# **Тестирование разработанного продукта**

## **Анализ и выбор методов тестирования**

Существует достаточно много разновидностей тестирования: системное, регрессионное, функциональное, стресс-тестирование, тестирование производительности, нагрузочное тестирование, тестирование безопасности, тестирование локализации, юзабилити тестирование. Тестирование программного обеспечения (ПО) является неотъемлемой частью цикла разработки ПО.

Целью тестирования являлось выявление ошибок в программном обеспечении. Тестирование позволяет выявить следующие групповые ошибки:

* нестабильность работы модулей системы;
* некорректное изменение данных в базе данных;
* логические ошибки, связанные с нарушением алгоритмов в коде;
* большую нагрузку на СУБД;
* нарушение безопасности приложения.

Выбор был сделан в пользу таких стилей тестирования:

* Тестирование белого ящика. Основным методом тестирования Белого ящика является анализ покрытия кода. Анализ покрытия кода устраняет пробелы в наборе тестовых примеров. Он определяет области программы, которые не выполняются набором тестовых случаев. После выявления пробелов вы создаете контрольные примеры для проверки непроверенных частей кода, тем самым повышая качество программного продукта.
* Тестирование черного ящика. При тестировании методом черного ящика основное внимание уделяется проверке ваших функциональных требований. Тестирование предполагает проверку работы функций системы, а нефункциональное – общие характеристики программы.

## **Результаты тестирования**

После окончания разработки программного обеспечения, было проведено тестирование. В ходе тестирования, была проверена работоспособность программы и её отказоустойчивость.

**Тестирование белого ящика**

Для того чтобы протестировать программу на работоспособность был создан проект модульного тестирования внутри приложения.

Тест был проведен для проверки наличия конкретного материала в базе данных. Будут представлены как Позитивные тесты, подтверждающие наличие данного материала, так и Негативные тесты, свидетельствующие об отсутствии информации о нем.

В базе данных содержатся материалы, представленные на рисунке 1.

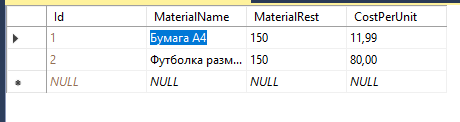


Рисунок 1 - Таблица «Материалы»

Негативные тесты: – тесты для проверки устойчивости ПО к негативным входным данным. На рисунке 2 представлен код, в котором ниже в поля проверки были вставлены несуществующие данные. Программа показала, что таких данных на материал не существует.

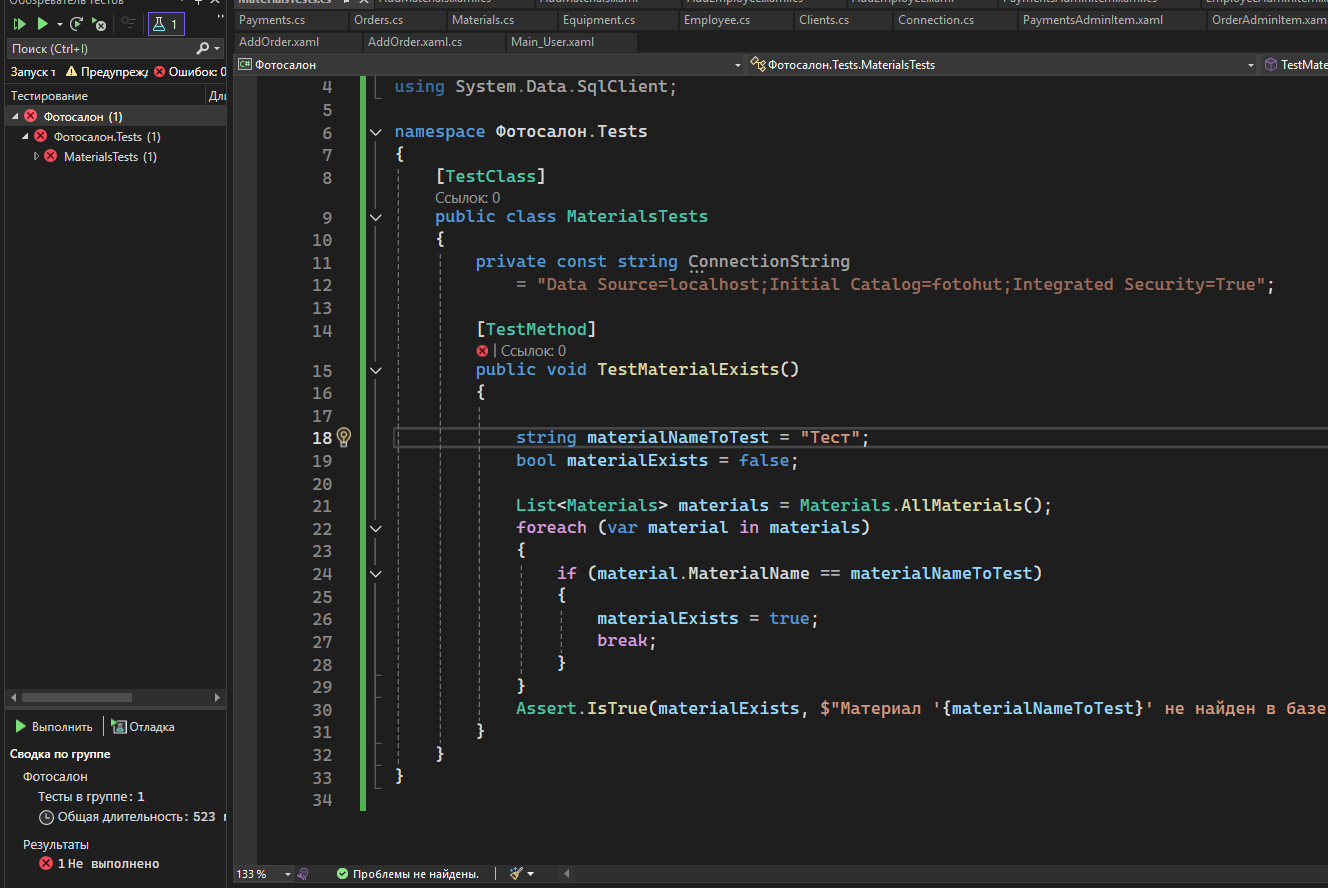


Рисунок 2 - Отрицательный результат теста

Позитивные тесты - тесты, предназначенные для проверки, что программа выполняет свое основное предназначение. На рисунке 51 представлен код, в котором ниже в поля проверки были вставлены существующие данные. Программа показала, что такие данные существует и происходит переход к форме выбора.

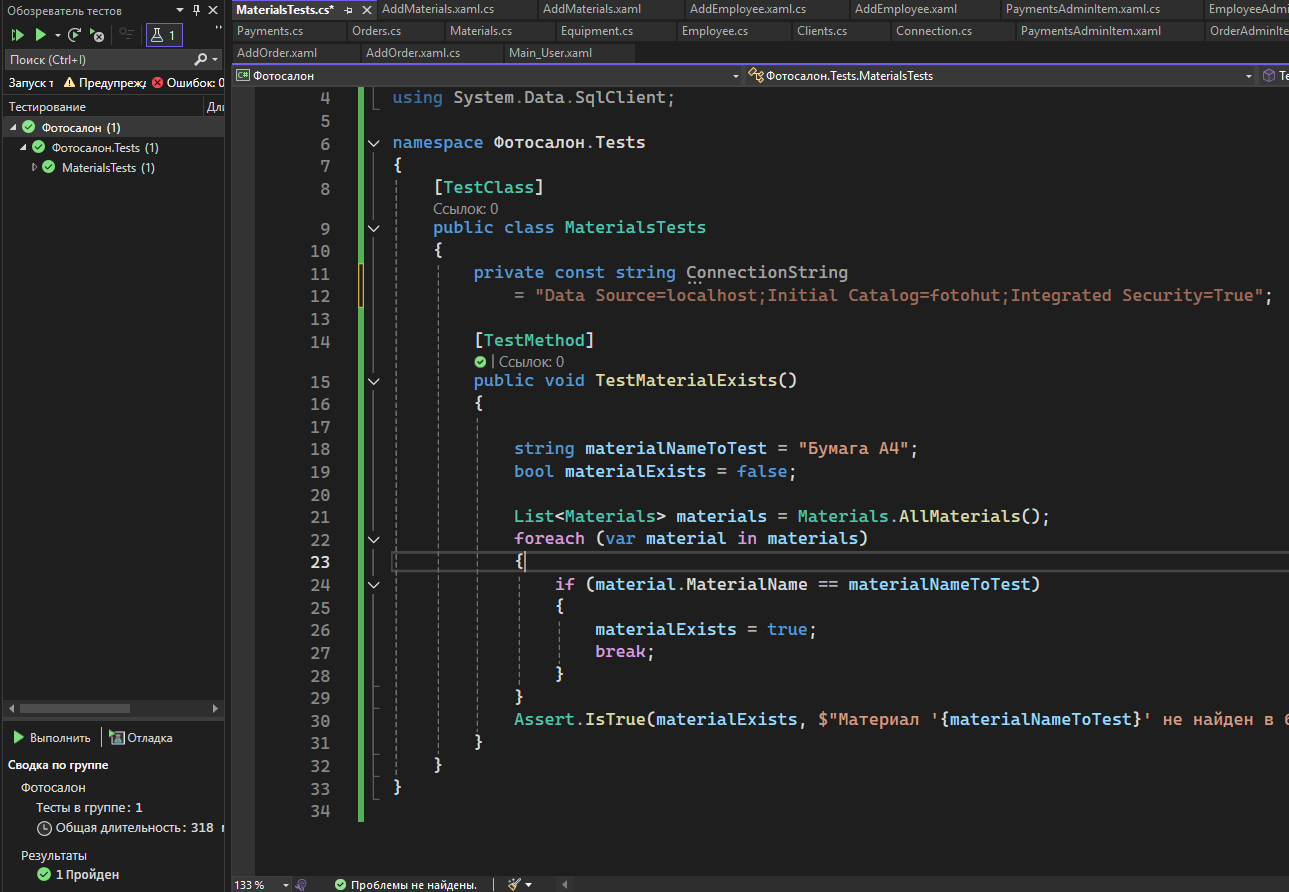


Рисунок 3 - Положительный результат теста

**Метод черного ящика – проверка работы функции регистрации**

В базе данных содержатся клиенты, представленные на рисунке 4.

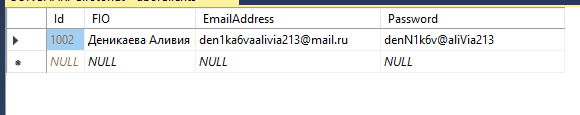


Рисунок 4 – Таблица «Клиенты»

В таблице 1 представлен позитивный результат теста. Выполнен вход за пользователя.

Таблица 1 - Позитивный результат теста

|  |  |
| --- | --- |
| Тестовый пример # | TC\_AA\_01 |
| Приоритет тестирования | Высокий |
| Заголовок/название теста | Авторизация\_01 |
| Краткое изложение теста | Успешное определение логина и пароля из базы данных |
| Этапы теста | 1.Ввод данных  2.Получение результатов |
| Тестовые данные | Адрес эл. почты: den1ka6vaalivia213@mail.ru  Пароль: denN1k6v@aliVia213 |
| Ожидаемый результат | Чтение из полей программы и проверка присутствие введенных данных в базе данных, если данные введены верно, то переход на следующий модуль пользователя |
| Фактический результат | Выполнение входа и вывод сообщения «Вы успешно вошли как пользователь Деникаева Аливия» |
| Статус | Зачет |
| Предварительное условие | Определение адреса эл. почты и пароля закрепленных за пользователем |
| Постусловие | Система ожидает действий со стороны пользователя |
| Примечания/комментарии | Тест пройден успешно |

В таблице 2 представлен негативный результат теста. Вход не выполнен

Таблица 2 - Негативный результат теста

|  |  |
| --- | --- |
| Тестовый пример # | TC\_AA\_02 |
| Приоритет тестирования | Высокий |
| Заголовок/название теста | Авторизация\_02 |
| Краткое изложение теста | Успешное определение логина и пароля из базы данных |
| Этапы теста | 1.Ввод данных  2.Получение результатов |
| Тестовые данные | Логин: vsadsajn21o33@mail.ru  Пароль: Anatolyvsadsajn21o33 |
| Ожидаемый результат | Чтение из полей программы и проверка присутствие введенных данных в базе данных, если данные введены верно, то переход на следующий модуль пользователя |
| Фактический результат | Таких данных на такого пользователя в базе не имеется |
| Статус | Зачет |
| Предварительное условие | Определение адреса эл. почты и пароля закрепленных за пользователем |
| Постусловие | Система ожидает действий со стороны пользователя |
| Примечания/комментарии | Тест пройден успешно |

В таблице 3 представлен позитивный результат теста. Выполнен вход за администратора.

Таблица 3 - Позитивный результат теста

|  |  |
| --- | --- |
| Тестовый пример # | TC\_AA\_03 |
| Приоритет тестирования | Высокий |
| Заголовок/название теста | Авторизация\_03 |
| Краткое изложение теста | Успешное определение логина и пароля из базы данных |
| Этапы теста | 1.Ввод данных  2.Получение результатов |
| Тестовые данные | Логин: admin;  Пароль: admin; |
| Ожидаемый результат | Чтение из полей программы и проверка присутствие введенных данных в условии программы, если данные введены верно, то переход на следующий модуль  администратора |
| Фактический результат | Вход за администратора, для редактирования таблиц выполнен |
| Статус | Зачет |
| Предварительное условие | Определение логина и пароля закрепленных за администратором |
| Постусловие | Система ожидает действий со стороны администратора |
| Примечания/комментарии | Тест пройден успешно |

Подводя итог всего описанного выше хочется отметить, что тестирование делает код стабильным и предсказуемым. Поэтому код, покрытый тестами, гораздо проще масштабировать и поддерживать.