавиатуры, название файла

## Выходные данные:

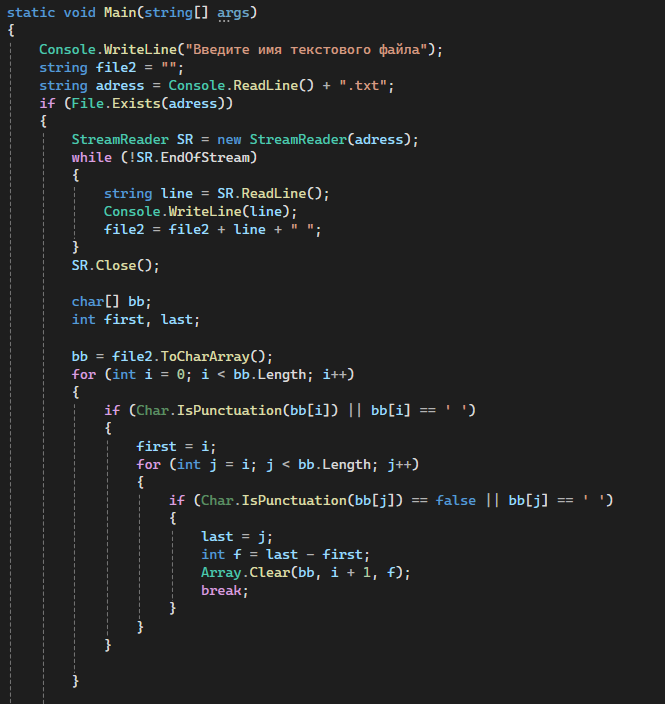
bb: стринговый массив, выводятся элементы по очереди на консоль

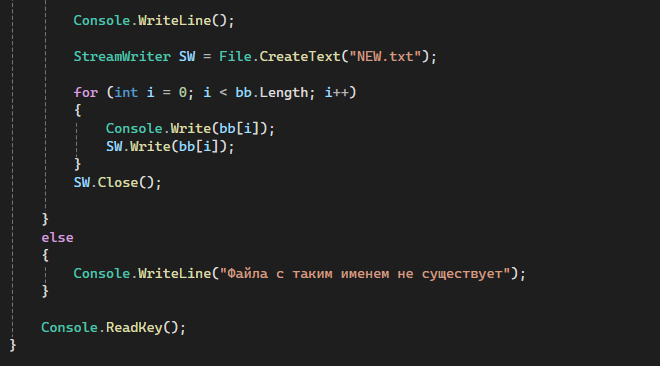
**Блок-схема** *(Рис.1)*



***Рис.1****. Блок-схема*

**Код программы** *(Рис.2):*

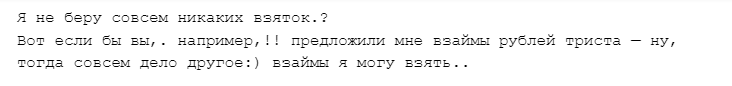




***Рис.****2. Код программы*

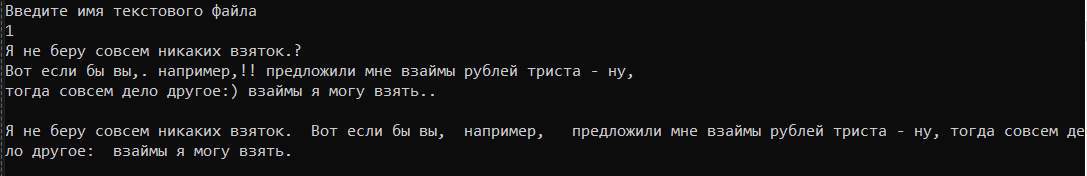
## Результат:

«Данные в ‘text.txt’» *(*Рис.3*)*



***Рис.****3****.*** *Содержимое ‘text.txt’*

## «Результат работы программы в ‘text.txt’» *(*Рис.4*)*



***Рис.4.*** *Результат.*

# Отчет по теме: «Разработка тестирующей программы»

**Задание:** Разработать объектно-ориентированное приложение, проводящее тестирование ученика, на основе информации, хранимой в файле. Общее количество вопросов в тесте не менее 5**.**

**Входные данные**: «one.txt» - файл с вопросами.

**Выходные данные:** сообщение о количестве правильных ответов, MessageBox, вывод всех результатов, string.

## Код программы:

* 1. Организация работы с radioButton (рис.1):

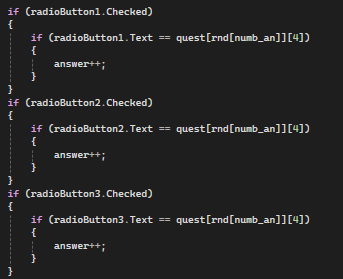


Рис.1. Код для работы с radioButton

* 1. Организация работы с checkBox (рис.2):

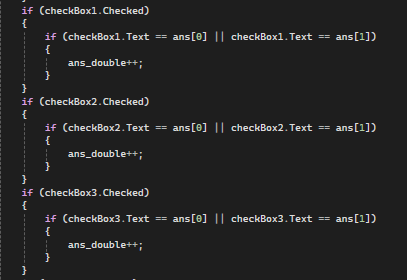


Рис.2. Код для работы с checkBox

* 1. Вывод вопросов с перемешиванием порядка (рис. 3):

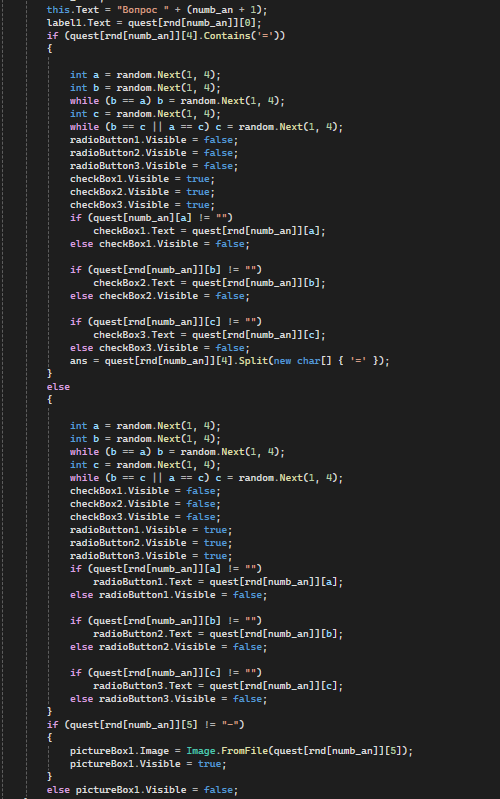


Рис.3. Вывод вопросов теста.

* 1. Шифрование файла с вопросами (рис. 4,5):



Рис. 4. Перенос оригинала файла с вопросами в другой файл.

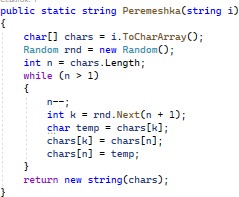


Рис.5. Метод шифрования вопросов в исходном файле.

* 1. Вывод результата теста на экран (рис. 6)

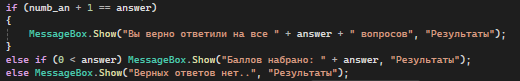


Рис.6. Вывод результата прохождения теста.

* 1. Удаление вопросов из теста *(*Рис.7*)*:

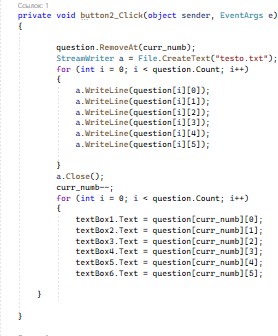


Рис.7.Удаление вопросов.

* 1. Изменение вопросов в тесте (Рис.8):

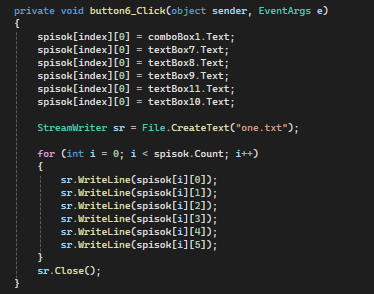


Рис.8.Изменение вопросов

* 1. Добавление вопросов в тест *(*Рис.9*):*

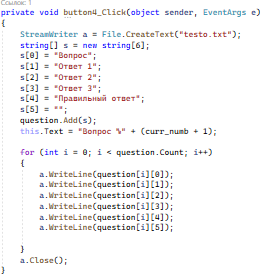


Рис.9.Добавление вопросов

## Пользовательский интерфейс:

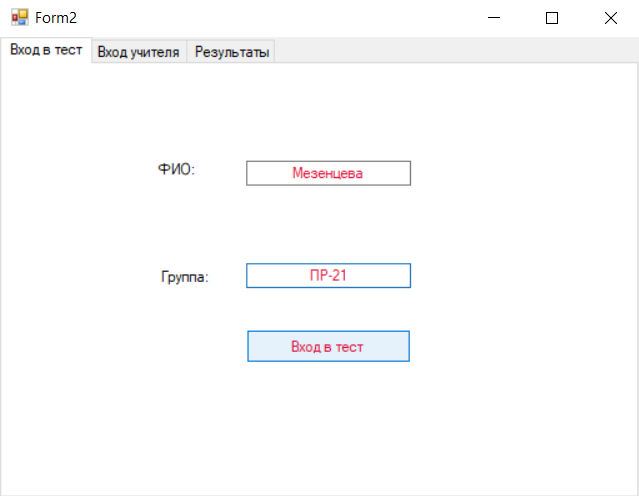
* 1. Регистрация студента (рис.10): 

Рис.10. Регистрация студента.

* 1. Форма входа преподавателя (рис.11):

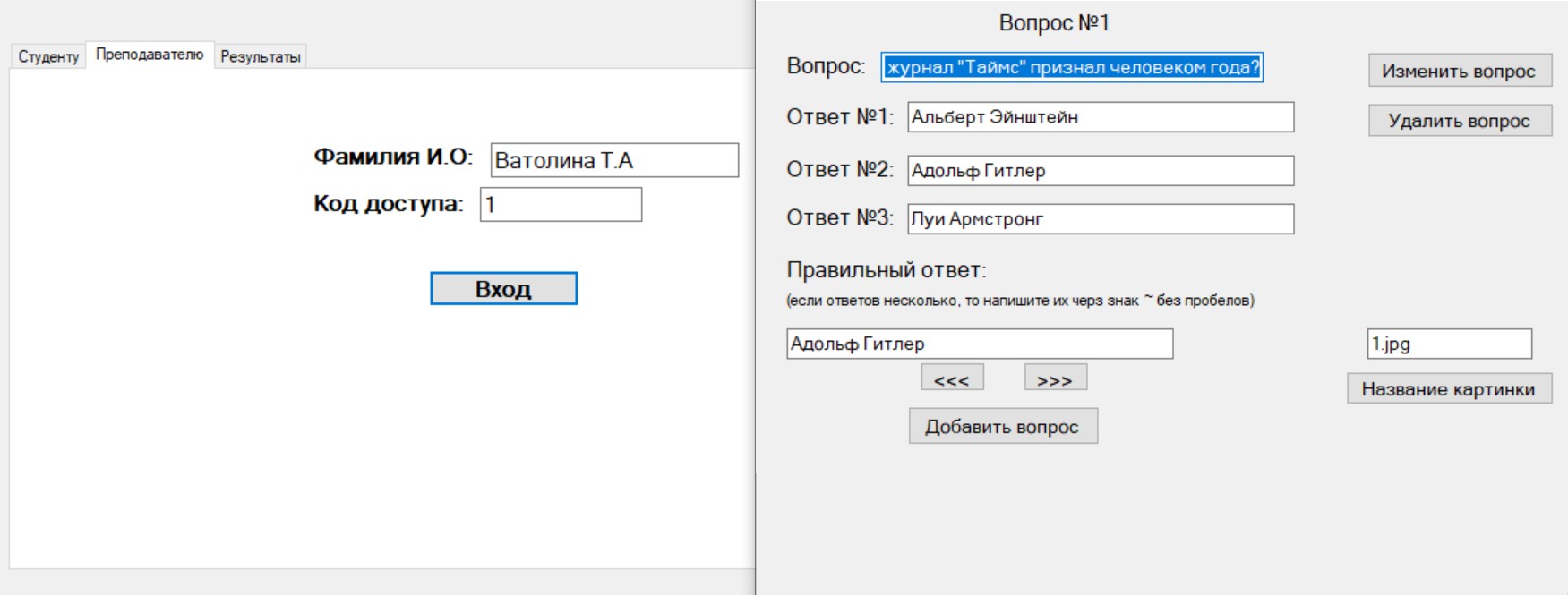


Рис.11. Вход преподавателя в режим конструктора.

* 1. Оформление вопросов (рис. 12):

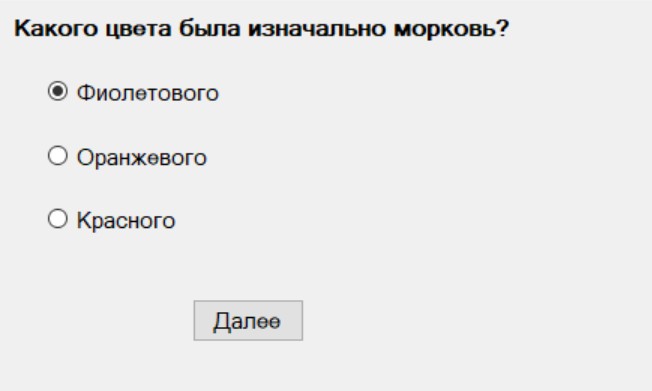


Рис.12. Оформление вопросов.

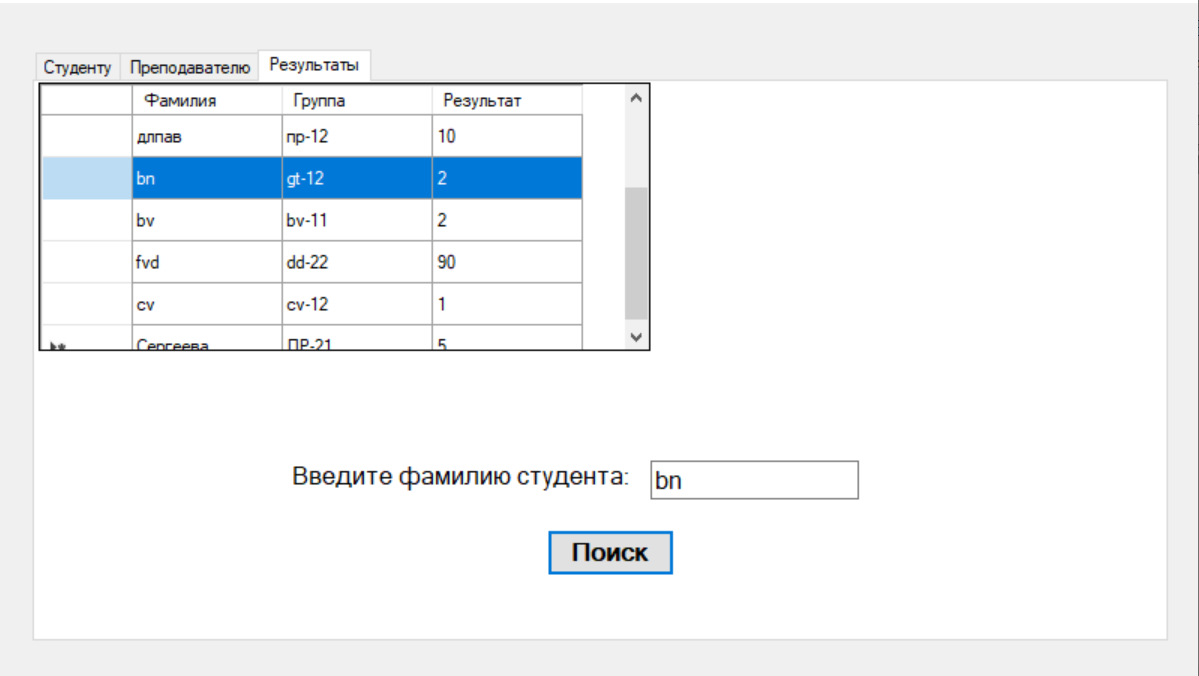
* 1. Вывод таблицы с результатами и поиск (рис.13):

Рис.13. Таблица с результатами.

# Отчёт по теме: «Разработка игры ‘Парные картинки’».

**Задание:** Разработать объектно-ориентированное приложение-игру

«Парные картинки»**.**

**Входные данные**: nickname - никнэйм (имя/псевдоним) пользователя,string; size - сложность игры, int.

**Выходные данные:** таблица с результатами игры.

## Код программы:

* 1. Создание новой разметки на форме (рис. 1):

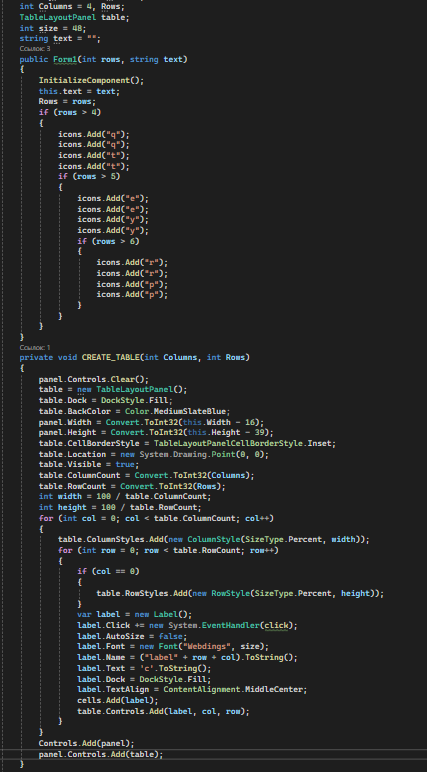
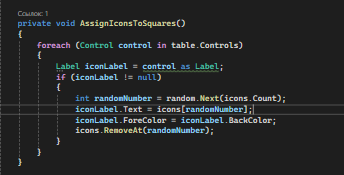


Рис. 1. Новая разметка на форме.

* 1. Размещение карточек на форме (рис.2):

Рис. 2. Размещение карточек на форме.

* 1. Переворачивание карточек в поиске пары с автоматическим переворачиванием (рис.3):

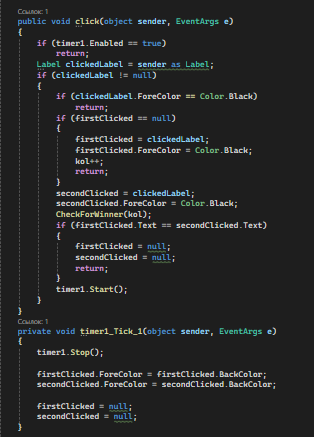


Рис.3. Переворачивание картинок.

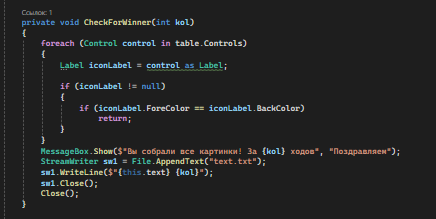
* 1. Вывод информации о количестве ходов с проверкой на то, что пользователь действительно прошёл игру (рис. 4):

Рис. 4. Вывод информации о количестве ходов.

* 1. Занесение информации о игроке в файл с результатами (рис. 5):



Рис. 5. Занесение информации о игроке в файл.

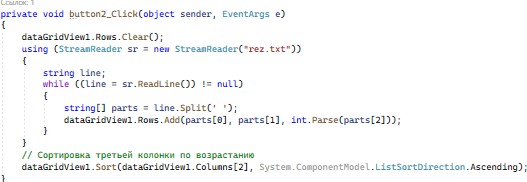
* 1. Сортировка результатов (рис.6):

Рис.6. Сортировка результатов.

## Пользовательский интерфейс:

* 1. Форма регистрации в игру (рис.7):

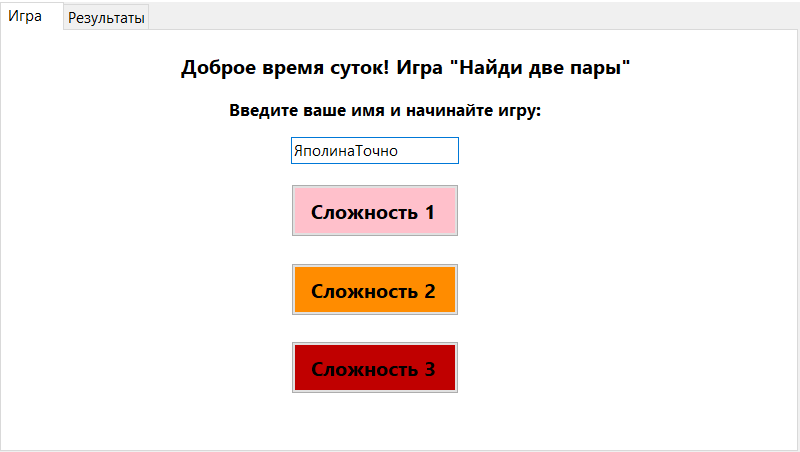


Рис.7. Регистрация в игре.

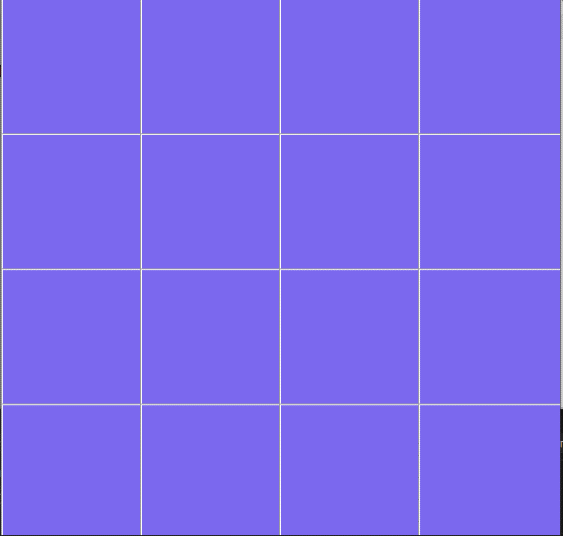
* 1. Внешний вид игрового поля (рис. 8):

Рис.8. Игровое поле.

* 1. Уведомление о результате игры (рис.9):

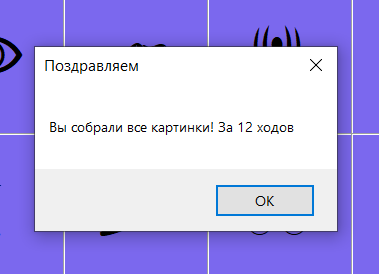


Рис. 9. Уведомление с результатом.

* 1. Отсортированная таблица результатов (рис.10):

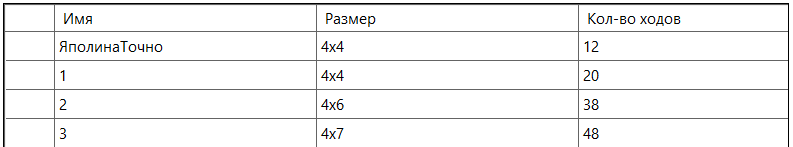


Рис. 10. Отсортированная таблица.

# Отчет по теме: «Разработка игры ‘Пятнашки’».

**Задание:** Разработать объектно-ориентированное приложение-игру

«Пятнашки»**.**

**Входные данные:** n **-** размер игры, int; text – никнэйм (имя/псевдоним) пользователя, string.

**Выходные данные:** таблица с результатами игры.

## Код программы:

* 1. Ввод никнэйма пользователя и сложности игры (рис.1):



Рис.1. Ввод данных пользователя и выбор сложности.

* 1. Вывод результатов в таблицу (рис.2):

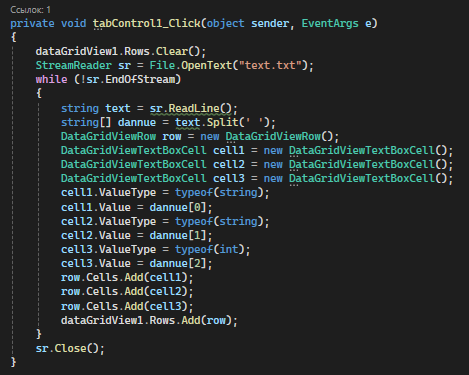


Рис. 2. Вывод результатов игры.

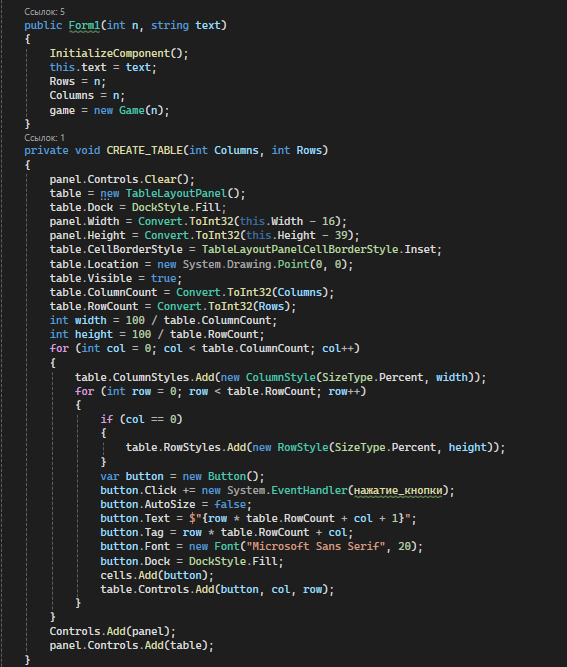
* 1. Создание новой разметки формы для игры (рис.3)

Рис. 3. Создание новой разметки.

* 1. Начало игры (рис. 4):

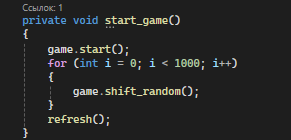


Рис. 4. Начало игры.

* 1. Метод, переводящий позицию и обратный (рис. 5,6):

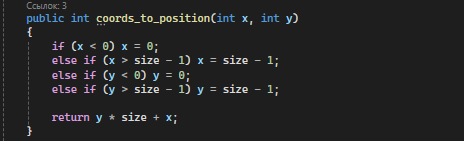


Рис. 5. Перевод позиции.

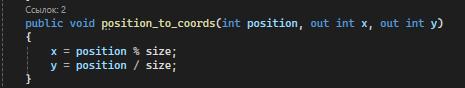


Рис. 6. Обратный метод.

* 1. Функция, готовящая поле к игре (рис.7):

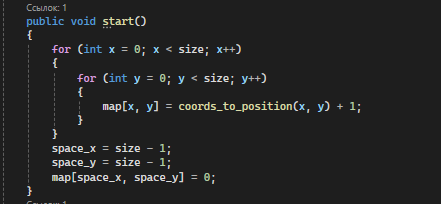
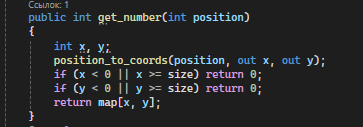


Рис.7. Подготовка к игре.

* 1. Метод позволяющий по тегу (позиции) возвращать надпись на кнопке, в зависимости от ее положения в массиве (рис. 8):

Рис. 8. Надпись на кнопке в зависимости от положения.

* 1. Обновление форма (рис. 9):

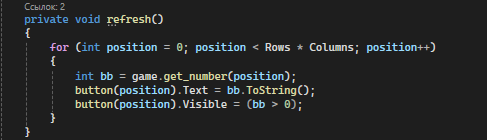


Рис. 9. Обновление формы.

* 1. Обмен надписей (рис. 10):

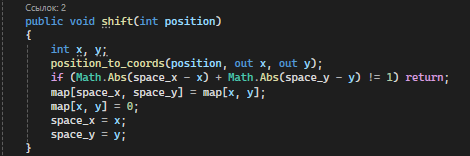


Рис. 10. Обмен надписей.

* 1. Перемешивание надписей на кнопках (рис.11):

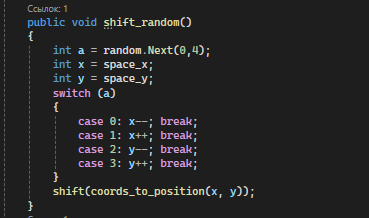


Рис. 11. Перемешивание надписей.

* 1. Проверка результата игры (рис. 12):

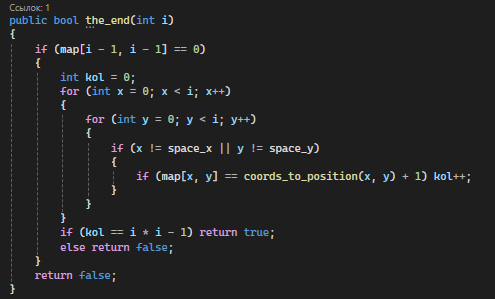


Рис.12. Проверка результата игры.

## Пользовательский интерфейс:

* 1. Регистрация в игру (рис.13):



Рис. 13. Регистрация.

* 1. Игровой интерфейс (рис.14):

Рис. 14. Игра.

* 1. Таблица с результатами (рис.15):

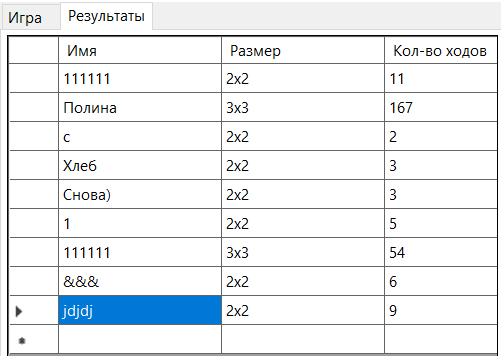


Рис.15. Таблица с результатами.

# Отчёт по теме: «Разработка документации».

## Задание:

**\*2 строк.**

**Входные данные:** nameFile - имя файла, вводится с клавиатуры, string.

**Выходные данные:** max\_str – Самая длинная общая подстрока, string.

* 1. Техническое задание (рис.1,2).

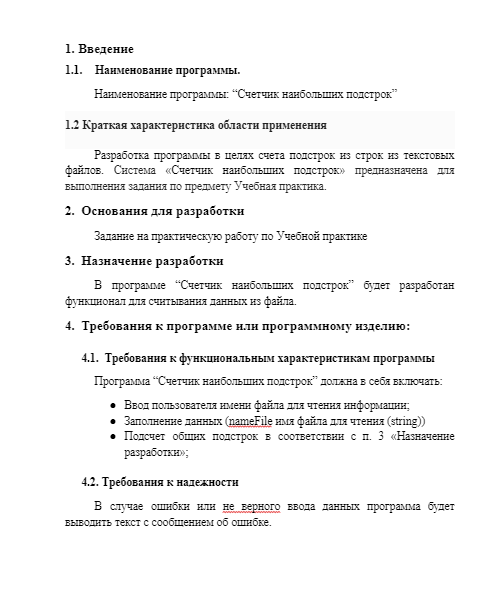


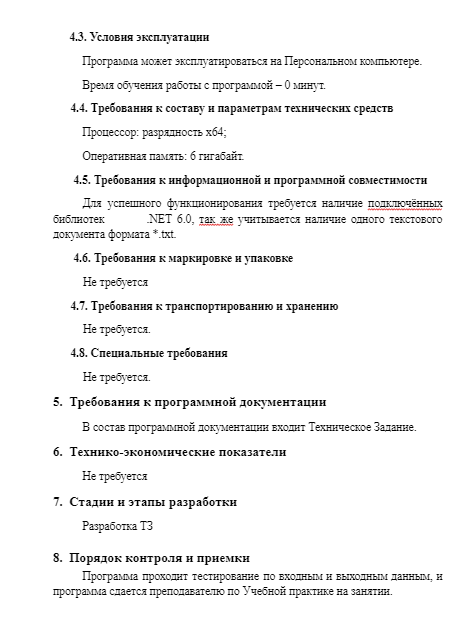
Рис.1. Техническое задание, страница номер 2

Рис. 2. Техническое задание, страница номер 3.

* 1. Тест-план с описанием входных и выходных данных (рис. 3,4).

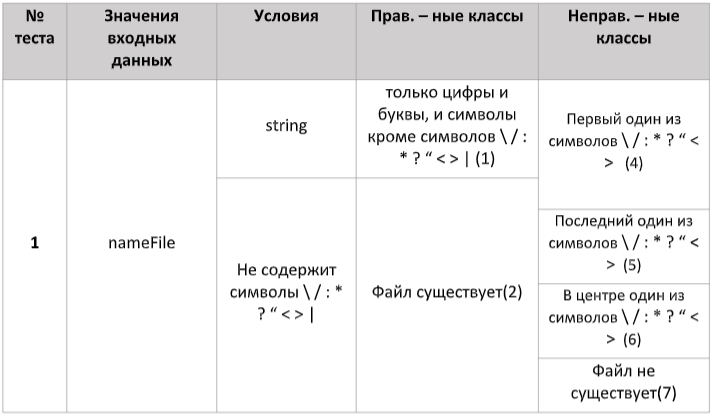


Рис. 3. Тест-план, ограничения.

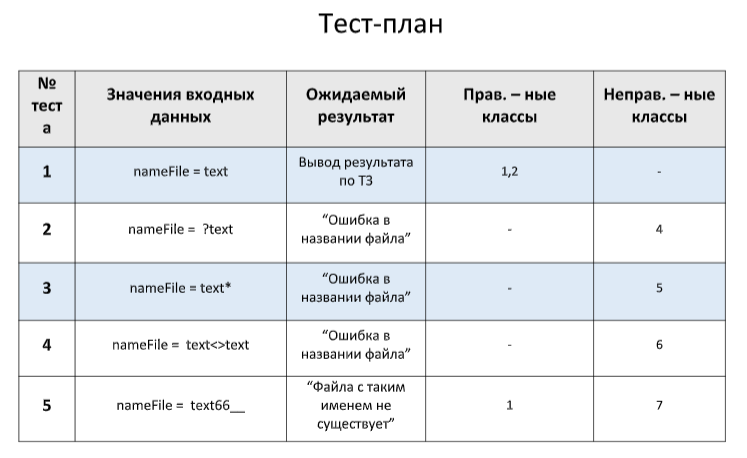


Рис.4. Тест-план, тесты.

* 1. Код программы.
     1. Проверка на запрещенные символы введённых пользователем данных (рис.5):



Рис. 5. Проверка на запрещенные символы.

* + 1. Считывание строк с заменой пустых вхождений (рис. 6):

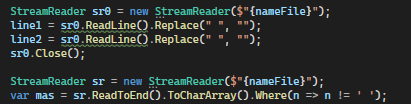


Рис.6. Замена пустых вхождений.

* + 1. Цикл, проходящий по строкам, запоминающий самые длинные подстроки(рис.7)

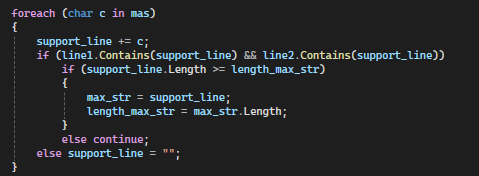


Рис. 7. Цикл, запоминающий самые длинные подстроки.

* + 1. Обработка полученной из класса Reader информации, обработка для вывода пользователю. (рис.8):

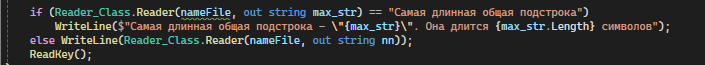


Рис. 8. Вывод пользователю.

* 1. Модульные тесты.
     1. Тесты на проверку корректности введенного названия файла пользователем (рис.9):

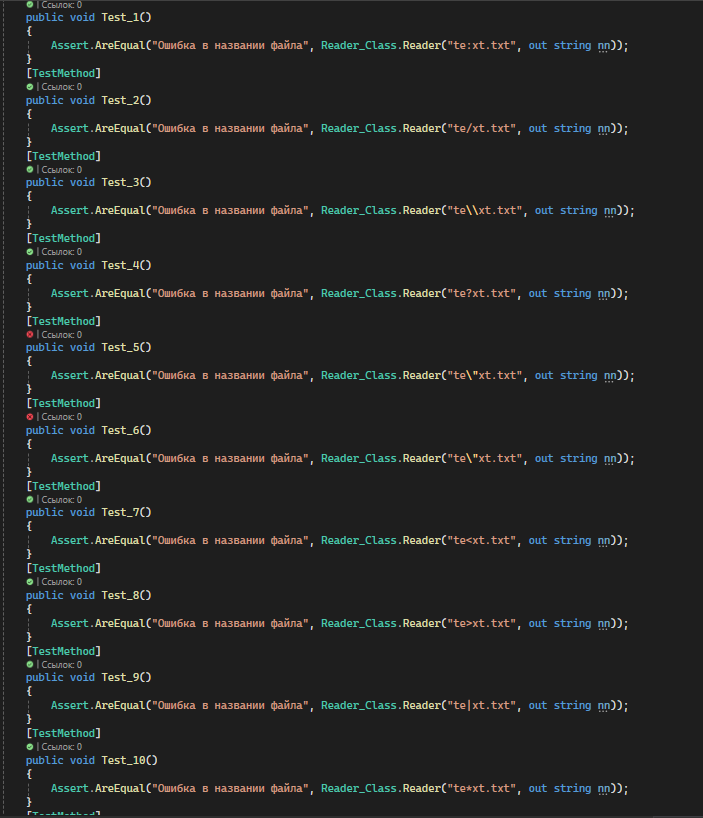


Рис. 9. Тесты на проверку корректности.

* + 1. Тест на вывод сообщения о том, что файл введенный пользователем не найден (рис.10):

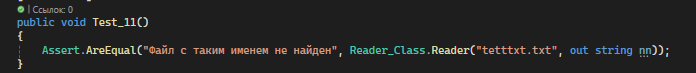


Рис. 10. Тест на существование файла.

* + 1. Тест на вывод сообщения о нахождении самой длинной общей подстроки (рис.12):

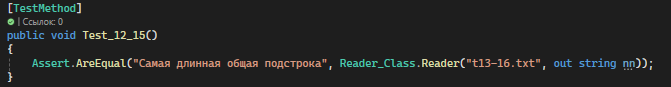


Рис. 12. Тест на нахождение самой длинной подстроки.

1. **Отчёт по теме: «Работа с базами данных. Индивидуальное**

**задание»**

**Задание:** Разработать базу данных по варианту или самостоятельно выбранной теме. Сделать запросы на вывод информации из всех таблиц, поиск в таблицах через inner join, редактирование и удаление информации, реализовать несколько составных запросов.

**Входные данные:** данные из базы данных «LibraryDB».

**Выходные данные:** данные, основанные на запросах пользователя.

## Таблицы в режиме конструктора:

* 1. Таблица «Books» (рис.1):

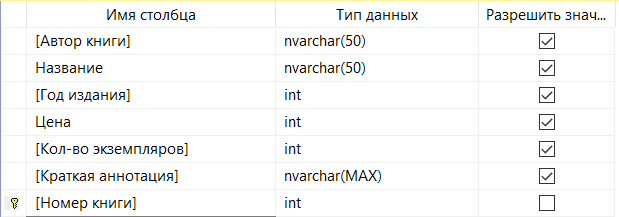


Рис.1. Таблица «Racers».

* 1. Таблица «Dates» (рис.2):

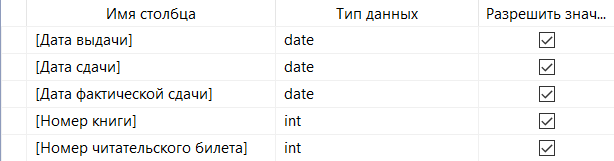


Рис.2. Таблица «ResultsCanada».

* 1. Таблица «Readers» (рис.3):

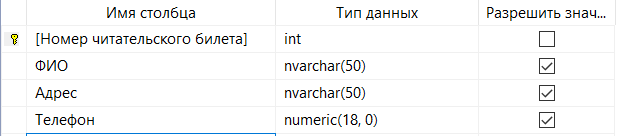


Рис.3. Таблица «Гран-при».

## Код программы:

* 1. Вывод информации из таблицы (рис.4,5,6):

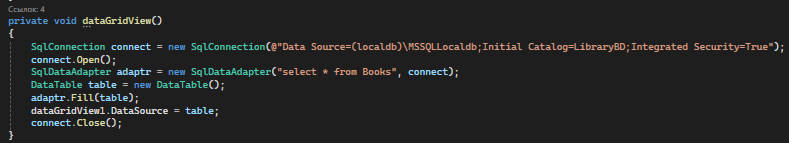


Рис.4. Вывод информации из таблицы «Books».

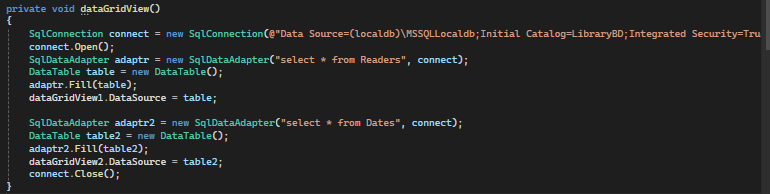


Рис.5. Вывод информации из таблиц «Readers» и «Dates».

* 1. Поиск по номеру читательского билета (рис.8):

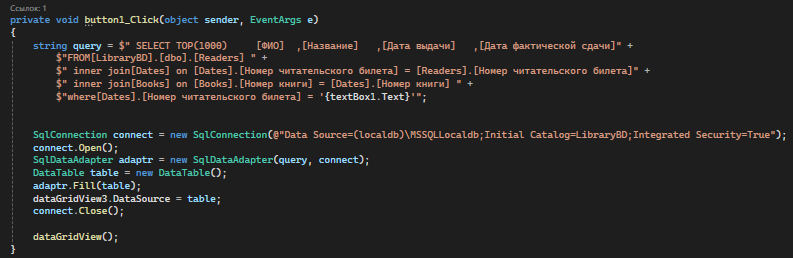


Рис.8. Поиск по номеру читательского билета.

* 1. Поиск за кем закреплена книга (рис.9):

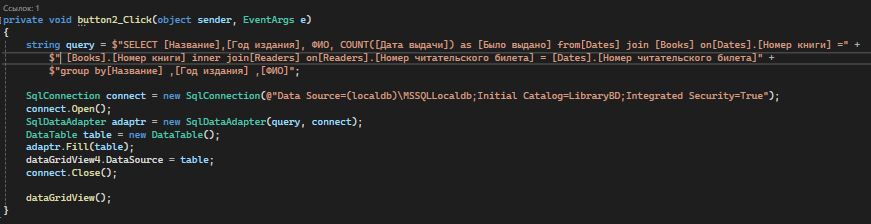


Рис.9. Поиск за кем закреплена книга.

* 1. Анализ должников, кто не сдавал книги больше 120 дней (рис.10):

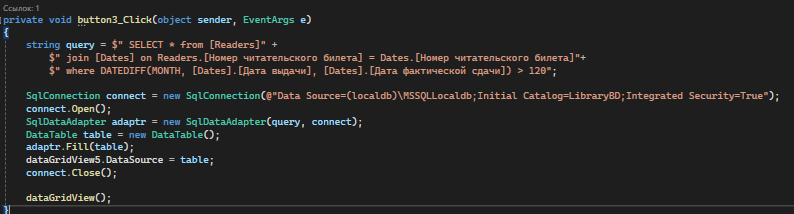


Рис.10. Анализ должников, кто не сдавал книги больше 120 дней.

* 1. Удаление по номеру книги (рис.11):

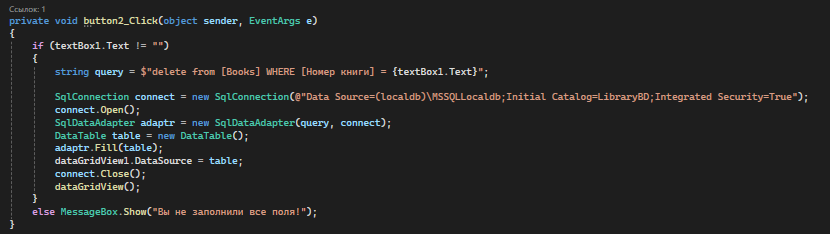


Рис.11. Удаление.

* 1. Добавление книги (рис.12)

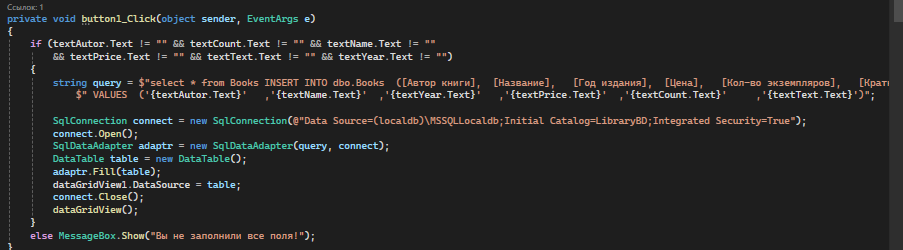


Рис.12. Добавление.

* 1. Редактирование выбранной строки (рис.13)

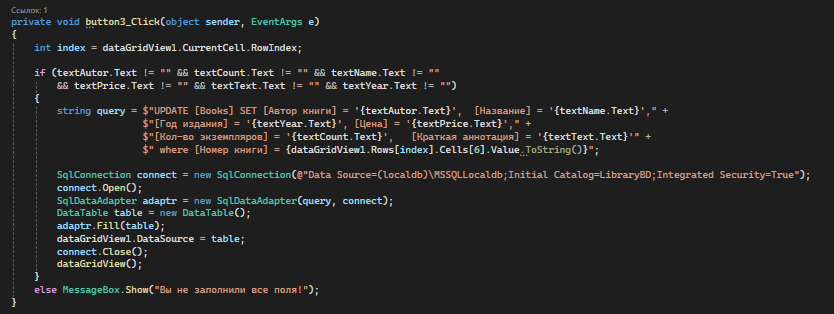


Рис. 13. Редактирование

## Пользовательский интерфейс:

* 1. Главная форма (рис.14):

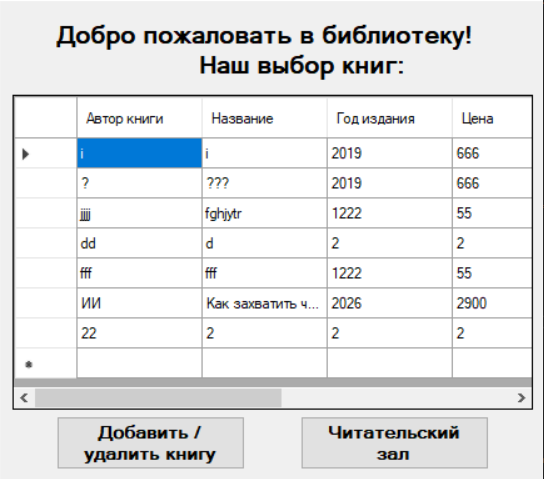


Рис. 14. Главная форма.

* 1. Добавление, удаление и редактирование книги, вторая форма (рис.15):

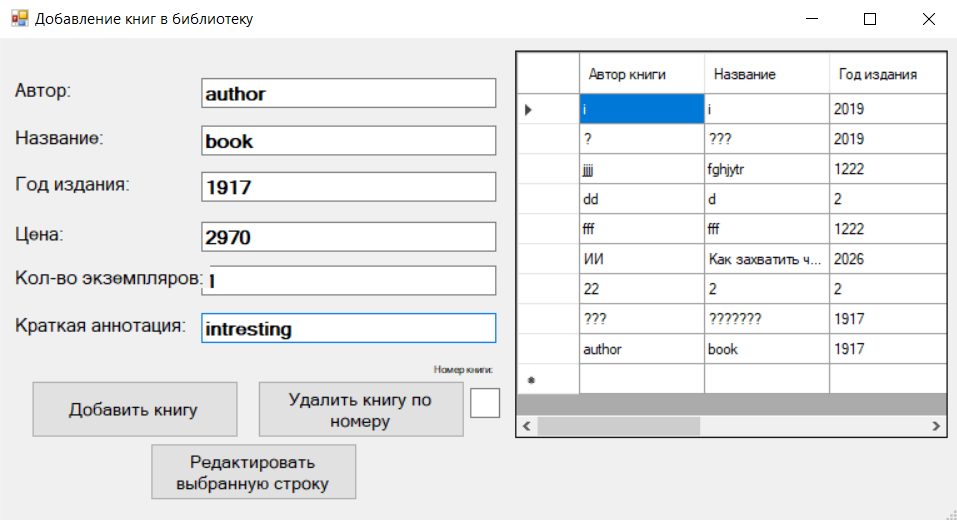


Рис. 15. Добавление, удаление и редактирование книги.

* 1. Выполнение различных запросов, третья форма(рис.16):

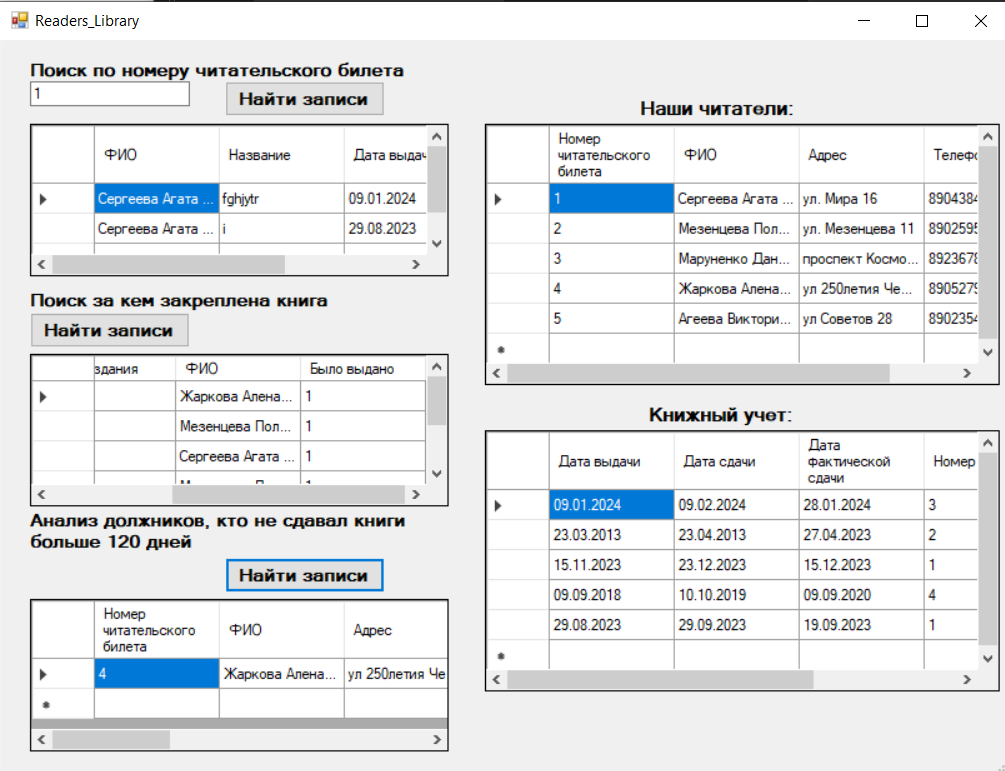


Рис. 16. Выполнение различных запросов.