**Проект «калькулятор»**

**Список требований к проекту калькулятора**

▎1. **Функциональные требования**

• Основные арифметические операции: Калькулятор должен поддерживать сложение, вычитание, умножение и деление.

• Дополнительные функции: Возможность выполнять операции с процентами, квадратные корни и возведение в степень.

• История операций: Пользователь должен иметь возможность просматривать историю предыдущих вычислений.

• Поддержка скобок: Возможность выполнения операций с использованием круглых скобок для изменения порядка вычислений.

• Копирование и вставка: Пользователь должен иметь возможность копировать результаты и вставлять их в другие приложения.

• Поддержка различных форматов ввода: Калькулятор должен принимать как десятичные, так и дробные числа.

▎2. **Нефункциональные требования**

• Интерфейс пользователя: Должен быть интуитивно понятным, простым и удобным для использования.

• Кроссбраузерная совместимость: Калькулятор должен работать во всех современных веб-браузерах (Chrome, Firefox, Safari и т.д.).

• Мобильная адаптивность: Интерфейс должен быть адаптирован для работы на мобильных устройствах.

• Производительность: Время отклика калькулятора должно быть минимальным, даже при сложных вычислениях.

• Безопасность: Защита данных пользователя, включая предотвращение возможных уязвимостей.

▎3. **Требования к тестированию**

• Юнит-тесты: Все функции калькулятора должны быть протестированы с использованием юнит-тестов.

• Тестирование интерфейса: Проверка на наличие ошибок в пользовательском интерфейсе и взаимодействии с ним.

• Тестирование производительности: Оценка скорости работы калькулятора под нагрузкой.

**Пользовательские истории**

1. **Как пользователь, я хочу выполнять основные арифметические операции**, чтобы быстро получать результаты расчетов.

Критерии приемки:

• Пользователь может ввести два числа и выбрать операцию (сложение, вычитание, умножение или деление).

• Результат отображается на экране.

2. **Как пользователь, я хочу использовать скобки в расчетах**, чтобы изменить порядок вычислений.

Критерии приемки:

• Пользователь может вводить выражения с использованием скобок.

• Калькулятор корректно обрабатывает порядок операций.

3. **Как пользователь, я хочу использовать калькулятор на мобильном устройстве**, чтобы иметь доступ к нему в любое время.

Критерии приемки:

• Интерфейс адаптирован для мобильных устройств.

• Все функции доступны на мобильной версии.

4. **Как пользователь, я хочу видеть сообщения об ошибках при некорректном вводе**, чтобы понимать, что пошло не так.

Критерии приемки:

• Если введено некорректное выражение, отображается понятное сообщение об ошибке.

5. **Как пользователь, я хочу выполнять дополнительные математические операции** (например, квадратный корень), чтобы расширить функциональность калькулятора.

Критерии приемки:

• Пользователь может выбрать дополнительные функции из меню и получить результат.

6. **Как пользователь, я хочу, чтобы** **интерфейс был простым и интуитивно понятным**, чтобы легко ориентироваться в функционале калькулятора.

Критерии приемки:

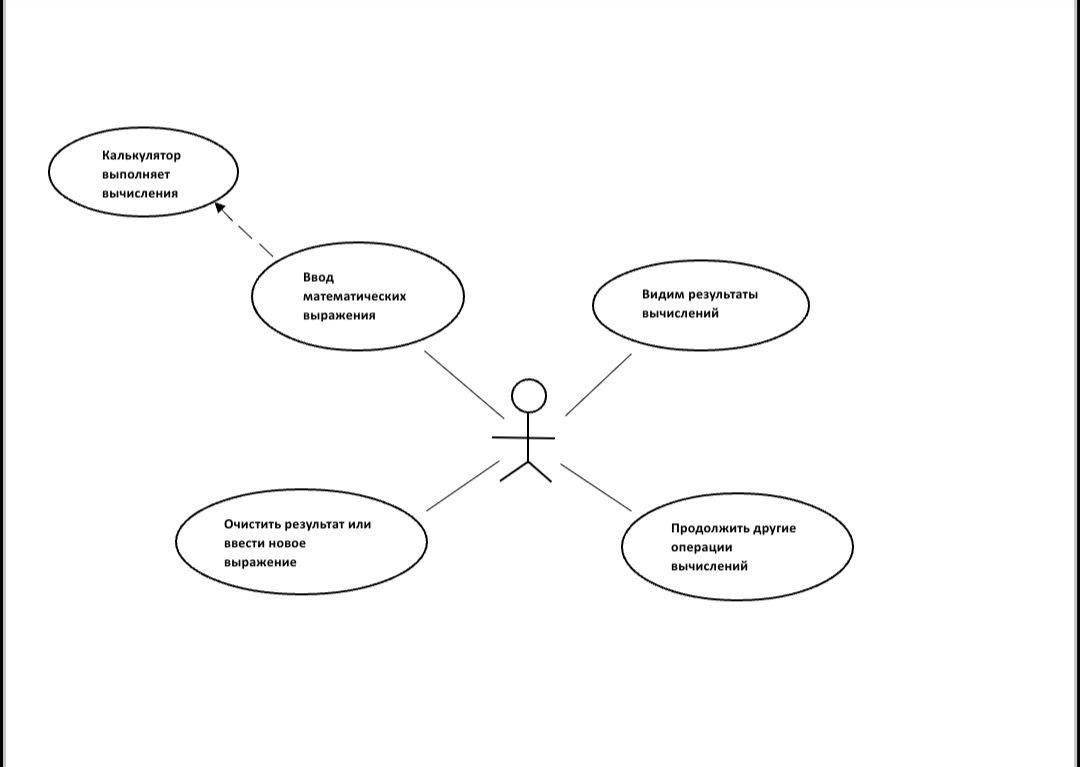
• Все кнопки и функции четко обозначены.

• Доступ к основным функциям не требует много времени.

**Заключение**

Сформированный список требований и пользовательские истории помогут четко определить цели проекта калькулятора и обеспечить его успешную реализацию. Каждый аспект должен быть тщательно проработан для создания удобного и функционального инструмента для пользователей.

**Диаграмма вариантов использования**



**Модули**

|  |  |
| --- | --- |
| ID | User Story |
| evaluateExpression | Как разработчик, я хочу обрабатывать выражение, чтобы его можно было вычислять. |
| applyOperation | Как разработчик, я хочу выполнять стандартные арифметические операции, чтобы вычислять их результат. |
| applyUnaryOperation | Как разработчик, я хочу выполнять унарные операции, чтобы вычислять результат тригонометрических выражений. |

**Итоги проекта по созданию калькулятора**

1. Цели и задачи:

• Определение, были ли достигнуты первоначальные цели проекта.

• Оценка выполнения поставленных задач, таких как разработка интерфейса, реализация функций и тестирование.

2. Функциональность:

• Список реализованных функций (основные операции, расширенные функции, графики и т.д.).

• Оценка пользовательского опыта: насколько интуитивно понятен интерфейс, легко ли выполнять расчеты.

3. Технические аспекты:

• Описание технологий и инструментов, использованных в проекте (язык программирования, библиотеки, фреймворки).

• Оценка производительности калькулятора: скорость выполнения операций, стабильность работы.

4. Тестирование и отладка:

• Результаты тестирования: количество найденных и исправленных ошибок.

• Методы тестирования, использованные для проверки функциональности и надежности.

5. Отзыв пользователей:

• Сбор отзывов от пользователей: что им понравилось, а что можно улучшить.

• Анализ отзывов для выявления возможных направлений для доработки.

6. Документация:

• Наличие и качество документации для пользователей и разработчиков.

• Описание процесса установки и использования калькулятора.

7. Временные рамки:

• Сравнение фактических временных затрат с запланированными.

**Итоги:**

Эти шаги позволяют создать функциональный и надежный калькулятор, который сможет выполнять основные математические операции с точностью и эффективностью.