**Отчет по тестированию «С# Calculator»**

**Цель:**

Провести функциональное и нефункциональное тестирование приложения "Калькулятор".

**Описание приложения(требования к проекту):**

Приложение "С# Calculator" должно поддерживать базовые арифметические операции (сложение, вычитание, умножение, деление) и работу с дробными числами. Интерфейс приложения включает кнопки для цифр, основных арифметических операций, кнопки "C" (очистить), "CE" (удалить последние введенные данные), и "=" (вычислить результат).

**Тестировал:** Маслик Антон Дмитриевич (кандидат на позицию AQA-инженер(junior))

**Задачи:**

1. Функциональное тестирование
2. Нефункциональное тестирование
3. Результат тестирования проекта

**Функциональное тестирование**

**Сложение чисел**

* Тест 1:
  + Описание проблемы: Проверка сложения целых чисел
  + Шаги для воспроизведения:

1. Введите: “2 + 3”
2. Нажмите на кнопку “=”

* Ожидаемый результат: Результат равен 5
* Фактический результат: Результат равен 5 ([Link on screenshot](https://i.imgur.com/u0fjU3L.png))
* Тест 2:
  + Описание проблемы: Проверка сложения дробных чисел
  + Шаги для воспроизведения:

1. Введите: “1.5 + 2.5”
2. Нажмите на кнопку “=”

* Ожидаемый результат: Результат равен 4
* Фактический результат: Результат равен 4 ([Link on screenshot](https://i.imgur.com/xYj4CaF.png))

**Вычитание чисел**

* Тест 1:
  + Описание проблемы: Проверка вычитания целых чисел
  + Шаги для воспроизведения:

1. Введите: “5 - 2”
2. Нажмите на кнопку “=”

* Ожидаемый результат: Результат равен 3
* Фактический результат: Результат равен 3 ([Link on screenshot](https://i.imgur.com/vMUEB7I.png))
* Тест 2:
  + Описание проблемы: Проверка вычитания дробных чисел
  + Шаги для воспроизведения:

1. Введите: “5.5 - 2.5”
2. Нажмите на кнопку “=”

* Ожидаемый результат: Результат равен 3
* Фактический результат: Результат равен 3 ([Link to screenshot](https://i.imgur.com/rtPxNCf.png))

**Умножение чисел**

* Тест 1:
  + Описание проблемы: Проверка умножения целых чисел
  + Шаги для воспроизведения:

1. Введите: “3 \* 4”
2. Нажмите на кнопку “=”

* Ожидаемый результат: Результат равен 12
* Фактический результат: Результат равен 12 ([Link to screenshot](https://i.imgur.com/zyfhrYc.png))
* Тест 2:
  + Описание проблемы: Проверка вычитания целых чисел
  + Шаги для воспроизведения:

1. Введите: “2.5 \* 2”
2. Нажмите на кнопку “=”

* Ожидаемый результат: Результат равен 5
* Фактический результат: Результат равен 5 ([Link to screenshot](https://i.imgur.com/rDqn24i.png))

**Деление**

* Тест 1:
  + Описание проблемы: Проверка деления целых чисел
  + Шаги для воспроизведения:

1. Введите: “8 : 2”
2. Нажмите на кнопку “=”

* Ожидаемый результат: Результат равен 4
* Фактический результат: Результат равен 4 ([Link on screenshot](https://i.imgur.com/ZnLNa1I.png))
* Тест 2:
  + Описание проблемы: Проверка деления дробных чисел
  + Шаги для воспроизведения:

1. Введите: “7.5 / 2.5”
2. Нажмите на кнопку “=”

* Ожидаемый результат: Результат равен 3
* Фактический результат: Результат равен 3 ([Link to screenshot](https://i.imgur.com/fiBcpsg.png))
* Тест 3:
  + Описание проблемы: Проверка поведения при делении на 0
  + Шаги для воспроизведения:

1. Введите: “8 : 0”
2. Нажмите на кнопку “=”

* Ожидаемый результат: Исключение или сообщение об исключении(ошибке)
* Фактический результат: Исключение DevisionByZero ([Link on screenshot](https://i.imgur.com/BKTNqxA.png))

**Комбинированные операции**

* Тест 1:
  + Описание проблемы: Проверка возможности комбинирования
  + Шаги для воспроизведения:

1. Введите: “2 + 3 \* 4”
2. Нажмите на кнопку “=”

* Ожидаемый результат: Результат равен 14 и соблюдается приоритет операций
* Фактический результат: Результат равен 20, не соблюдается приоритет операций ([Link to screenshot](https://i.imgur.com/i0cvrqO.png))

**Очистка**

* Тест 1:
  + Описание проблемы: Проверка функции очистки
  + Шаги для воспроизведения:

1. Введите: “2 + 3”
2. Нажмите “C”

* Ожидаемый результат: Данные очищены
* Фактический результат: Данные очищаются, как ожидалось
* Тест 2:
  + Описание проблемы: Проверка функции очистки истории
  + Шаги для воспроизведения:

1. Введите: “2 + 3”
2. Нажмите “CE”

* Ожидаемый результат: Очищены только последние данные
* Фактический результат: Очищаются только последние данные, как ожидалось

**UI/UX тестирование**

* Тест 1:
  + Описание проблемы: Проверка отображения кнопок и их работы
  + Шаги для воспроизведения:

1. Нажимать все кнопки
2. Просматривать, что они печатаются
3. Производить рандомные операции вычисления

* Ожидаемый результат: Кнопки отображаются и корректно работают
* Фактический результат: Кнопки отображаются и корректно работают
* Тест 2:
  + Описание проблемы: Проверка анимации отклика на кнопках
  + Шаги для воспроизведения:

1. Нажимать все кнопки
2. Просматривать, чтобы анимация начиналась
3. Просматривать, чтобы анимация заканчивалась после отклика

* Ожидаемый результат: Кнопки отображаются и корректно работают
* Фактический результат: Анимация работает, после отклика пропадает

**Тестирование производительности**

* Тест 1:
  + Описание проблемы: Проверка корректной производительности
  + Шаги для воспроизведения:

1. Ввести “9999999999 \* 9999999999”
2. Отследить время отклика

* Ожидаемый результат: Время отклика корректное
* Фактический результат: Время отклика корректное

**Граничные значения**

* Тест 1:
  + Описание проблемы: Проверка корректности при минимальном значении
  + Шаги для воспроизведения:

1. Ввести “-1.79769313486232E+308 + 28”
2. Нажать “=”

* Ожидаемый результат: Корректное значение для double типа
* Фактический результат: Корректное значение для double типа
* Тест 2:
  + Описание проблемы: Проверка корректности при максимальном значении
  + Шаги для воспроизведения:

1. Ввести “1.79769313486232E+308”
2. Нажать “=”

* Ожидаемый результат: Корректное значение для double типа
* Фактический результат: Корректное значение для double типа

**Тестирование устойчивости**

* Тест 1:
  + Описание проблемы: Проверка корректной работы при непрерывном вводе значения
  + Шаги для воспроизведения:

1. Непрерывно вводить значения
2. Отслеживать корректность работы

* Ожидаемый результат: Устойчивость корректная
* Фактический результат: Устойчивость корректная

**Тестирование на разных устройствах**

* Тест 1:
  + Описание проблемы: Проверка поведения при изменения размера окна
  + Шаги для воспроизведения:

1. Изменять разрешение окна
2. Изменять разрешение экрана
3. Отслеживать поведение GUI

* Ожидаемый результат: Поведение GUI соответствует нормам
* Фактический результат: Поведение GUI соответствует нормам

**Выявленные баги**

**Комбинированные операции (Тест №1)**

* Ожидаемый результат: Результат равен 14 и соблюдается приоритет операций
* Фактический результат: Результат равен 20, не соблюдается приоритет операций

**Предложения по решению проблемы:**

1. Проблема с приоритетом операций: В текущей реализации не учитывается приоритет операций (умножение и деление должны выполняться перед сложением и вычитанием).
2. Изменение метода Calculate:
3. Разделить операции на два этапа: сначала обработать умножение и деление, затем сложение и вычитание.
4. Создать временный список для хранения результатов после первого этапа.

**Реализация:**

Пройтись по prevValues и выполнить все умножения и деления, сохраняя результаты во временном списке.

Затем пройтись по временному списку и выполнить сложения и вычитания.

**Тестирование:**

После внесения изменений протестировать калькулятор с выражением, ожидающим результат 14, чтобы убедиться, что приоритет операций соблюдается.

**Оптимизация:**

Убедиться, что код остается читаемым и понятным после внесенных изменений.