

СОДЕРЖАНИЕ

1. Анализ предметной области.....	2
2. Требования к ПО.....	3
2.1. Функциональные требования:	3
2.2. Нефункциональные требования:	3
2.3. Наполнение потоков функционалом:	4
3. Навигационная диаграмма интерфейсов	6
4. Диаграмма классов.....	8
5. Блок-схемы алгоритмов.....	10
6. Сокращённое руководство пользователя.....	21

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.

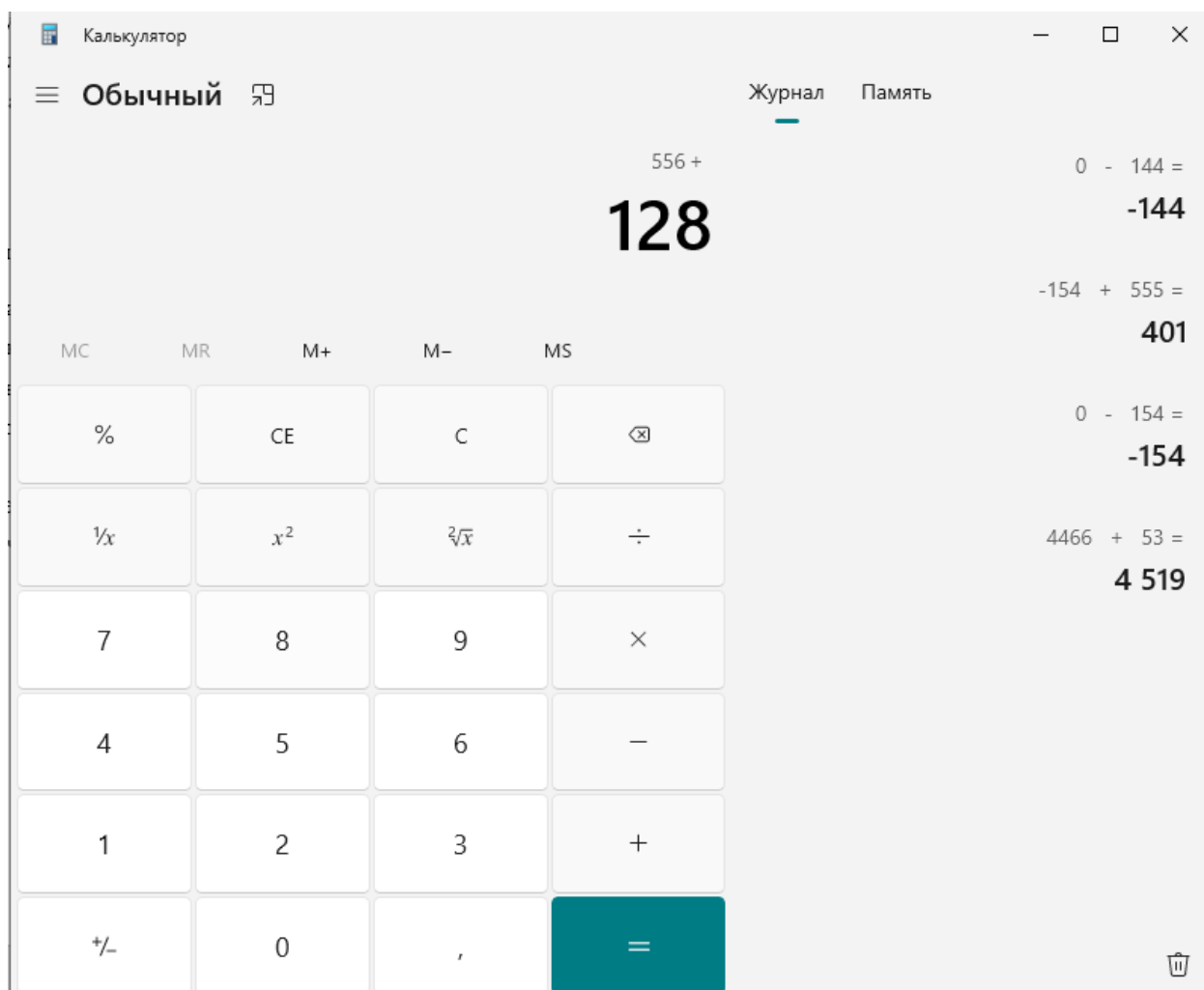


Рисунок 1. Внешний вид обычного калькулятора windows

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.

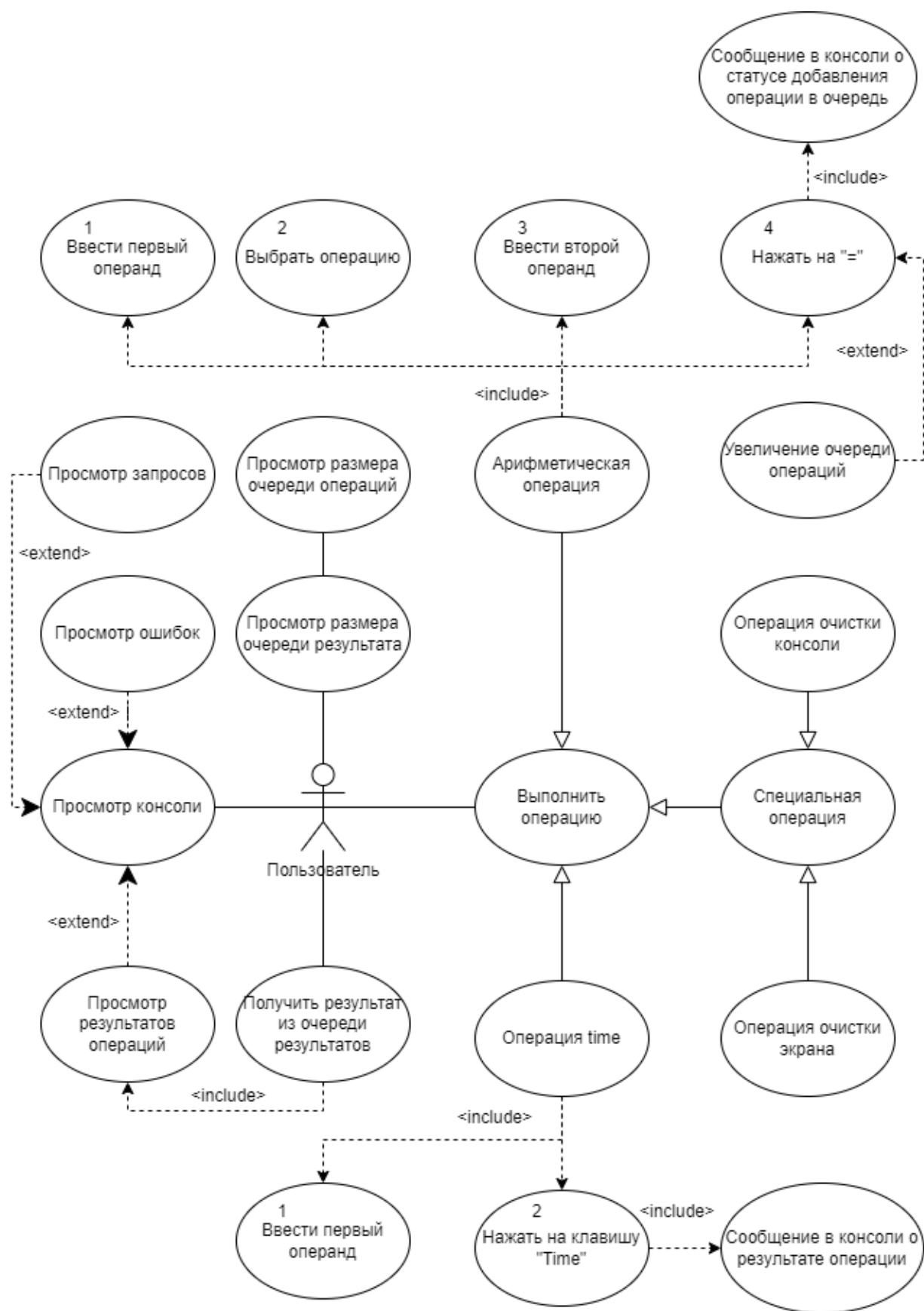


Рисунок 2 – Диаграмма вариантов использования для пользователя приложения
“Калькулятор”

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.

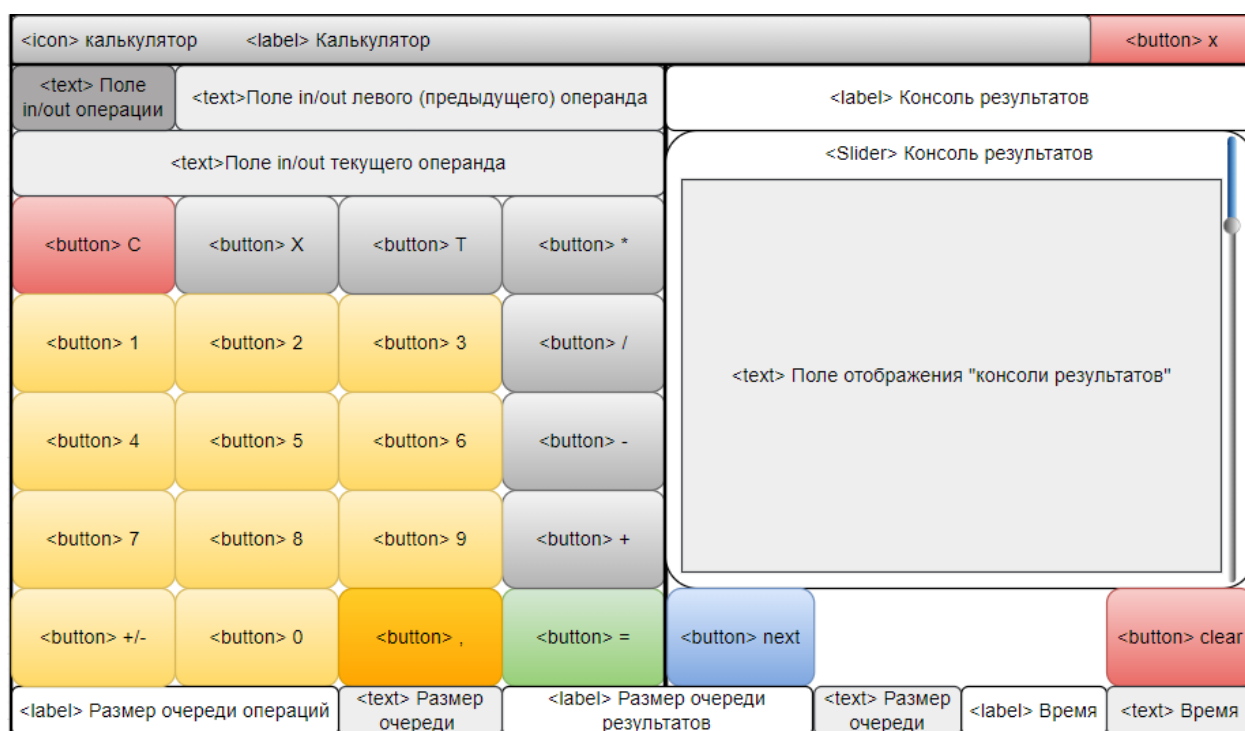


Рисунок 3 - Основной интерфейс приложения “Калькулятор”

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. Диаграмма классов

Отношения между классами, а также их базовую структуру можно описать в виде диаграммы классов, представленной на рисунках 5-6. Пояснения к диаграмме классов могут быть найдены в файлах-источниках программы, где каждый класс, поле и метод имеет пояснение своего назначения.

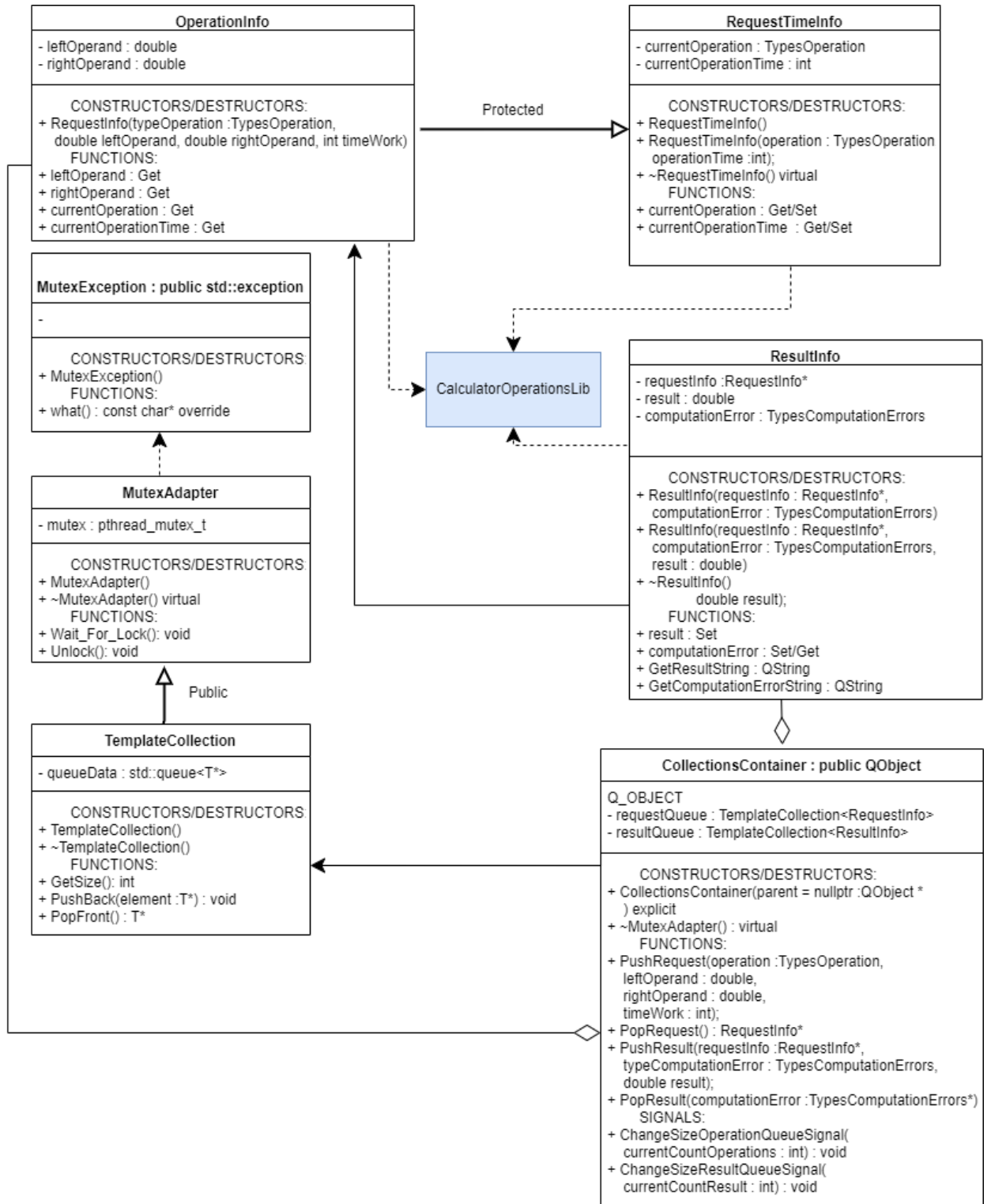


Рисунок 5 - Модель области “Калькулятор” в виде UML-диаграммы классов, часть 1

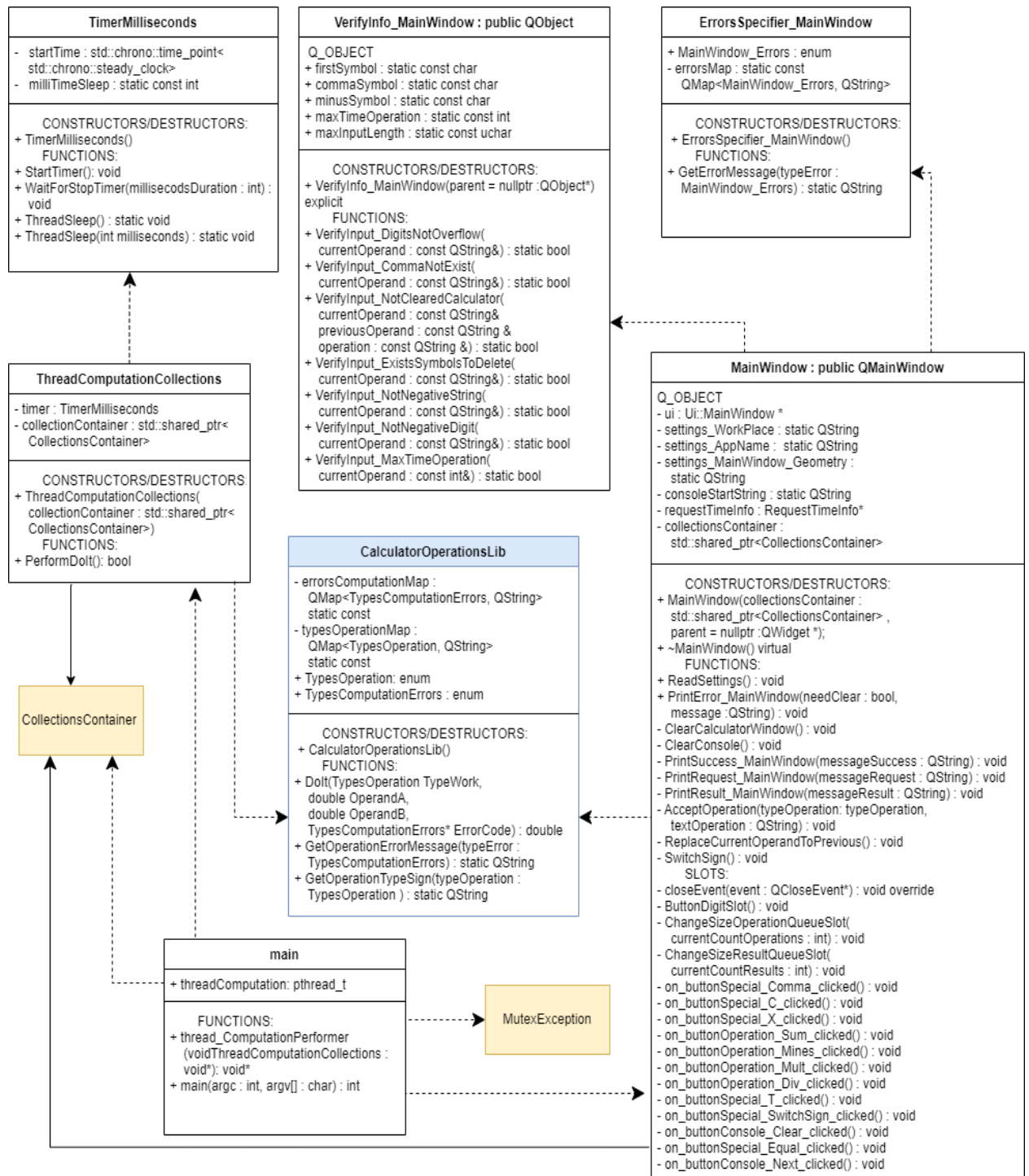


Рисунок 6 - Модель области “Калькулятор” в виде UML-диаграммы классов, часть 2

5. Блок-схемы алгоритмов

Алгоритмы работы программы могут быть описаны в виде блок-схем, которые отображают последовательность действий для достижения того или иного результата. Блок-схемы алгоритмов разрабатываемого ПО “калькулятор” представлены на рисунках 7-18. Каждая схема составлена для собственной операции ввода, кроме того, отдельно были рассмотрены задачи потоков приложения.

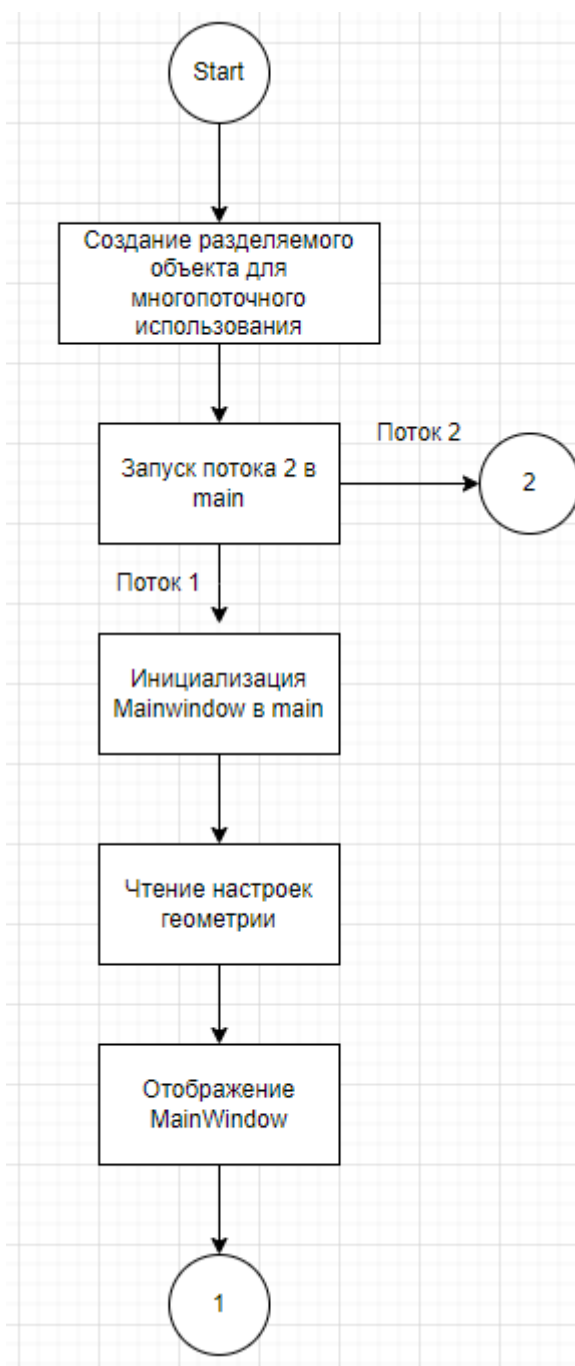


Рисунок 7 - Блок-схема работы main

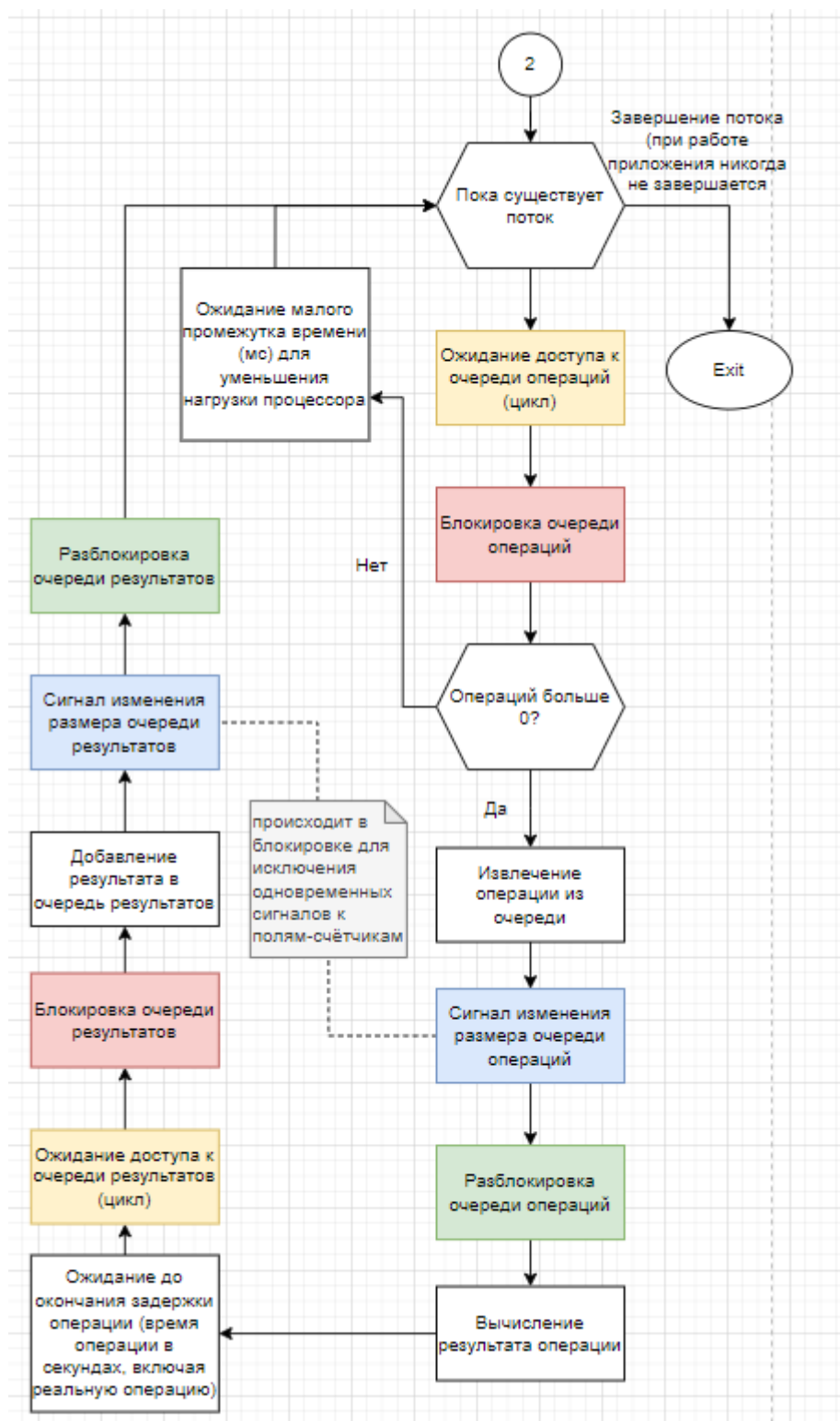


Рисунок 8 - Блок-схема работы второго потока

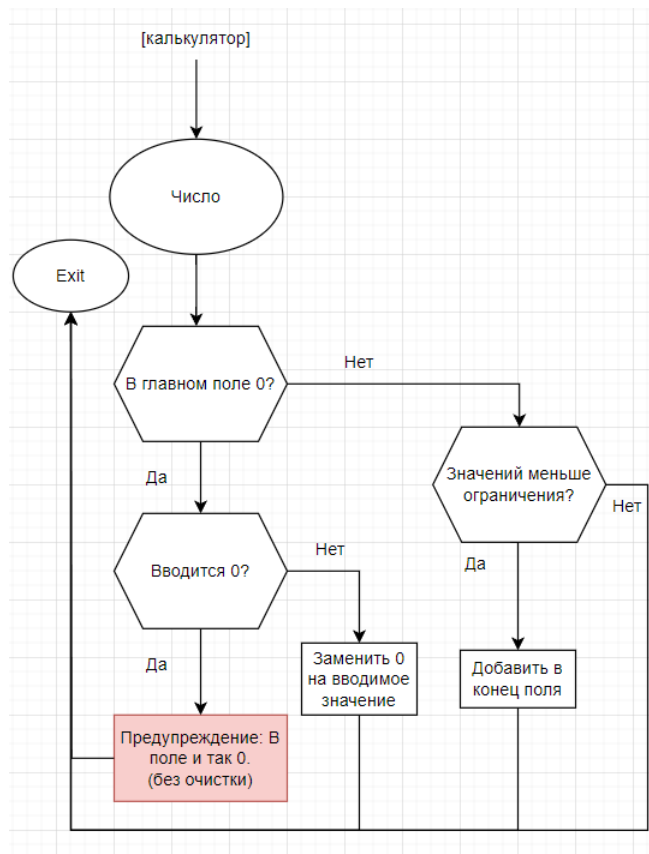


Рисунок 9 - Блок-схема ввода числа

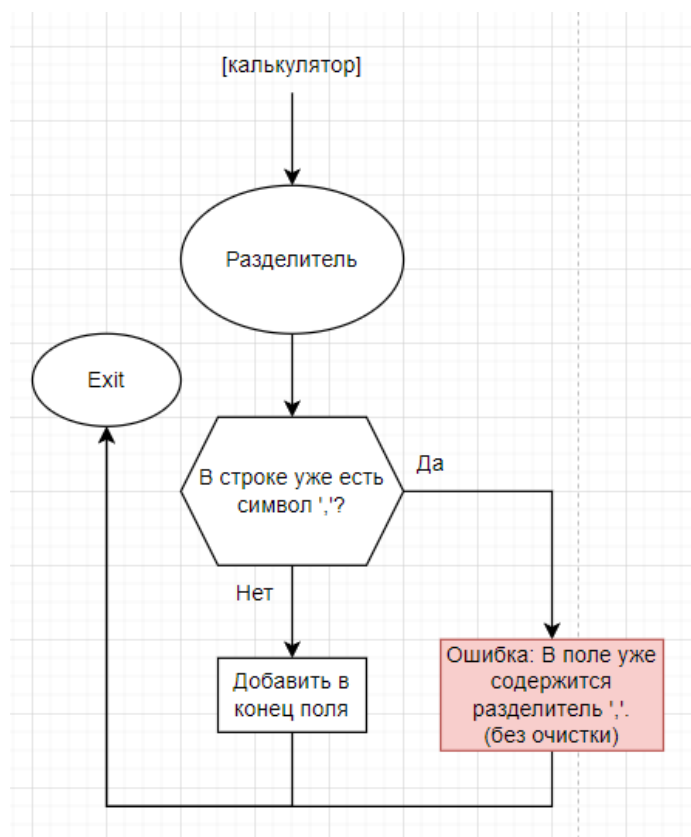


Рисунок 10 - Блок-схема ввода разделителя

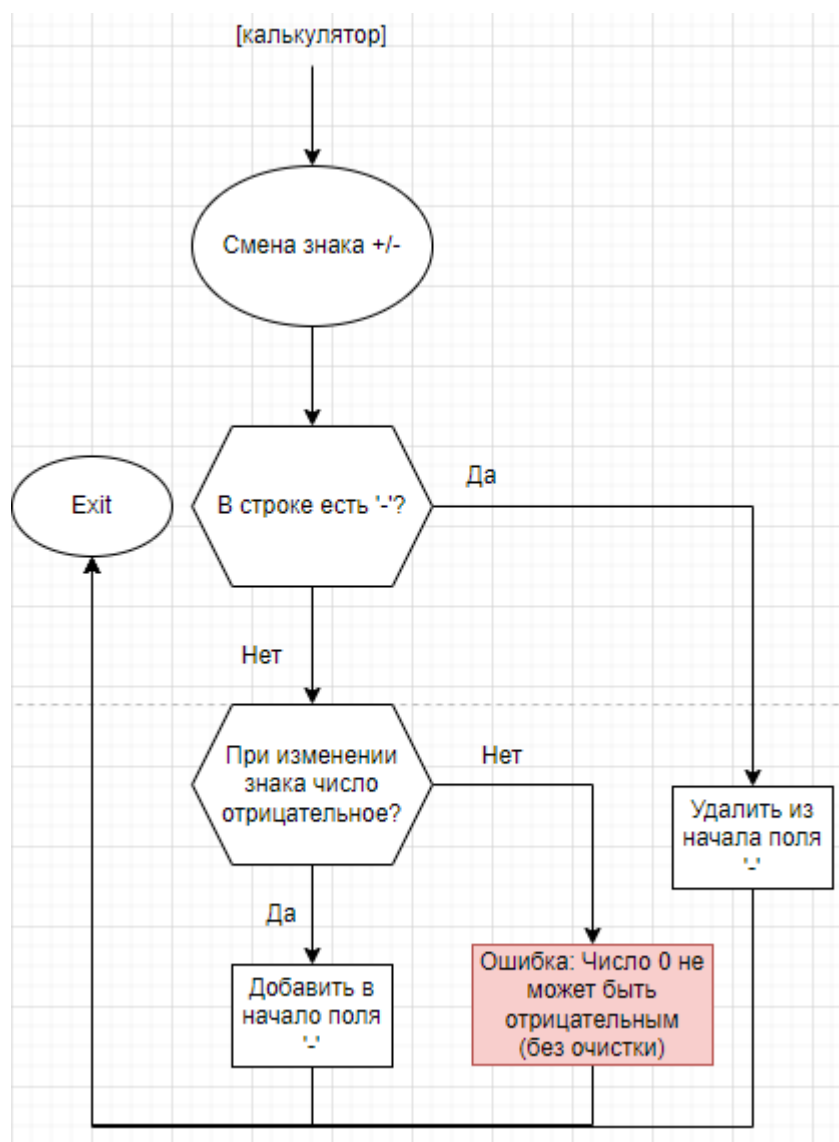


Рисунок 11 - Блок-схема для смены знака числа

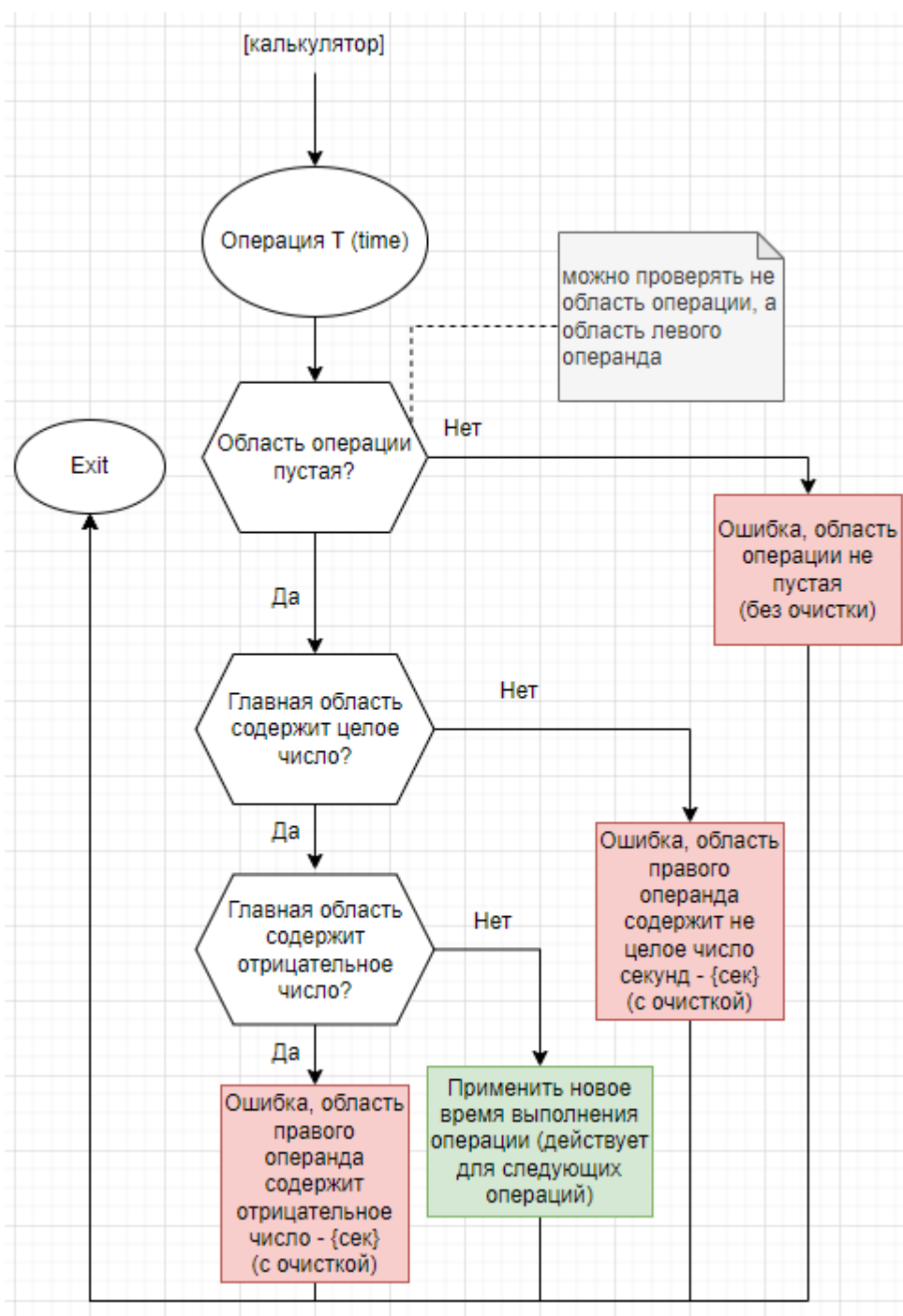


Рисунок 12 - Блок-схема применения времени операции

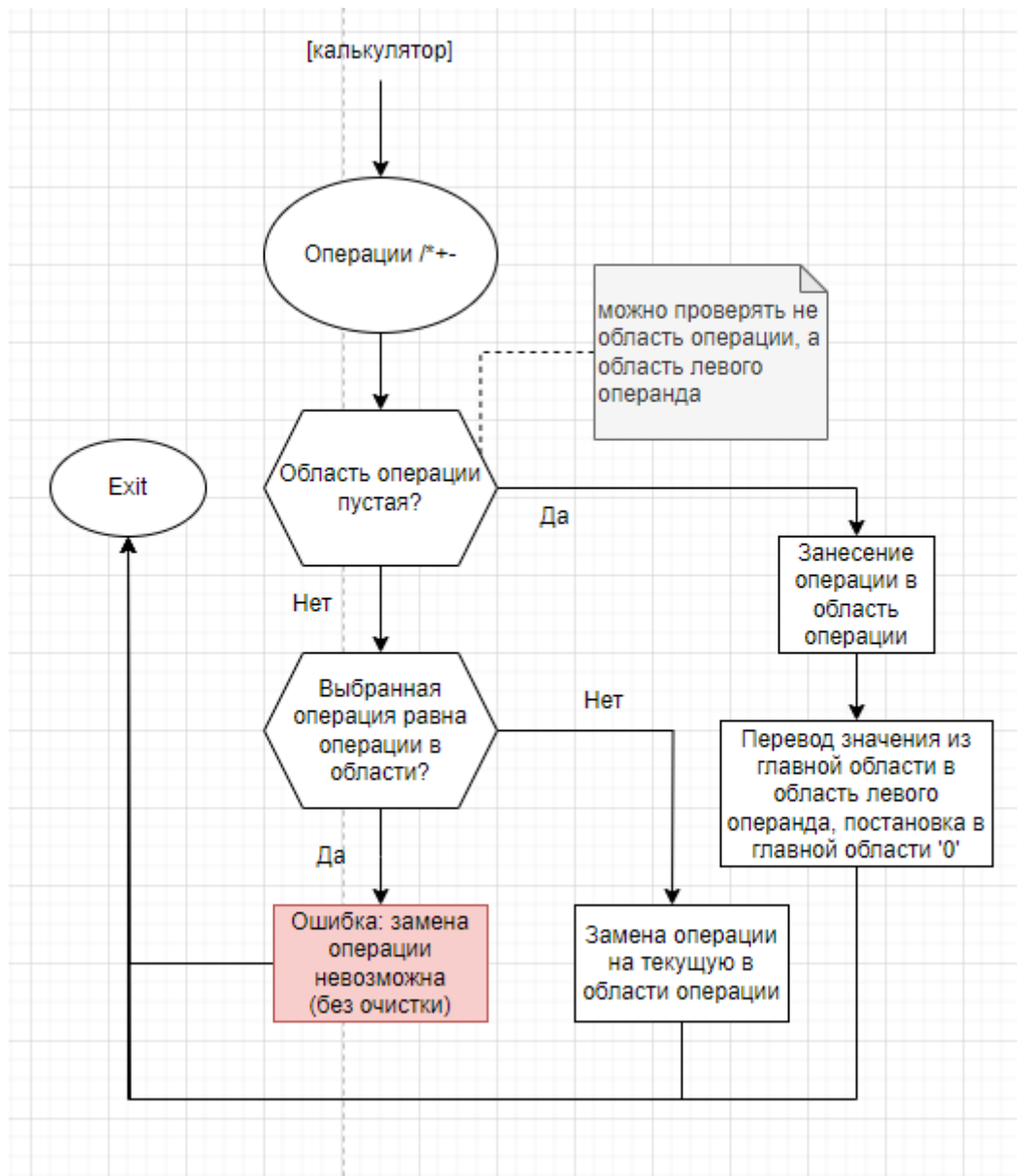


Рисунок 13 - Блок-схема применения арифметической операции

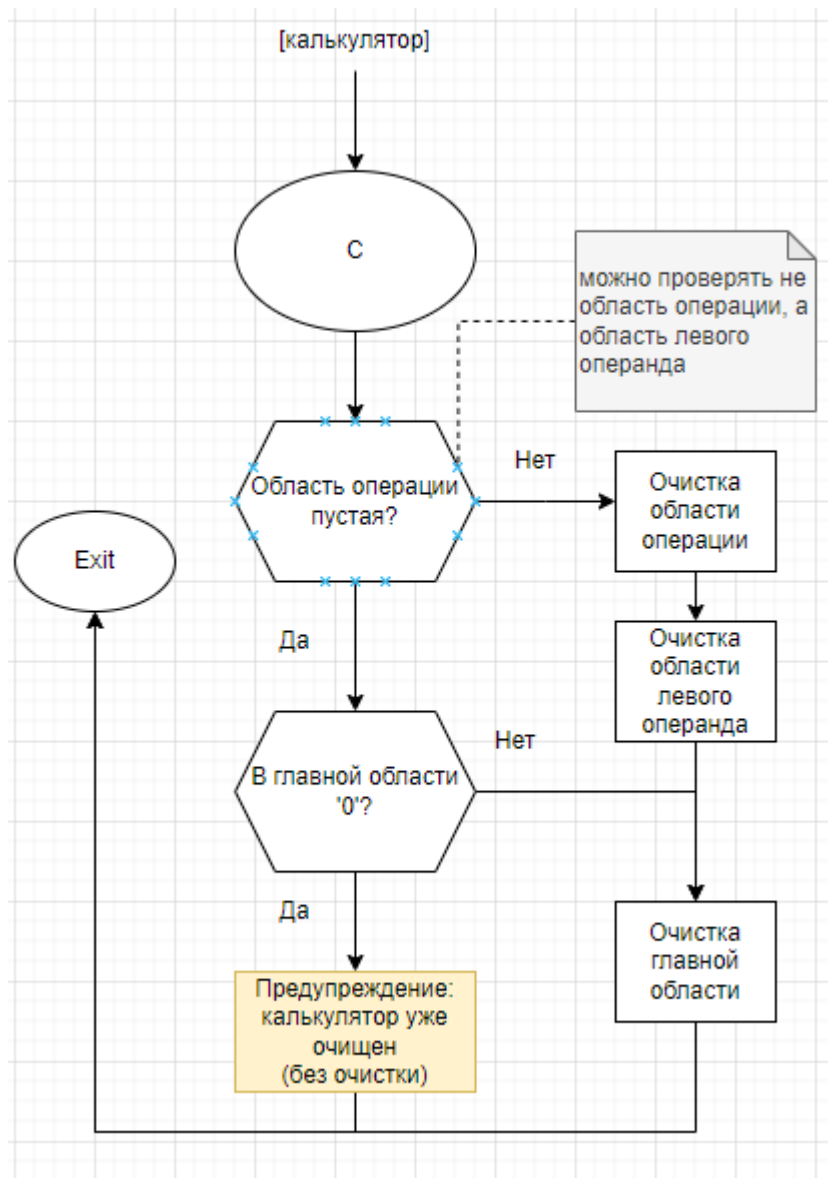


Рисунок 14 - Блок-схема очистки области калькулятора

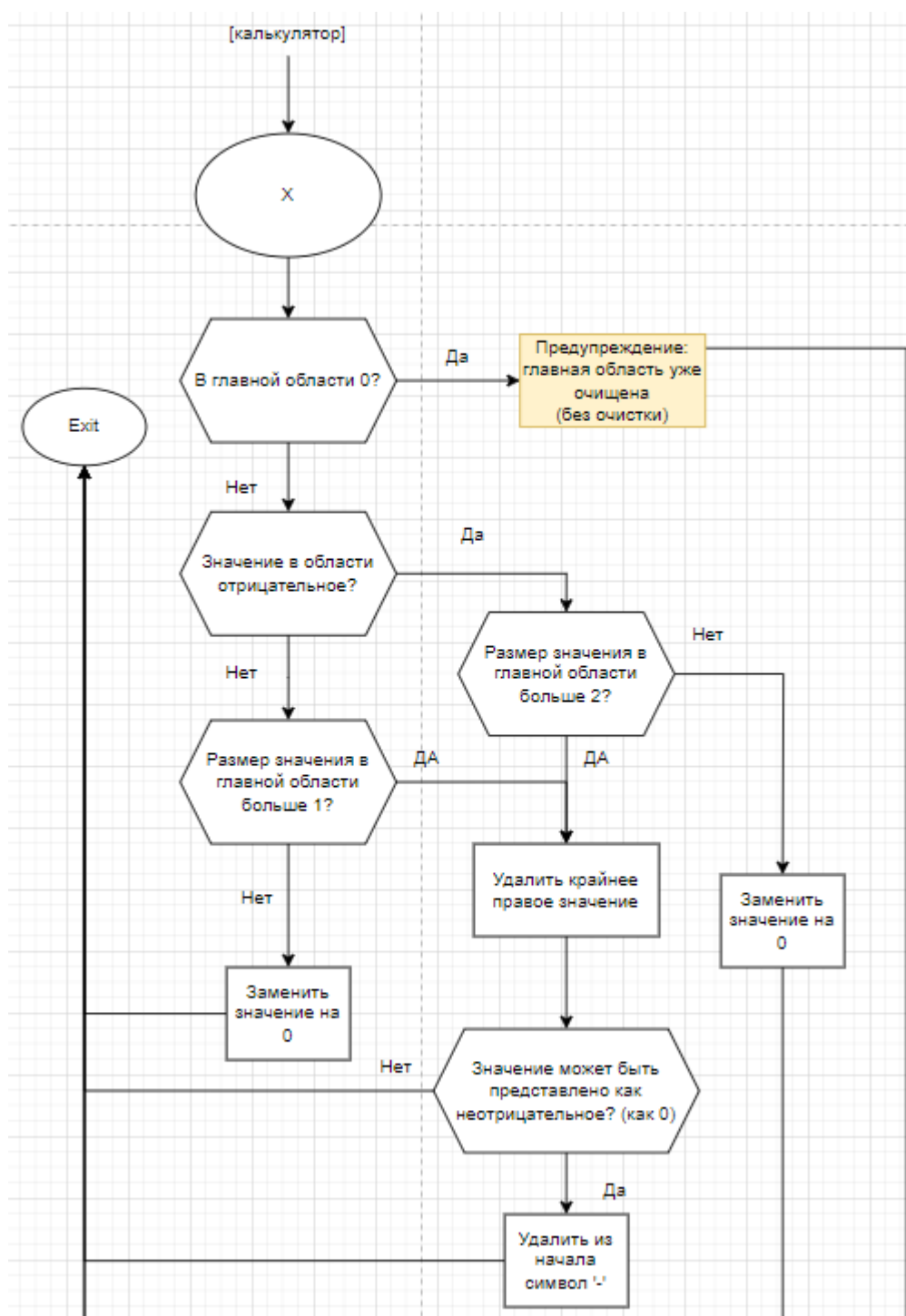


Рисунок 15 - Блок-схема удаления символа

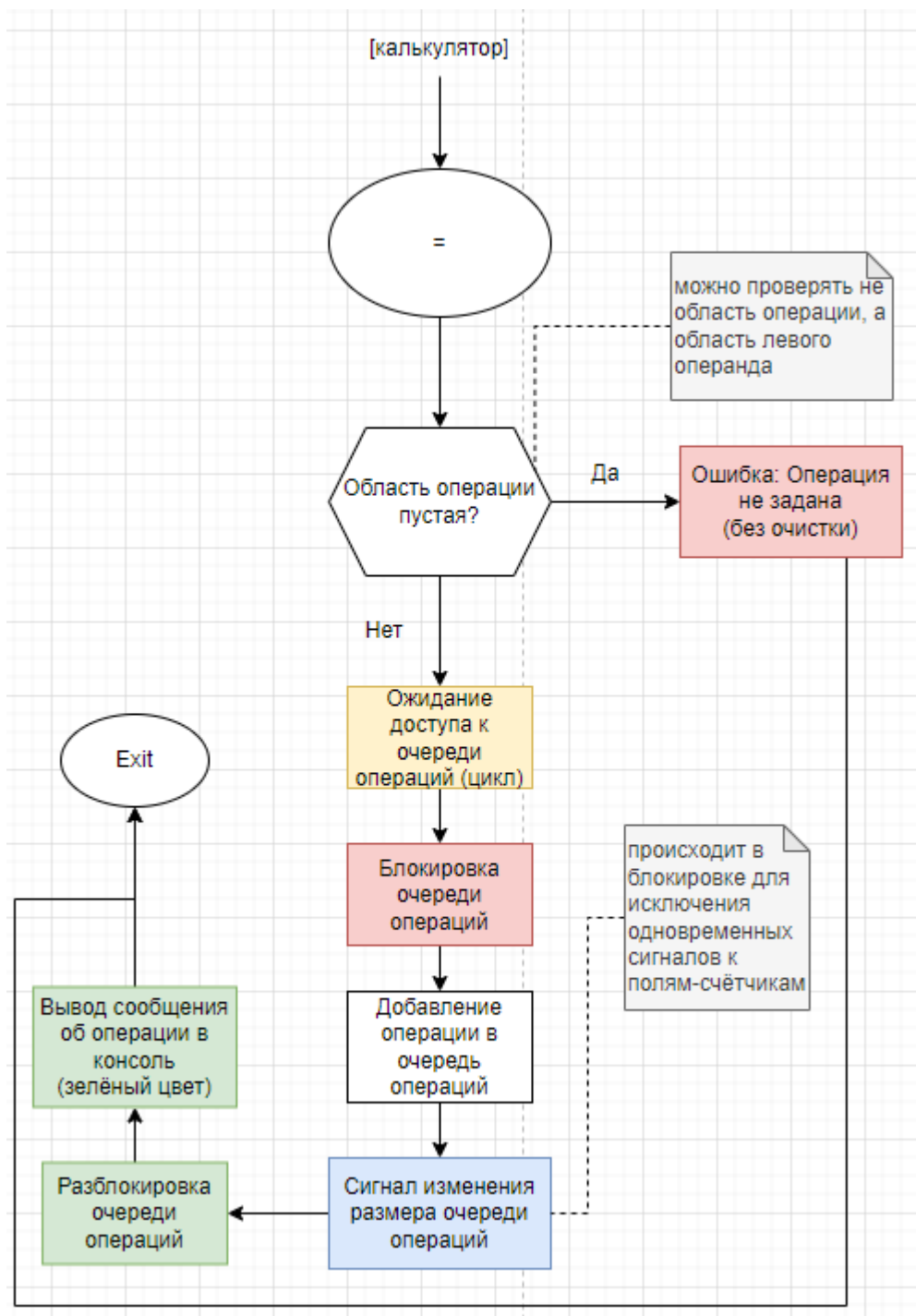


Рисунок 16 - Блок-схема добавления запроса в очередь

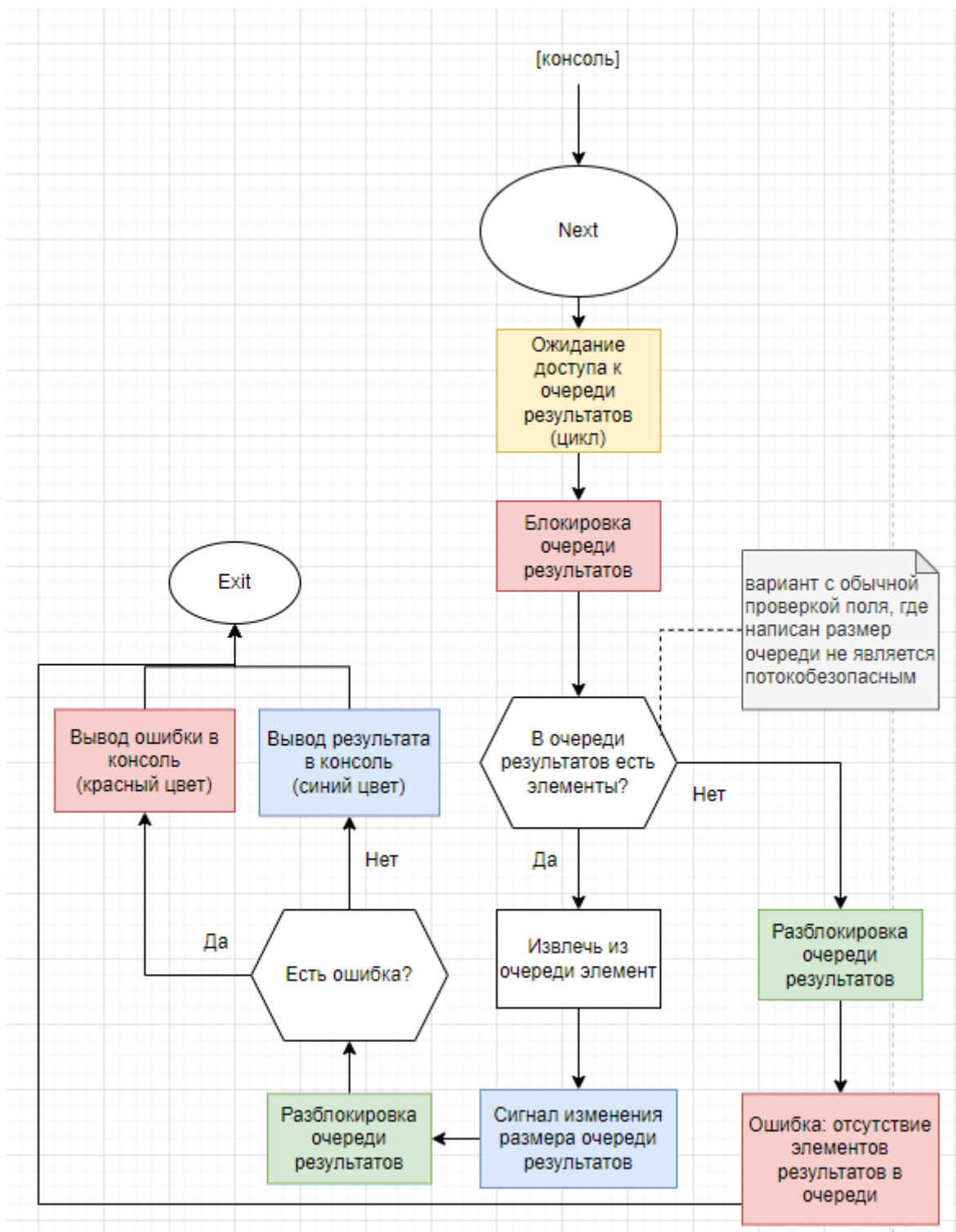


Рисунок 17 - Блок-схема получения результата из очереди результатов



Рисунок 18 - Блок схема операции очистки консоли

6. Сокращённое руководство пользователя

Данное руководство поясняет то, как работать с интерфейсом приложения, но не отображает алгоритмы достижения тех или иных операций (поэтому и сокращённое). Алгоритмы выполнения операций доступны и детально описаны в виде блок-схем в пункте 5.

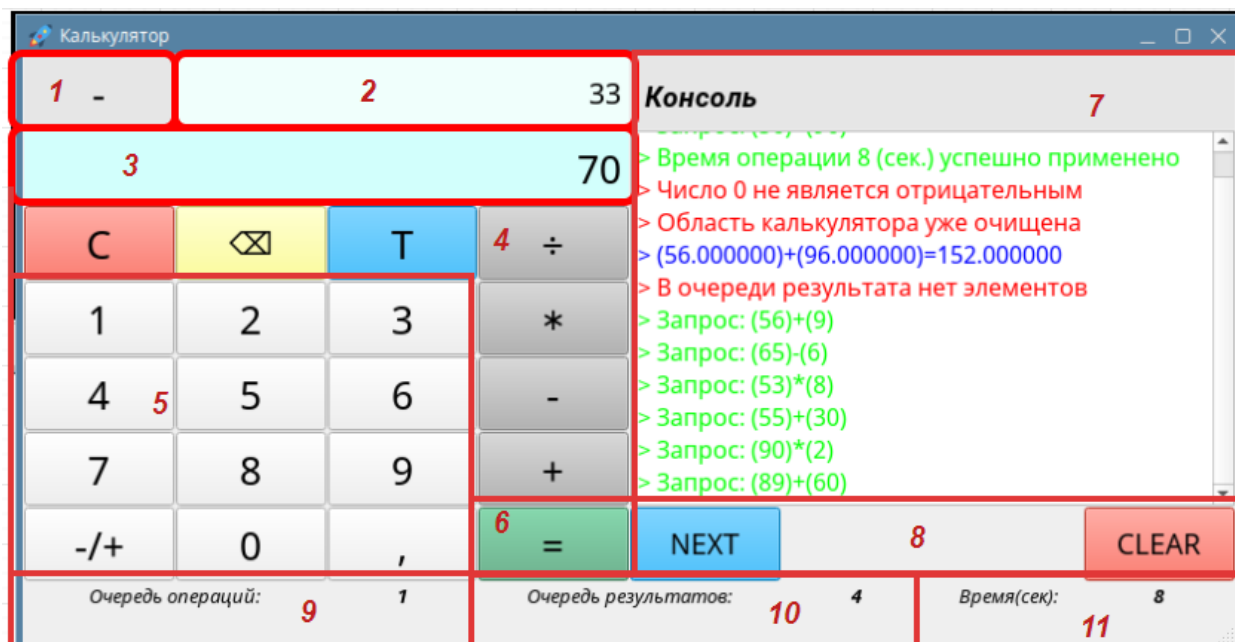


Рисунок 19 – Разделение интерфейса на области с подписью

Области интерфейса приложения и пояснения в соответствии с рисунком 19:

- 1) Область отображения операции.
- 2) Область левого операнда – заполняется значением из области 3 при первом нажатии любой из операций (+-*.).
- 3) Область правого операнда и текущего ввода.
- 4) Области операций слева направо и сверху вниз:
 - Очистка экрана калькулятора;
 - Удалить крайний правый символ из области 3;
 - Изменить время выполнения операции в секундах для всех последующих операций (отсутствие задержки – ввод 0);
 - Арифметические операции деления, умножения, вычитания и сложения.
- 5) Ввод чисел в область 3 с возможностью смены их знака (знак меняется только тогда, когда введен не 0).

- 6) Операция 'равно' или выполнить - производит добавление запроса в очередь операций (при наличии операции в поле 1).
- 7) Консоль с выводом сообщений о запросах (зелёный текст), результатах (синий текст) и ошибках (красный текст) и возможностью прокрутки.
- 8) Область функций консоли слева направо:
- Извлечь результат из очереди результатов;
 - Очистить консоль.
- 9) Отображение размера очереди операций в реальном времени.
- 10) Отображение размера очереди результатов в реальном времени.
- 11) Отображение текущего времени операций (применяется для всех операций, которые вводятся далее до тех пор, пока не будет изменено текущее время операции) – позволяет для каждой отдельной операции установить собственное время выполнения и имитировать большую длительность выполнения.

Особенности ПО:

- 1) При выходе сохраняет свои размеры и положение, восстанавливает при следующем открытии.
- 2) Не сохраняет содержимое консоли и не предоставляет инструменты для их внутреннего сохранения. В связи с этим при завершении работы программы все введённые данные стираются и не отображаются при последующем открытии.
- 3) Если в очереди операций не менее 1 запроса, то стоит учитывать, что 1 запрос находится в обработке и не содержится ни в одной из очередей до тех пор, пока не будет получен результат и занесён в очередь результатов.
- 4) Время обработки запроса удовлетворяет заданному времени в секундах. Если время обработки запроса поставлено в 0, то задержка после вычислений производиться не будет.
- 5) Были заданы следующие контекстные ограничения:
 - время выполнения операции не более 100 секунд (для того, чтобы обезопасить от слишком большого времени ожидания завершения операции при проверке),

- размер строки без учёта знака ‘-’ 48 символов (позволяет отобразить строку на мониторе без необходимости прокрутки с диагональю от 14 дюймов).
- 6) Проверки на конкретный блокирующий поток не осуществляется, т.к. предполагается, что если один поток произвёл блокировку, то далее он произведёт и освобождение (при необходимости разграничения блокирующих потоков решается добавлением переменной текущего блокирующего потока как поля в `MutexAdapter`).
- 7) Т.к. в ТЗ не были заданы свойства интерфейса по точности вывода, то достижения оптимального варианта визуализации не достигалось. Таким образом, результат может быть обрезан в соответствии с дефолтными ограничениями выводящей функции (6 символов после запятой).