**Тестирование**

Модульное тестирование приложения можно провести используя конструкцию try catch. Блок try содержит код, который может вызвать исключение. Если в этом коде возникает исключительная ситуация, т.е. ошибка, то выполнение кода мгновенно переходит на блок catch, в котором предположительно проходит обработка этой ситуации. В приложении при обработке исключения выводится сообщение об ошибке. Данная конструкция используется в методах добавления данных.



Рисунок 1 – Обработка исключения

Данные для модульного тестирования представлены в таблицах 1 – 4

Таблица 1 - Тест-кейсы для формы добавления записи в таблицу Клиенты

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Входные данные | Цель | Описание | Ожидаемый результат | Результат |
| 1 | Ввод корректных данных:  Иван  Иванов  Иванович  89000000000  (Рисунок 19) | Тестирование ввода данных | Тестирование проверки корректности данных и вводу их в таблицу | Создание новой строки в таблице Клиенты | Создание новой строки в таблице Клиенты |
| 2 | Ничего не вводить  (Рисунок 20) | Тестирование ввода данных | Тестирование проверки корректности данных и вводу их в таблицу | Вывод сообщения об ошибке | Вывод сообщения об ошибке |
| 3 | Ввод некорректных данных (номер телефона превышает 11 символов)  Иван  Иванов  Иванович  8849524856875873487  (Рисунок 21) | Тестирование ввода данных | Тестирование проверки корректности данных и вводу их в таблицу | Вывод сообщения об ошибке | Вывод сообщения об ошибке |

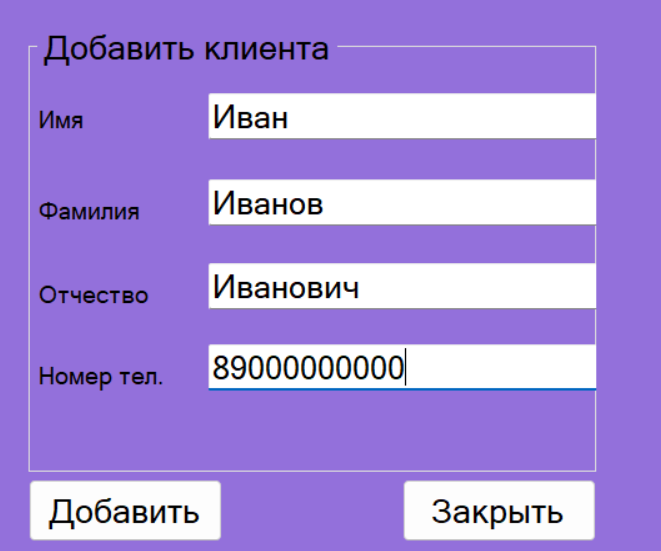


Рисунок 2 – Тестирование добавления корректных значений

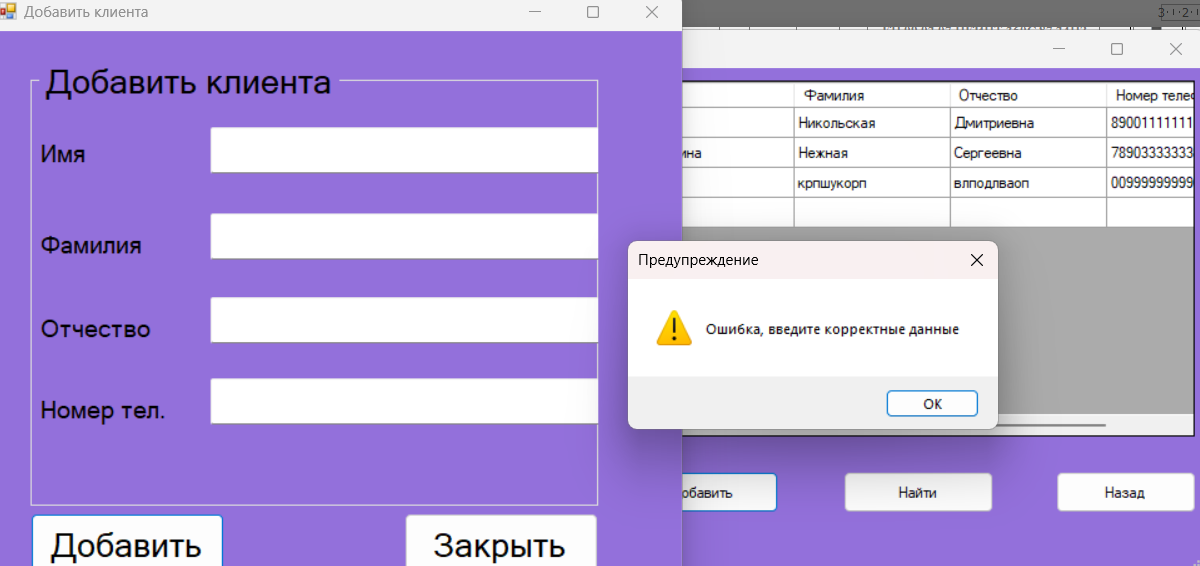


Рисунок 3 – Тестирования добавления пустых значений

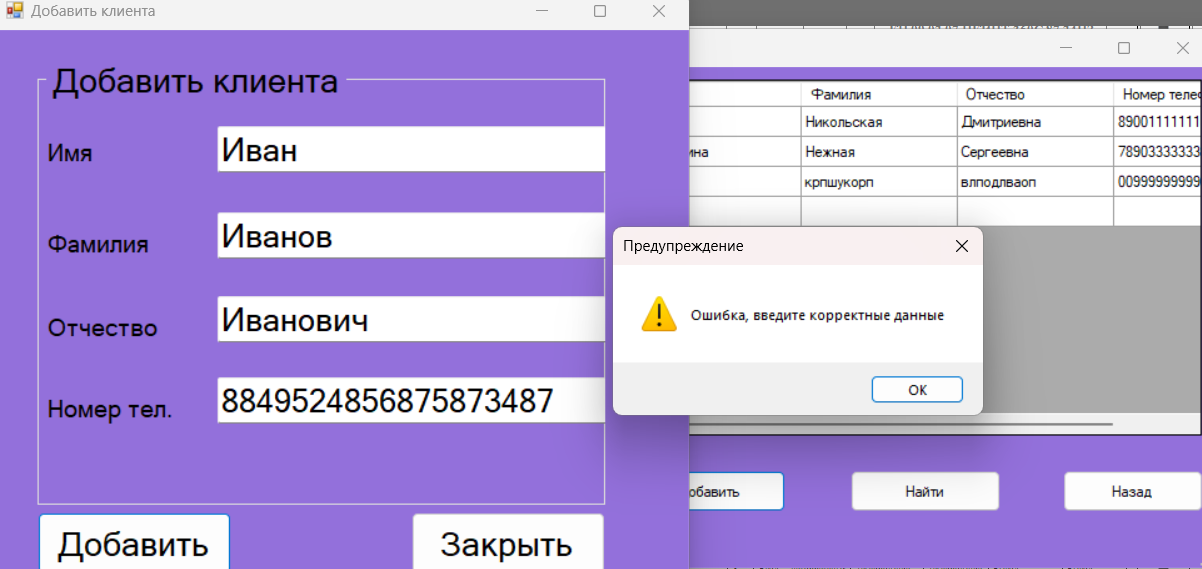


Рисунок 4 – Тестирование добавления корректных значений

Таблица 2 - Тест-кейсы для формы поиска записей в таблице Клиенты

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Входные данные | Цель | Описание | Ожидаемый результат | Результат |
| 1 | Ввод корректных данных:  Анна  (Рисунок 22) | Тестирование поиска | Тестирование нахождения данных соответствующих поиску | Выделение найденной строки | Выделение найденной строки |
| 2 | Ничего не вводить  (Рисунок 23) | Тестирование поиска | Тестирование нахождения данных соответствующих поиску | Вывод сообщения об ошибке | Вывод сообщения об ошибке |
| 3 | Ввод некорректных данных (Отсутствуют в таблице)  Пётр  (Рисунок 24) | Тестирование поиска | Тестирование нахождения данных соответствующих поиску | Вывод сообщения об ошибке | Вывод сообщения об ошибке |

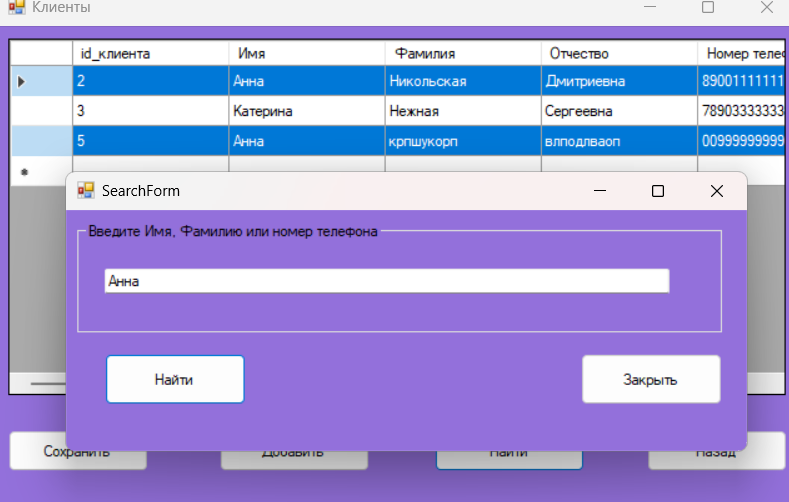


Рисунок 5 – Тестирование поиска корректных значений

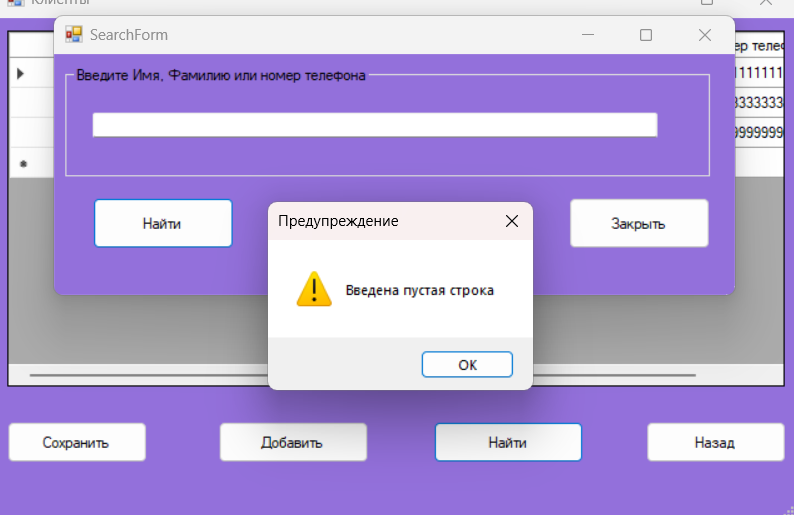


Рисунок 6 – Тестирования поиска пустого значения

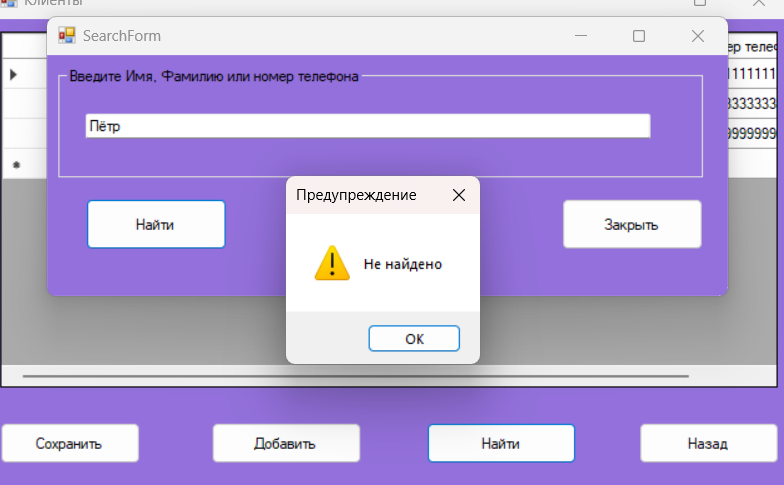


Рисунок 7 – Тестирование поиска некорректных значений

Таблица 3 - Тест-кейсы для формы добавления записи в таблицу Заказы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Входные данные | Цель | Описание | Ожидаемый результат | Результат |
| 1 | Ввод корректных данных:  3  2  Укорачивание брюк  500  11.12.2024  11.12.2024  (Рисунок 25) | Тестирование ввода данных | Тестирование проверки корректности данных и вводу их в таблицу | Создание новой строки в таблице Заказы | Создание новой строки в таблице Заказы |
| 2 | Ничего не вводить  (Рисунок 26) | Тестирование ввода данных | Тестирование проверки корректности данных и вводу их в таблицу | Вывод сообщения об ошибке | Вывод сообщения об об ошибке |
| 3 | Ввод некорректных данных  12  6  Ушить  10000  13.12.2025  13.12.2024  (Рисунок 27) | Тестирование ввода данных | Тестирование проверки корректности данных и вводу их в таблицу | Вывод сообщения об ошибке | Вывод сообщения об об ошибке |

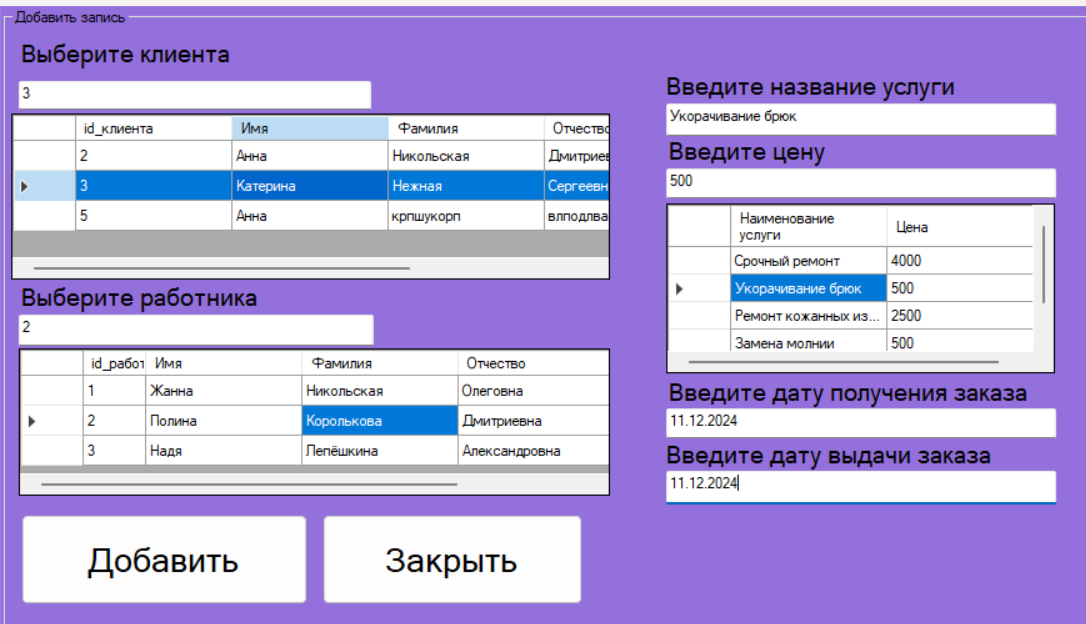


Рисунок 8 - Тестирование добавления корректных значений

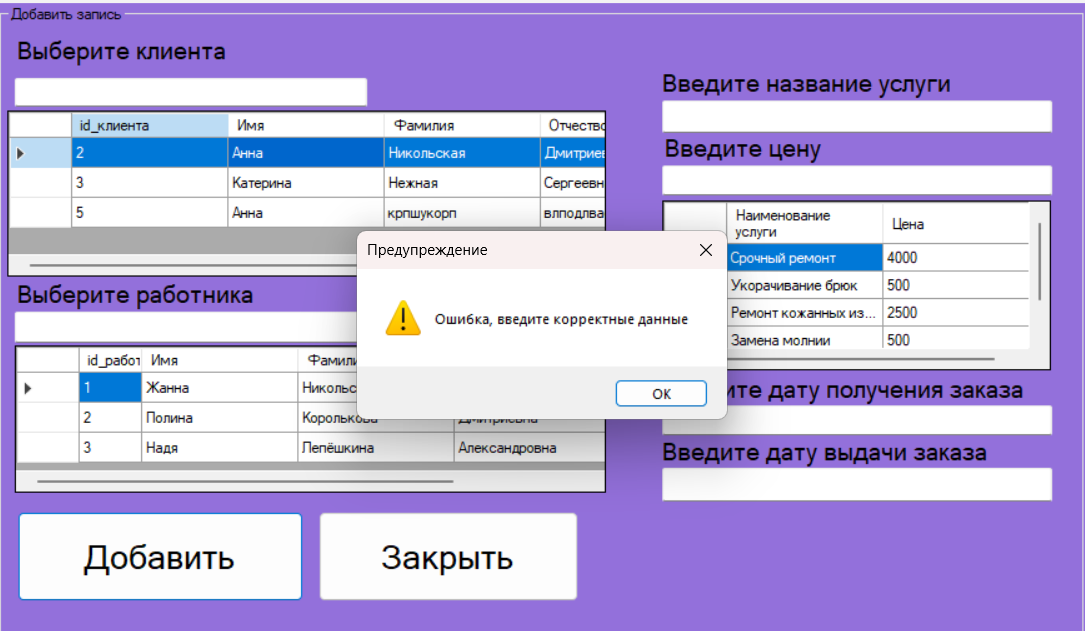


Рисунок 9 – Тестирования добавления пустых значений

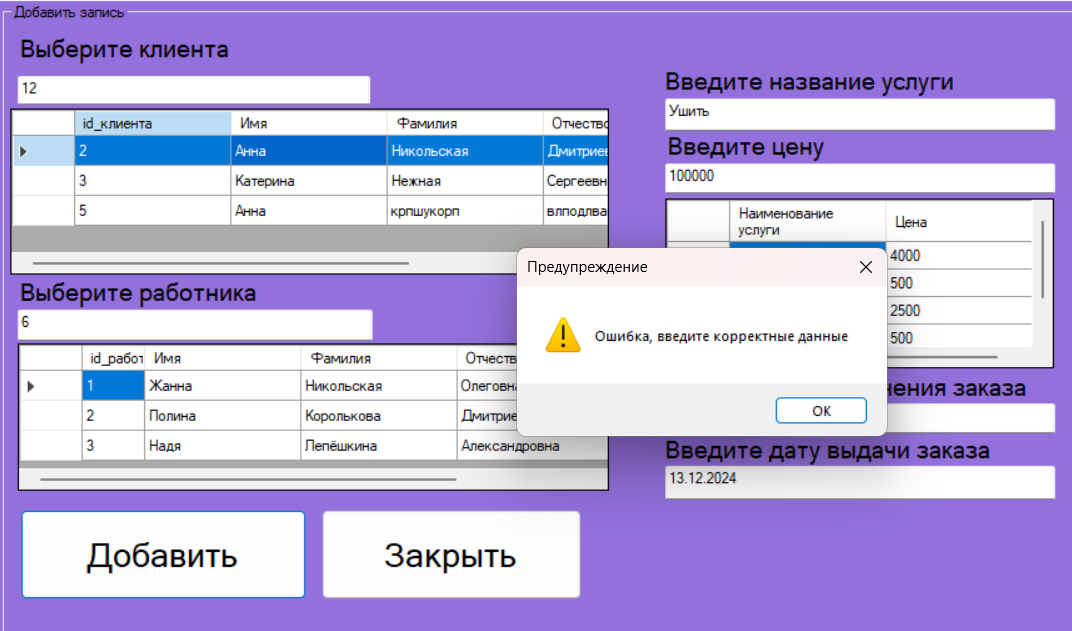


Рисунок 10 - Тестирование добавления некорректных значений

Таблица 4 - Тест-кейсы для формы поиска записей в таблице Заказы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Входные данные | Цель | Описание | Ожидаемый результат | Результат |
| 1 | Ввод корректных данных:  1  (Рисунок 28) | Тестирование поиска | Тестирование нахождения данных соответствующих поиску | Выделение найденной строки | Выделение найденной строки |
| 2 | Ничего не вводить  (Рисунок 29) | Тестирование поиска | Тестирование нахождения данных соответствующих поиску | Вывод сообщения об ошибке | Вывод сообщения об ошибке |
| 3 | Ввод некорректных данных (Отсутствуют в таблице)  9  (Рисунок 30) | Тестирование поиска | Тестирование нахождения данных соответствующих поиску | Вывод сообщения об ошибке | Вывод сообщения об ошибке |

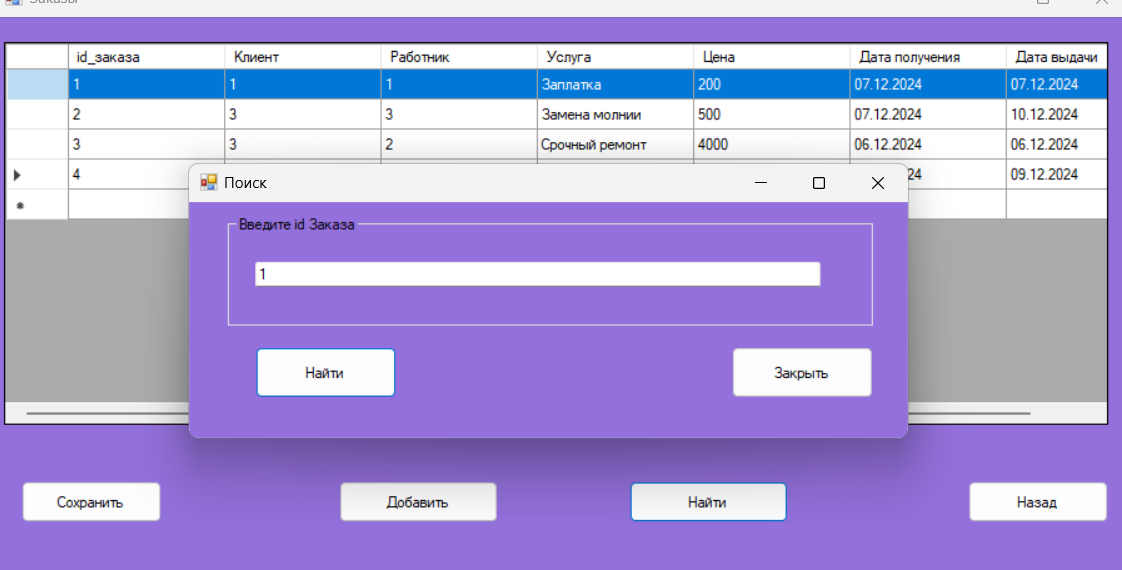


Рисунок 11 - Тестирование добавления корректных значений

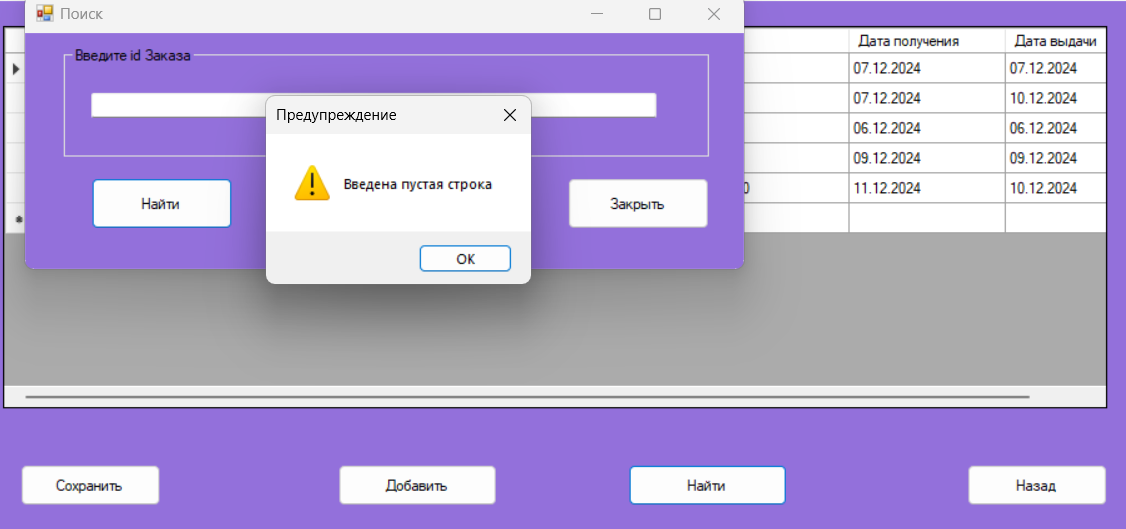


Рисунок 12 – Тестирования поиска пустого значения

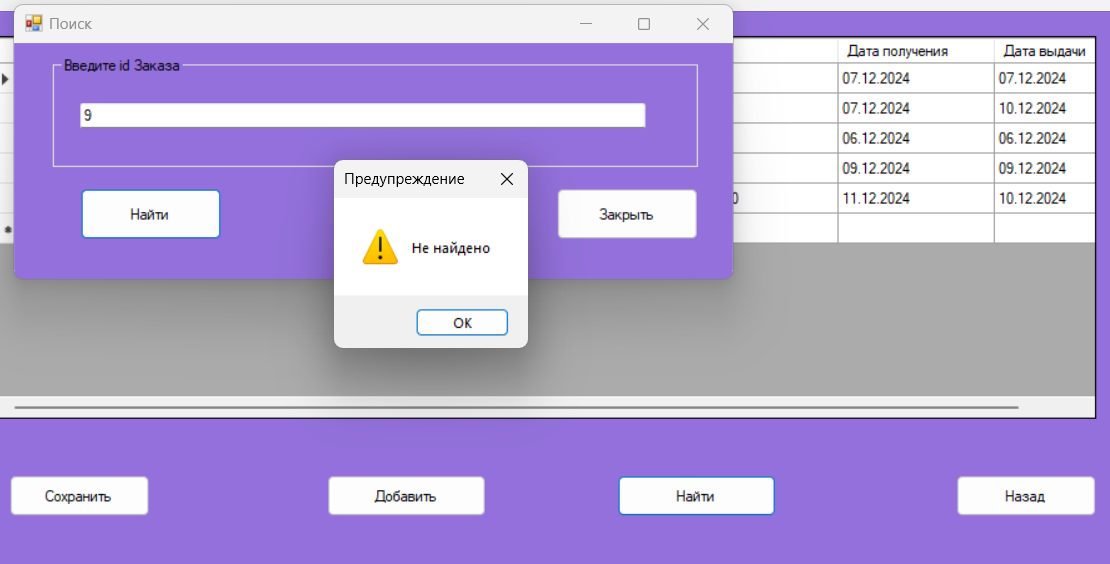
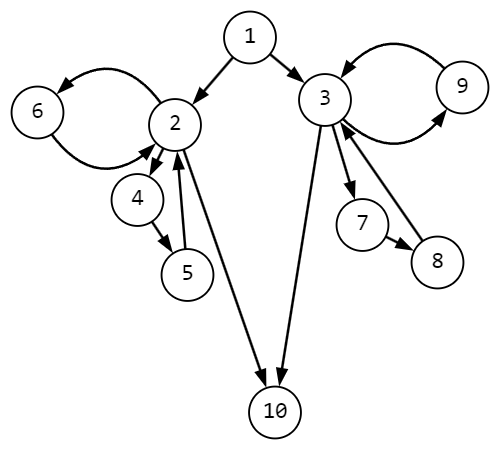


Рисунок 13 – Тестирование поиска некорректных значений

Интеграционное тестирование — это фаза тестирования программного обеспечения, которая проверяет взаимодействие между различными модулями или компонентами системы. В отличие от модульного тестирования, которое фокусируется на отдельных модулях, интеграционное тестирование проверяет, как эти модули работают вместе, как они обмениваются данными и как они взаимодействуют с другими частями системы.

Чтобы последовательно и безошибочно тестировать программу, необходимо составить тестовый граф. Тестовый граф используется в тестировании программного обеспечения, особенно в методах структурного тестирования, для обеспечения полного покрытия кода тестами. Он представляет собой визуальное представление логической структуры программы, отображающее пути выполнения и условия перехода между ними. Цель использования тестового графа — гарантировать, что каждый путь выполнения программы будет протестирован, по крайней мере, один раз, представлен на рисунке 18.



Пояснения по графу:

1. Запуск
2. Выбор пункта меню Клиенты
3. Выбор пункта меню Заказы
4. Добавление записи в таблицу Клиенты
5. Сохранение изменений в таблице Клиенты
6. Поиск в таблице Клиенты
7. Добавление записи в таблицу Заказы
8. Сохранение изменений в таблице Заказы
9. Поиск в таблице Заказы
10. Закрытие приложения

Рисунок 14 – Тестовый граф программы

Тестовые пути:

Т1: 1-2-4-5-2-10

Т2: 1-2-6-2-10

Т3: 1-2-4-5-2-6-2-10

Т4: 1-3-7-8-3-10

Т5: 1-3-9-3-10

Т6: 1-3-7-8-3-9-3-10

Таблица интеграционное тестирование

|  |  |  |
| --- | --- | --- |
| Идентификатор тестового пути | Цель теста | Ожидаемый результат |
| Т1 | Проверить связь между начальной формой, формой Клиенты, методами добавления | Успешный переход с начальной формы на форму Клиенты, с формы на вспомогательную форму и успешное добавление данных |
| Т2 | Проверить связь между начальной формой, формой Клиенты, методом поиска | Успешный переход с начальной формы на форму Клиенты, с формы на вспомогательную форму и успешное нахождение данных |
| Т3 | Проверить связь между начальной формой, формой Клиенты, методами добавления и поиска | Успешный переход с начальной формы на форму Клиенты, с формы на вспомогательную форму и успешное добавление и поиск данных |
| Т4 | Проверить связь между начальной формой, формой Заказы, методами добавления | Успешный переход с начальной формы на форму Заказы, с формы на вспомогательную форму и успешное добавление данных |
| Т5 | Проверить связь между начальной формой, формой Заказы, методом поиска | Успешный переход с начальной формы на форму Заказы, с формы на вспомогательную форму и успешное добавление и поиск данных |
| Т6 | Проверить связь между начальной формой, формой Заказы, методами добавления и поиска | Успешный переход с начальной формы на форму Заказы, с формы на вспомогательную форму и успешное добавление и поиск данных |

Системное тестирование — это тип тестирования программного обеспечения, который проверяет систему в целом, как единое целое. Он проводится после интеграционного тестирования и направлен на проверку того, что система соответствует всем требованиям, функционирует как ожидается и отвечает на запросы пользователя.

В отличие от модульного или интеграционного тестирования, которые фокусируются на отдельных компонентах или их взаимодействии, системное тестирование рассматривает систему как законченный продукт. Тестирование проводится, как правило, в среде, максимально приближенной к реальной рабочей среде.

Минимальные требования:

* процессор с тактовой частотой не менее 1 ГГц;
* 32/64-битная версия Microsoft Windows 11;
* оперативная память объемом не менее 2ГБ;
* 3 Мб свободного места на диске.

Программа тестируется на системе со следующими характеристиками:

* Процессор AMD Ryzen 5 5560U with Radeon Graphics 2.30 GHz;
* Оперативная память 16,0 ГБ (доступно: 15,4 ГБ);
* Тип системы 64-разрядная операционная система, процессор x64;
* SSD 475 ГБ;

С помощью Диспетчера задач можно посмотреть загрузку системы приложения. Они выглядят следующим образом:

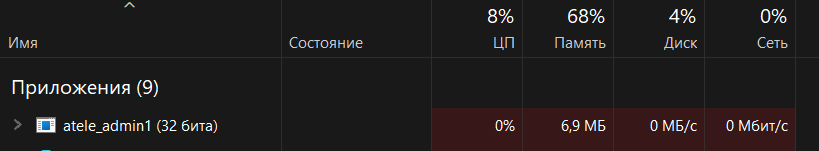


Рисунок 15 – Диспетчер задач

Как видно из рисунка 29, максимальное потребление ресурсов составило 6,9 МБ.