ЛАБОРАТОРНАЯ РАБОТА №2

“ТЕСТИРОВАНИЕ ТРЕБОВАНИЙ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ”

**Цель работы:** получения навыков создания тест-кейсов для приложения без спецификации.

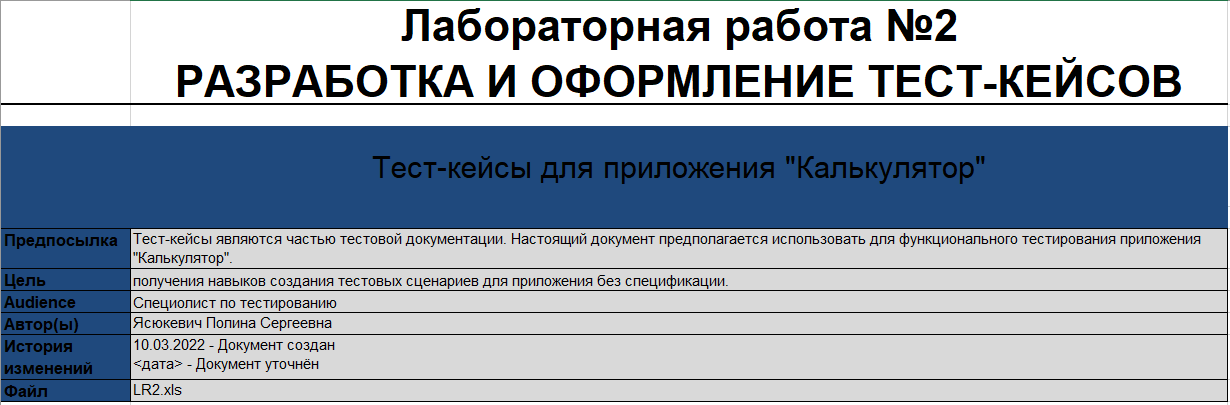


Рисунок 1 – Тест-кейс для приложения “Калькулятор”

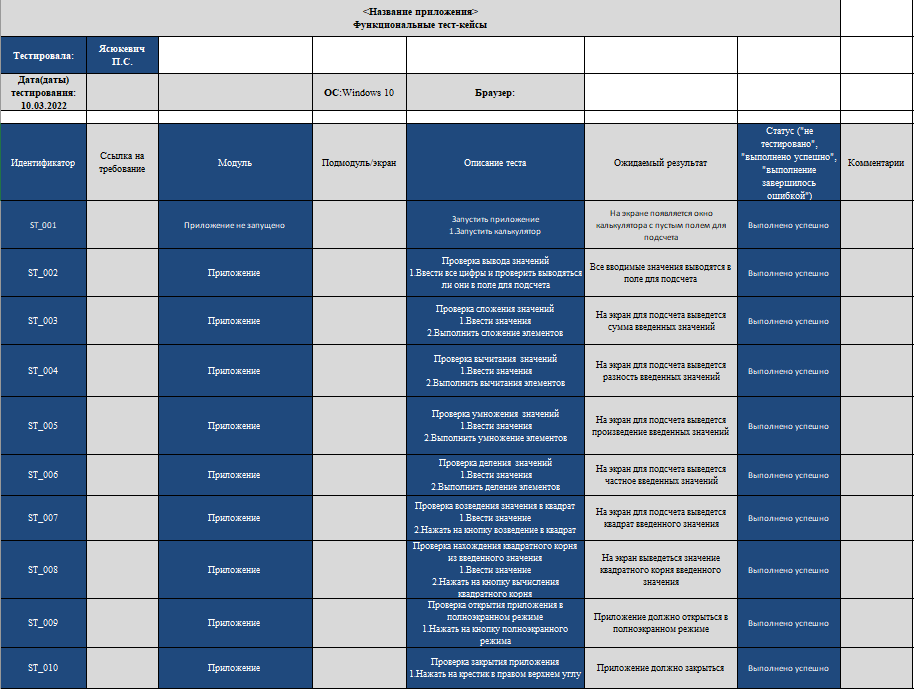


Рисунок 2 – Smoke test

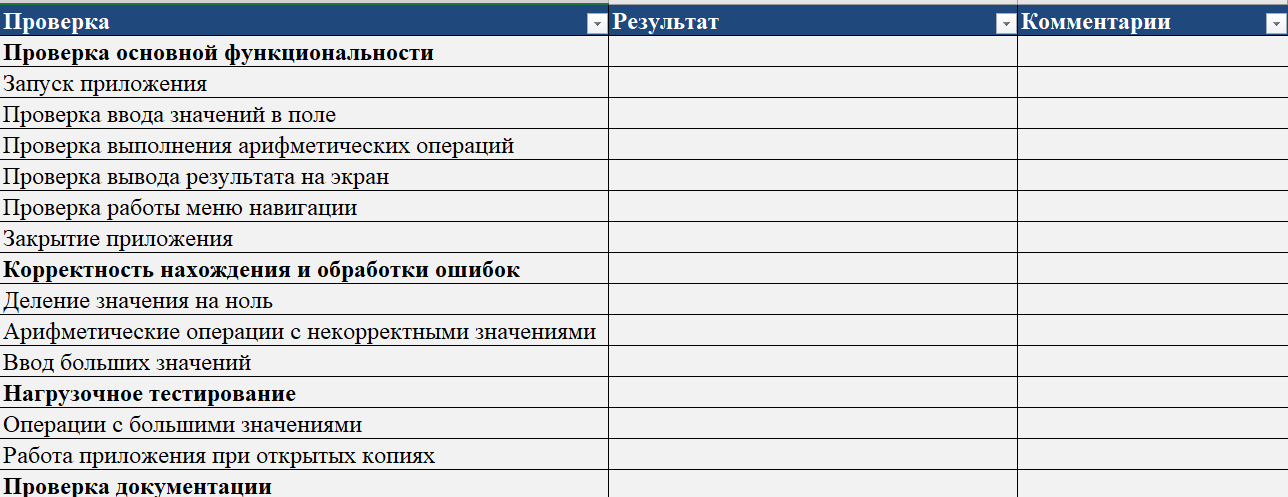


Рисунок 3 – Чек-лист для Critical Path test

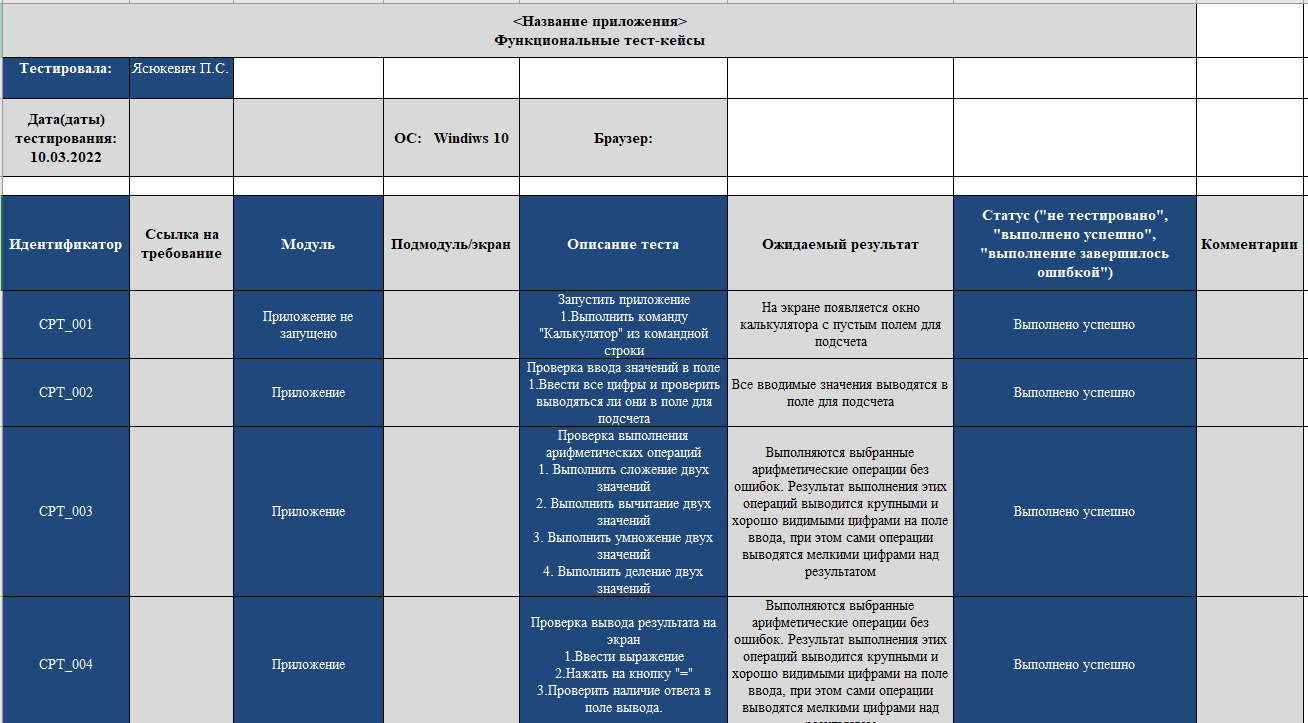


Рисунок 4.1 –Critical Path test

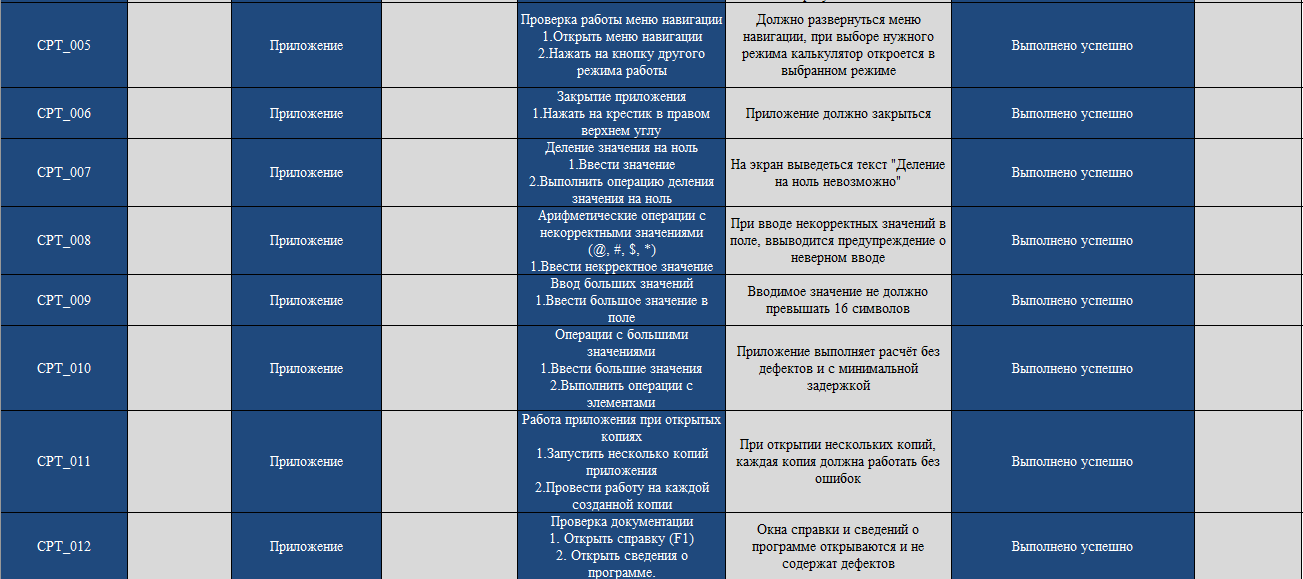


Рисунок 4.2 –Critical Path test

**Контрольные вопросы:**

**1.Какие разновидности тестов вы запомнили? (Можно вспоминать на примере тестирования ручки.)**

Beta Testing, Validation, Verification, Security Testing, Negative Testing, UI Testing, UI Testing Functional, Testing, Functional Testing Usability Testing, Compatibility Testing, Performance Testing

**2.Что такое классы эквивалентности?**

Класс эквивалентности – набор данных, обработка которых приводит к одному и тому же результату.

**3.Зачем нужны классы эквивалентности?**

Целью является не только сокращение числа тестов, но и сохранение приемлемого тестового покрытия.

**4.Какие вы помните признаки эквивалентности?**

1. Они направлены на поиск одной и той же ошибки
2. Если один из тестов обнаруживает ошибку, другие её тоже, скорее всего,

обнаружат

1. Если один из тестов НЕ обнаруживает ошибку, другие её тоже,

скорее всего, НЕ обнаружат

1. Тесты используют одни и те же наборы входных данных
2. Для выполнения тестов мы совершаем одни и те же операции
3. Тесты генерируют одинаковые выходные данные или приводят

приложение в одно и то же состояние (например открытие валидных

файлов одного типа и схожего объема, но с разнымсодержанием)

1. Все тесты приводят к срабатыванию одного и того же блока

обработки ошибок («error handling block»)

1. Ни один из тестов не приводит к срабатыванию определенного

блока обработки ошибок («error handling block») (например, блок

обработки ошибок открытия файла)

**5.Что такое граничные условия (в контексте классов эквивалентности)? Приведите несколько примеров.**

Это те места, в которых один класс эквивалентности переходит в другой. Например, одна группа тестов вызывает сообщение «вы ввели слишком маленькое число», а другая вызывает сообщение «вы ввели слишком большое число». Граница будет лежать где-то в районе чисел «самых больших из слишком маленьких» и «самых маленьких из слишком больших».

! Граничные условия очень важны, и их обязательно следует проверять в тестах, т.к. именно в этом месте чаще всего и обнаруживаются ошибки

**6.Какие рекомендации по разработке тестов вы запомнили?**

Начинайте с простых очевидных тестов. Затем переходите к более сложным тестам.

Помните о граничных условиях. Если остаётся время, занимайтесь исследовательским тестированием.

**7.Дайте определение тест–кейса.**

Тест-кейс (test-case) — набор входных данных, условий выполнения и ожидаемых результатов, разработанный с целью проверки того или иного свойства или поведения программного средства.

**8.Какую информацию о тест–кейсе необходимо записать?**

Любой тест-кейс обязательно включает в себя:

1. Индентификатор
2. Приоритет
3. Связанное с ним требование
4. Модуль и подмодуль
5. Заглавие
6. Исходные данные необходимые для выполнения
7. Шаги
8. Ожидаемые результаты

**9.Зачем нужны тест–кейсы?**

Тест-кейсы должен помочь нам провести проверку продукта без ознакомления со всей документацией. Написанный один раз, удобный в поддержке тест-кейс сэкономит много времени и сил тестировщикам.

**10.Какие свойства тест–кейсов вы знаете?**

1. Специфичными или общими.
2. Простыми или сложными.
3. Независимыми или связанными друг с другом.
4. Позитивными или негативными.

**11.Что такое позитивные и негативные тесты?**

Позитивное тестирование – это тестирование с применением сценариев, которые соответствуют нормальному (штатному, ожидаемому) поведению системы. С его помощью мы можем определить, что система делает то, для чего и была создана. Например, умножение на калькуляторе цифр 3 и 5.

Негативным называют тестирование, в рамках которого применяются сценарии, которые соответствуют внештатному поведению тестируемой системы. Это могут быть, например, исключительные ситуации или неверные данные. На примере калькулятора, это умножения числа 3 на грушу. Значение “груша” не является валидным для калькулятора.

**12.Каковы критерии хорошего тест–кейса?**

1. Обладает высокой вероятностью обнаружения

ошибки.

1. Не выполняет ненужных действий.
2. Не является избыточным по отношению к другим

тестам.

1. Исследует соответствующую («ту, которую надо»)

область приложения.

1. Позволяет легко диагностировать ошибку.
2. Делает обнаруженную ошибку очевидной.
3. Независим (каждый тест-кейс – это индивидуальный

сценарий с точкой входа и точкой выхода).

**13.Что такое тестовый сценарий?**

Тестовый сценарий – документ, описывает последовательность действий по выполнению теста.

**14.Какие вы знаете техники ускорения написания тестов?**

Copy-paste:

Если по ходу разработки тестов возникают вопросы, нужно писать их прямо в документ с тестами, помечая красным цветом. Используя т.н. «косметику» (жирный, подчёркнутый, наклонный шрифт, разные цвета т.д.). Это значительно повышает читаемость документа. Если пишутся тесты в файле, обязательно прописывайте в самом файле историю его изменения.

**15.Назовите основные шаги разработки тестов.**

1. Сбор информации
2. Разделение приложения на модули
3. Написание чек-листов
4. Написание тест-кейсов

**Вывод:** при выполнении задания для лабораторной работы получила практические навыки по разработке чек-листа и созданию тест-кейсов для приложения без спецификации.