СОДЕРЖАНИЕ

1. Анализ предметной области

Необходимо разработать аналог обычного калькулятора Windows, внешний вид которого изображён на рисунке 1. Соответственно, интерфейс для реализации необходимого согласно ТЗ функционала будет заимствован у данного калькулятора.

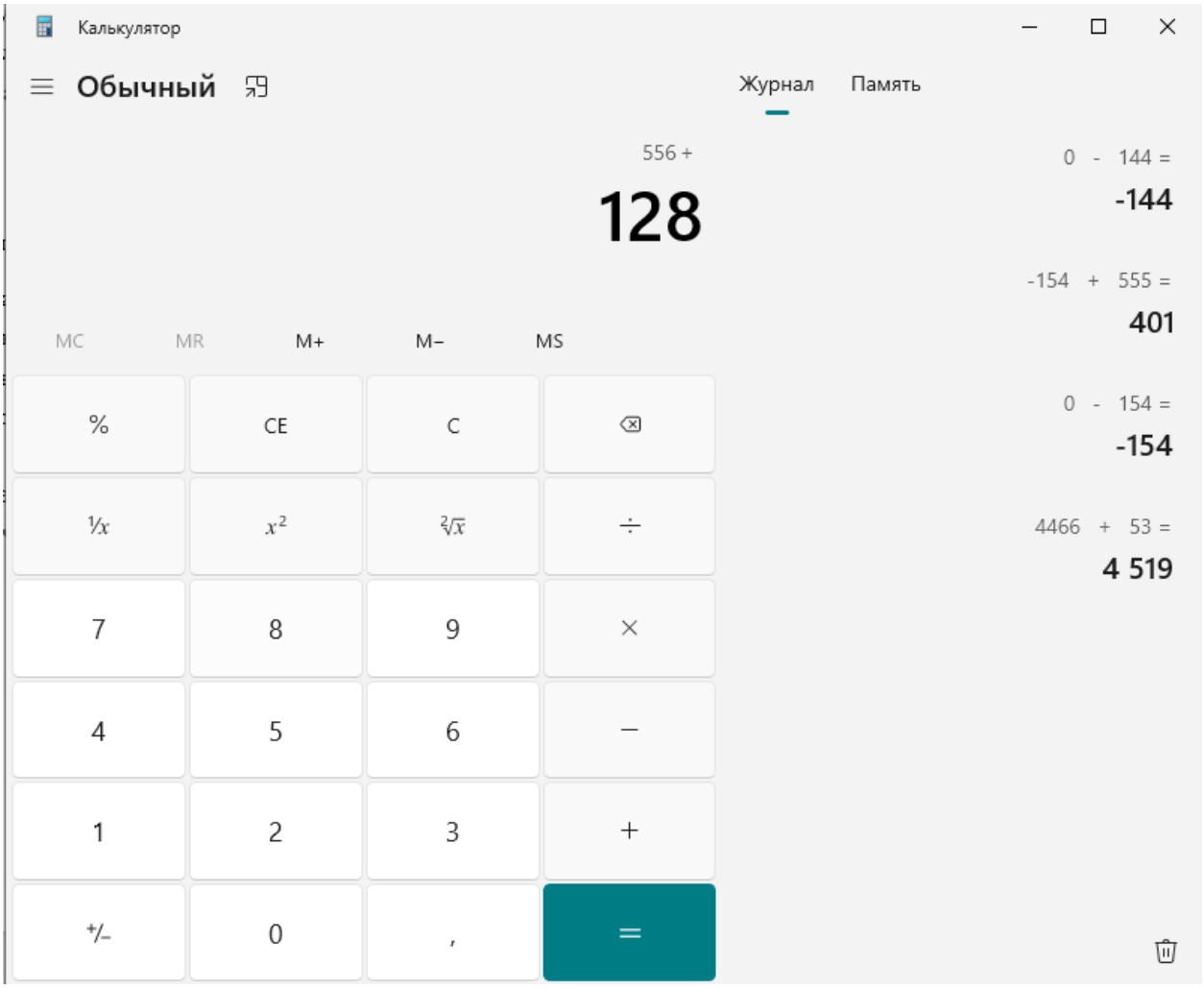

Рисунок 1. Внешний вид обычного калькулятора windows

2. Требования к ПО

Данная глава позволяет структурировать требования к ПО, понять общий функционал системы и выделить нефункциональные требования, которые необходимо реализовать в рамках ТЗ.

2.1. Функциональные требования:

1. Осуществление последовательного ввода операндов.
   1.1. Ввод второго операнда невозможен, если не введён первый.
   1.2. Ввод второго операнда возможен только лишь после того, как выбрана арифметическая операция.
   1.3. Ввод второго операнда оканчивается нажатием ‘=’.
   1.4. Ввод отрицательного числа начинается с постановки ’–'.
2. Выполнение операций + - / * с участием двух операндов.
3. Отправка в очередь операций при нажатии на ‘='.
   3.1. Если запрос не валидный: не добавляется в очередь, происходит очистка ввода, выводится сообщение в консоль.
4. Отображение запросов на вычисление, результаты вычислений, ошибки вычислений в консоли.
5. Обработка операции в соответствии с текущим временем на операцию с занесением результата в очередь результатов вычислений.
   5.1. При возникновении ошибки возвращать ошибку в параметре-указателе (в соответствии с ТЗ).
6. Изменение времени выполнения вычисления.
   6.1. Если время меньше 0: ошибка ввода.
   6.2. Если время равно 0: время обработки равно реальному времени операции.
   6.3. Если произведено нажатие в момент, когда ожидается ввод второго операнда: ошибка ввода.
7. По вызову извлечение первого результата из очереди с отображением на консоли
8. Отмена действия – удаление правого символа текущего операнда (нет в ТЗ, но необходима для удобства).
9. Очистка всех операндов (нет в ТЗ, но необходима для удобства).

2.2. Нефункциональные требования:

1. Организация запросов на вычисление в виде очереди.
2. Организация результатов вычислений в виде очереди.

3. Реализация двух потоков.
4. При выходе из приложения сохранять текущее положение и форму экрана, а при входе восстанавливать.
5. Отображать размеры очередей результатов и операций.
6. Консоль с цветным текстом: запросы - зеленые, результаты - синие, ошибки – красные.
7. Эргономичность интерфейса.
8. Обеспечение красивого и при этом доступного для пользователя интерфейса.
9. Обеспечение обработки ошибок: ввода, многократное нажатие кнопок, вычисления.
10. Все операции калькулятора выполняются из внешней SO на C++.

Примечание:

Для нефункциональных требований отсутствует разделение на подпункты в связи с малым объёмом задания.

2.3. Наполнение потоков функционалом:

1. Поток 1:
   1.1. По прошествии времени предыдущего вычисления из очереди запросов берется следующий элемент (если он там есть).
   1.2. Извлеченный элемент отправляется на обработку.
   1.3. Ожидание окончания вычисления и укладка в очередь результатов вычислений.
   1.4. Переход к пункту 1.1.
2. Поток 2:
   2.1. Выполнение основной работы программы (не включённой в поток 1).
   2.2. Формирование новых запросов в очередь запросов.
   2.3. Вывод результатов из очереди результатов.
   2.4. Переход к пункту 2.1.

Общий функционал пользователя можно отразить в виде диаграммы вариантов использования, отображённой на рисунке 2.

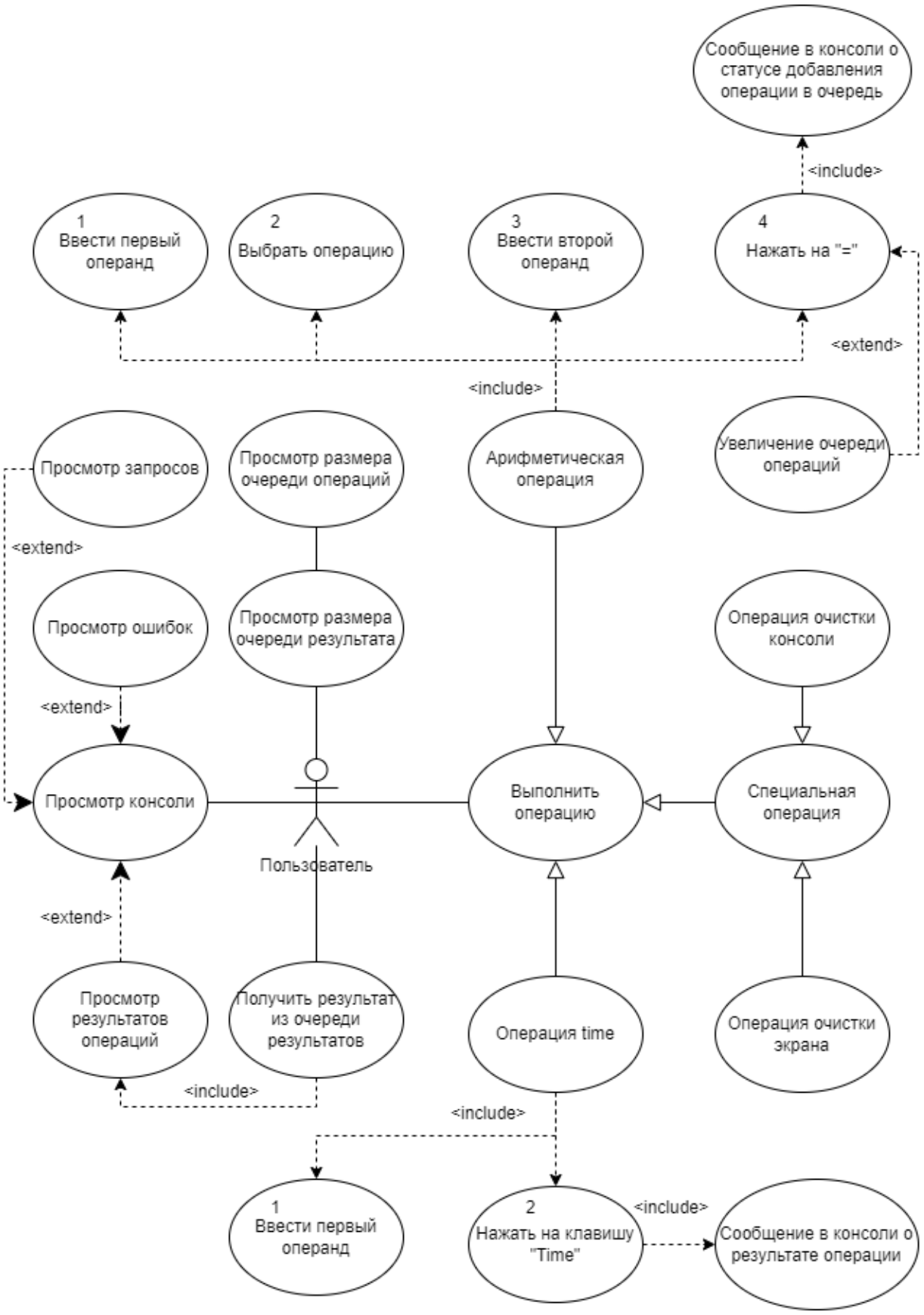

Рисунок 2 – Диаграмма вариантов использования для пользователя приложения “Калькулятор”

3. Навигационная диаграмма интерфейсов

В данном случае, для улучшения показателей эргономики приложения в приложении будет реализован интерфейс, состоящий из одного окна. В соответствии с этим навигация между окнами не разрабатывается. Интерфейс проектируемого приложения “Калькулятор” изображен на рисунке 3.

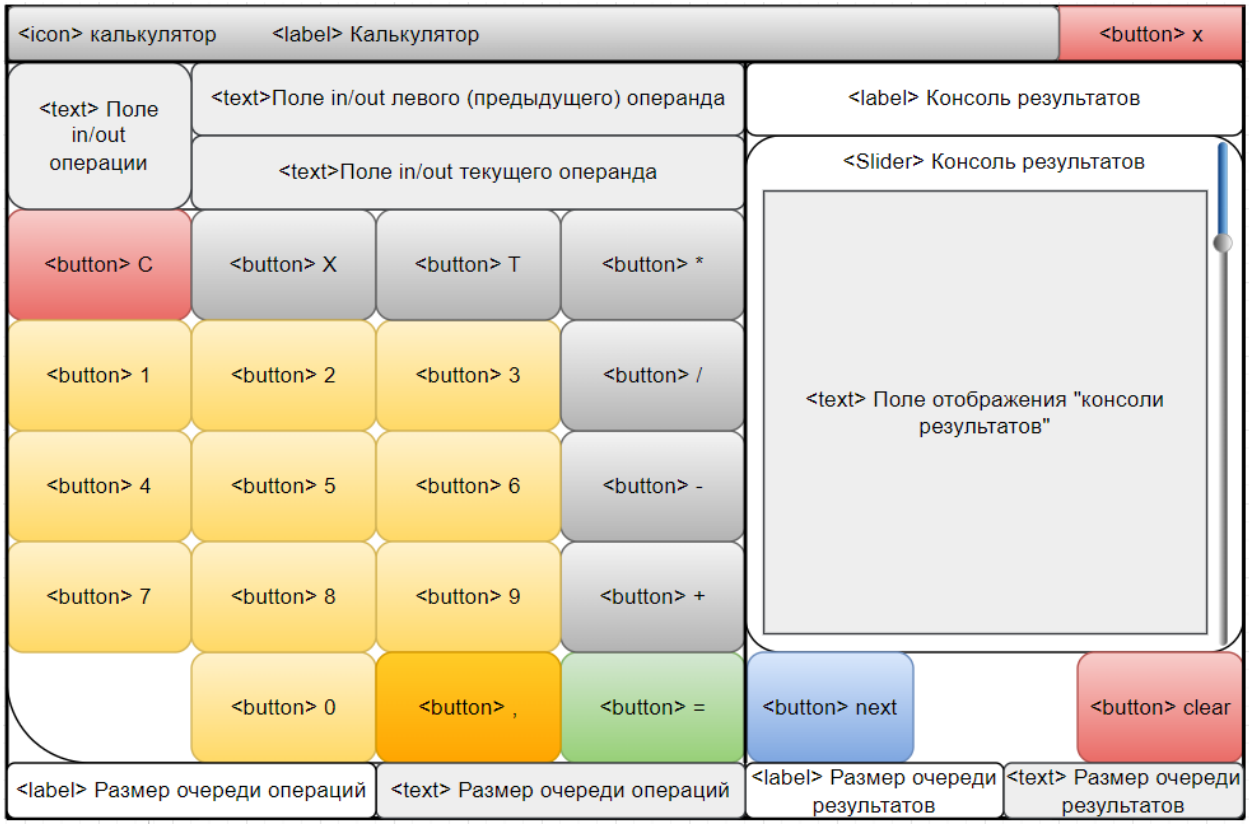

Рисунок 3 - Основной интерфейс приложения “Калькулятор”

Обозначения рисунка 3 имеют следующие определения:

a) <label> - отображаемый текст, как есть (в контексте не изменяем);

b) <icon> - картинка;

c) <button> - кнопка, при нажатии на которую происходит обработка;

d) <text> - текстовое поле (отображает изменяемую информацию);

e) <slider> - область прокрутки;

f) <textedit> (не используется) - поле, предназначенное для ввода текста.

На рисунке 4 отображена смысловая и функциональная нагрузка отдельных полей, представленных в шаблоне интерфейса на рисунке 3.

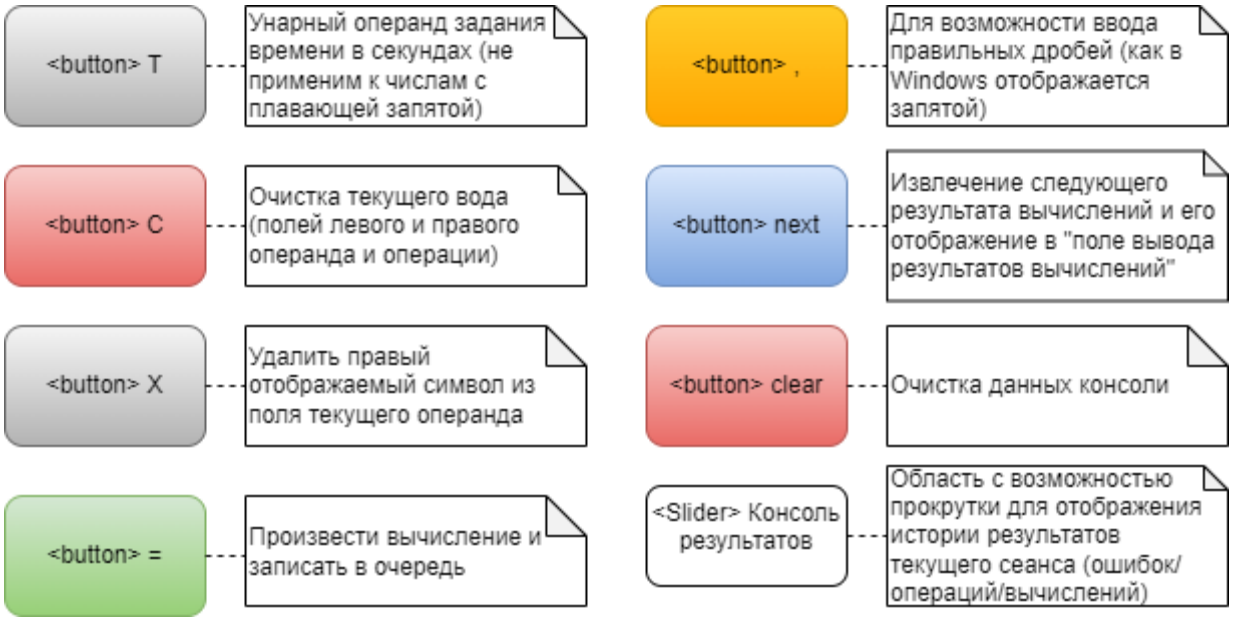

Рисунок 4 – поля приложения калькулятор и их смысловая нагрузка

4. Основные вопросы

1. ***Внешняя SO – это что имеется ввиду в контексте ТЗ?*** Я бы перевёл как получение Source object, а для этого достаточно просто отдельный cpp файл организовать. Но тут фигурирует слово “ВНЕШНЕЙ” – что наталкивает на мысль, что нужно делать отдельную внешнюю либу и её подключать. Если ни один из двух вариантов не является корректным, то применим третий вариант: я просто не понимаю значение аббревиатуры SO и мне необходимо её пояснить и прокомментировать, что именно предполагает ТЗ в этой части.

2. ***Какой критерий по “тыканью” может давать ошибку ввода?*** Это из разряда – изучите определённый набор людей; зная кто специально натыкивает, а кто нет, определите среднее время запросов/среднее количество запросов в определённый промежуток времени и вычислите границу между нормальными и теми, кто натыкивает))) Никогда с таким не сталкивался ни в одном калькуляторе. Если не задумываться о данной ошибке, то в случае, если пользователь тыкает на всё подряд, то дальше он может не пройти верификацию – конечно, минус в том, что он всё таки может пройти проверку и тогда ‘фейковые’ запросы могут нагрузить очередь и вычисления. ***Единственное, что лезет в голову, сравнивать операции полной очистки и ‘=’, если они нажимаются более одного раза подряд - результат просто вывод сообщения (только смысла особого в этом нет). Хотелось бы, чтобы вы прояснили этот момент.***

3. ***При ошибках ввода: производить очистку ввода/оставить последнее состояние/оставить на моё усмотрение*** (понятно, что ошибка выведется в консоль в любом случае)***?*** Пример такой ошибки: пользователь после ввода всех операндов нажимает не на ‘=’, а на любую другую операцию.

4. Кроме действий, которые описаны в ТЗ ***были добавлены действия, направленные на очистку ввода/вывода для удобства использования***.

5. В ТЗ прописано сохранять положение и форму окна при выходе. О сохранении данных и последнего отображения (в том числе консоли), ничего не написано. ***Соответственно делаю вывод, что сохранение данных при выходе не произвожу.***

6. По причине отсутствия ограничений на хранение данных в консоли, как и в журнале Windows-калькулятора, ***консоль будет хранить все результаты за сеанс и только в рамках сеанса.***