Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«Пермский национальный исследовательский политехнический университет» (ПНИПУ)

Электротехнический факультет

Кафедра информационных технологий и автоматизированных систем

Направление подготовки 09.03.01 «Информатика и вычислительная техника»

**Отчёт по творческой работе**

**по дисциплине**

**«Теория алгоритмов и структуры данных»**

**семестр** **2**

Выполнил студент гр. ИВТ-21-1б

Балдин Алексей Игоревич

Проверил:

Яруллин Денис Владимирович

\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

(оценка) (подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата)

г. Пермь – 2022

**Постановка задачи**

Создать приложение калькулятор и сделать задачу коммивояжера в соответствие с вариантом.

**Описание решения задачи коммивояжёра**

1. Для задачи коммивояжёра был применен метод «ветвей и границ», так как этого требует условие
2. Написание кода было выполнено на языке программирование «С++»

Также для визуализации идеи воспользовался программой «OpenGl», работал с данной программой, потому что она проста в понимании и по ней есть множество обучающих видео.

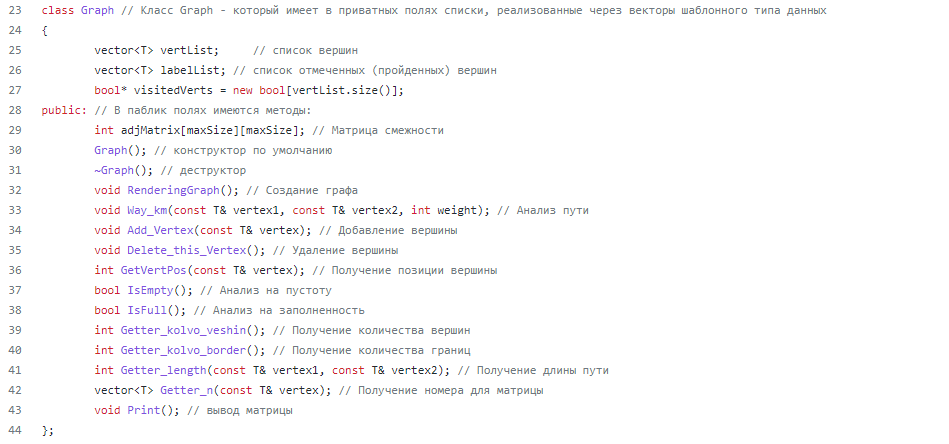
1. a) Возникли трудности с пониманием координат, которые смещают интерфейс, то есть была проблема - размещения «меню» и «кнопок» на дисплее. Но, поразмышляв над данной проблемой, рассмотрев несколько случаев, понял, как себя «ведут» координаты на дисплее.

б) Также была трудность в понимании алгоритма самой задачи, посмотрев различные пояснительные видео, все-таки понял, какая последовательность действий должна быть, и к какому результату нас приведёт тот или иной шаг в алгоритме.

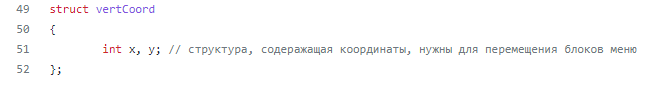
1. Автор гордиться тем, что сами пункты отображаются друг от друга на определенном расстоянии, также автор гордиться, что реализовано качественное «меню», то есть понятно за что отвечает та или иная «кнопка», нельзя не отметить дизайнерское решение, оно не является сложным, но оттенки цветов, их сочетаемость абсолютно не режет глаза, в меру приятные краски, не зря же говорят: «Все гениальное - просто». Также хочу отметить то, что в программе реализованы проверки на то, чтобы нельзя было вводить отрицательные «пути», а также реализовано их возможное переопределение. И, последнее, что хотел бы отметить, функция «reshape» - которая позволяет графу оставаться в поле видимости, при изменении самого окна.

**Описание основных моментов в коде**

В данном участке кода создаю класс Graph – в котором в private поле списки, реализованные через векторы. В public поле присутствуют методы, которые и позволяют нам производить действия над графом, также в этом поле есть матрица смежности, заполненная, пока что, максимальными элементами.

****

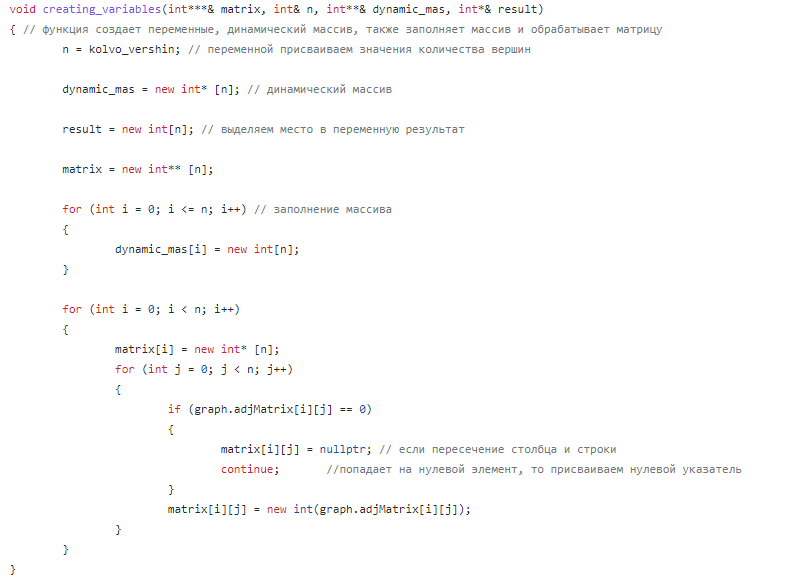
Структура, за счет которой смещается «меню», содержит поля x, y

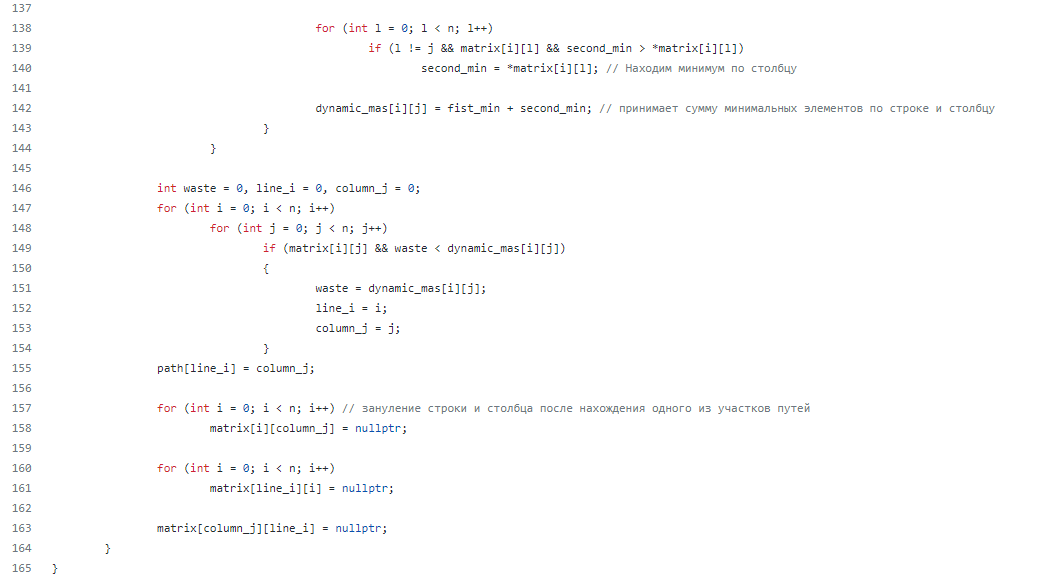
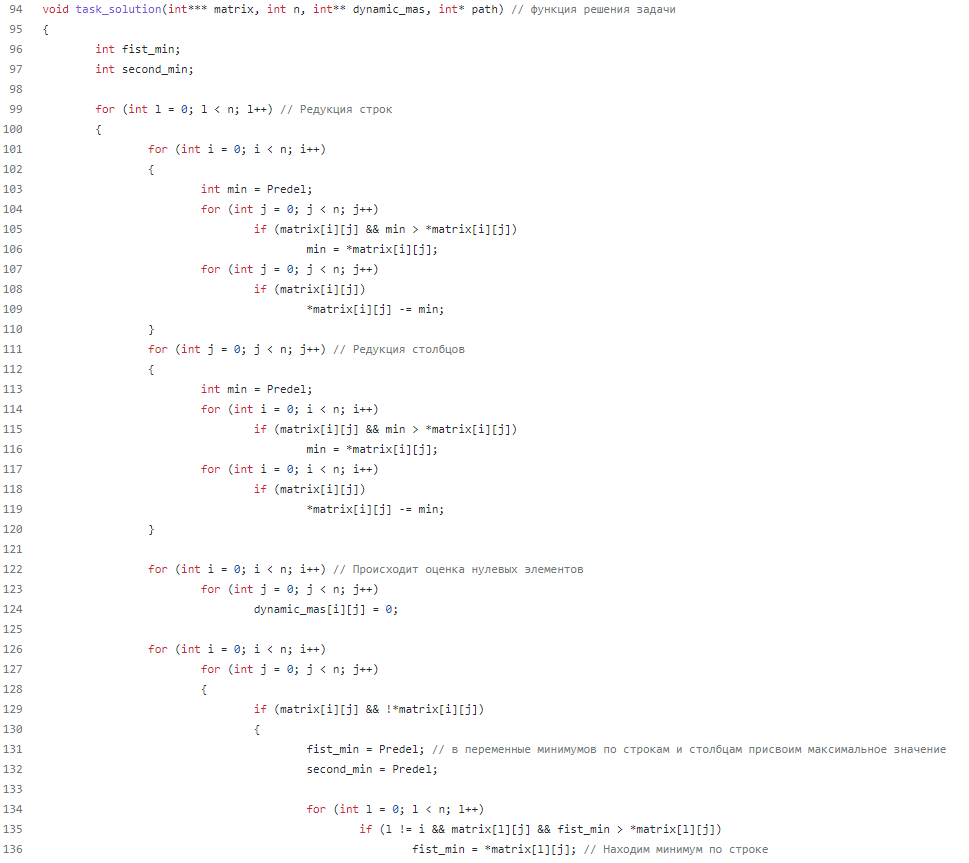


В данной строчке кода создаю объект класса Graph типа int. Через данный объект и будем обращаться к методам класса.

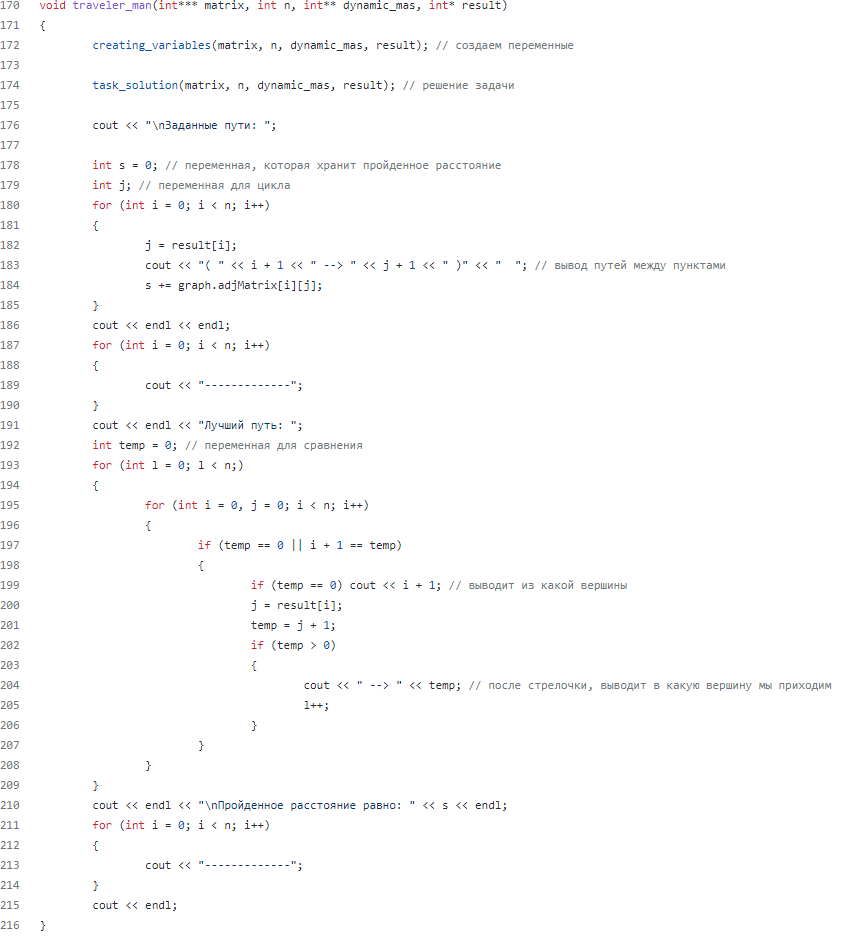


Приближаемся к функциям, которые решают данную задачу, данная функция занимается подготовкой переменных, массивов, для дальнейшей работы с другими функциями. Эта функция принимает: указатель на матрицу смежности по ссылке, количество пунктов по ссылке, указатель на динамический массив по ссылке, а также указатель на результат по ссылке. После происходят действия в циклах for. В первом for – происходит заполнение динамического массива от 0 до заданного количества пунктов. В следующем for, происходит обработка матрицы, а именно сначала выделяется память под переменную matrix, далее с помощью внутреннего цикла for – проверяется, если пересечение строки и столбца равны 0, значит мы должны указатель на данную ячейку сделать – nullptr.



Далее, проходя по важным функциям, разберем эту – функция task\_solution – также принимает параметры, как и первая, но последний указатель не результат, а на путь. Создаются переменные first и second – которые отвечают за минимум по строкам и по столбцам. Далее, как и написано в комментариях к коду идет редукция по строкам, то есть, через цикл for – происходит перебор строк с проверками на больше, меньше минимума (min), изначально переменной min – присвоено значение Predel – оно равно большому значению, которое никогда не будет достигнуто. В конце цикла происходит вычисление минимальных элементов. Аналогично происходит в случае с редукцией столбцов в последующих циклах. После нахождения минимального элемента и редукции строки и столбца, происходит оценка нулевых элементов, это значит, что если при пересечении строки и столбца, будет найден нулевой элемент, а он должен быть найден, так как в предыдущем шаге было вычитание, то данную строку и столбец убираем из рассмотрения, в коде это представлено в виде присваивания указателям – nullptr (начиная с 157 стр. кода).

В данной функции, которая также является важной, здесь происходит вызов функций и нахождение лучшего пути. Вызываются функции создания переменных и решения задачи, создаётся переменная s, она и будет хранить результат пройденного расстояния, переменная j – нужна для цикла, далее, с помощью цикла for – происходит вывод отрезков путей, которые мы ввели, после, также с помощью цикла for – происходит вывод лучшего пути и его длины.



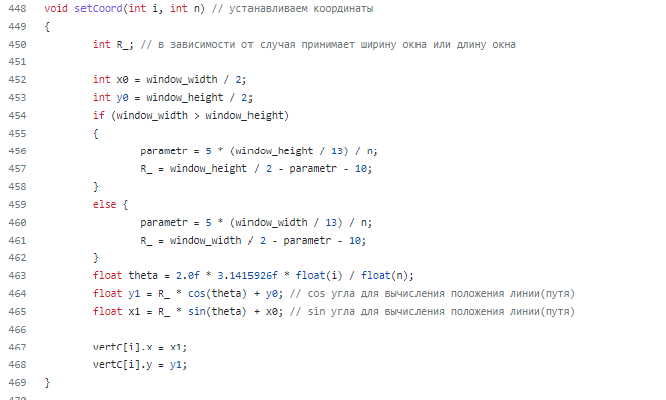
Далее в коде идет обработка действий всех методов.

В данном блоке кода происходит реализация интерфейса при создании графа:

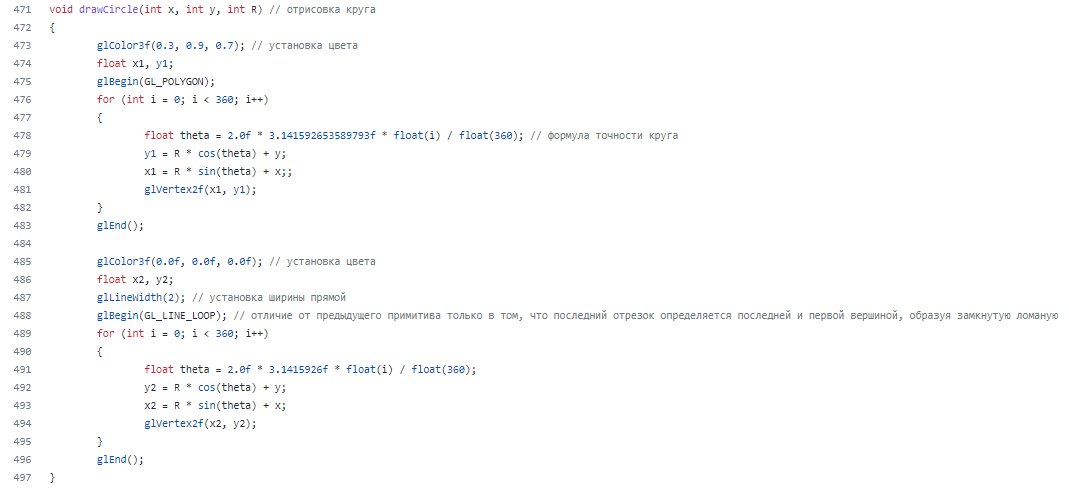
Сначала обработка создания путей и дорог, а также обработка протяженности дорог. Все пункты сопровождаются проверками. В конце возвращаем объект класса graph.



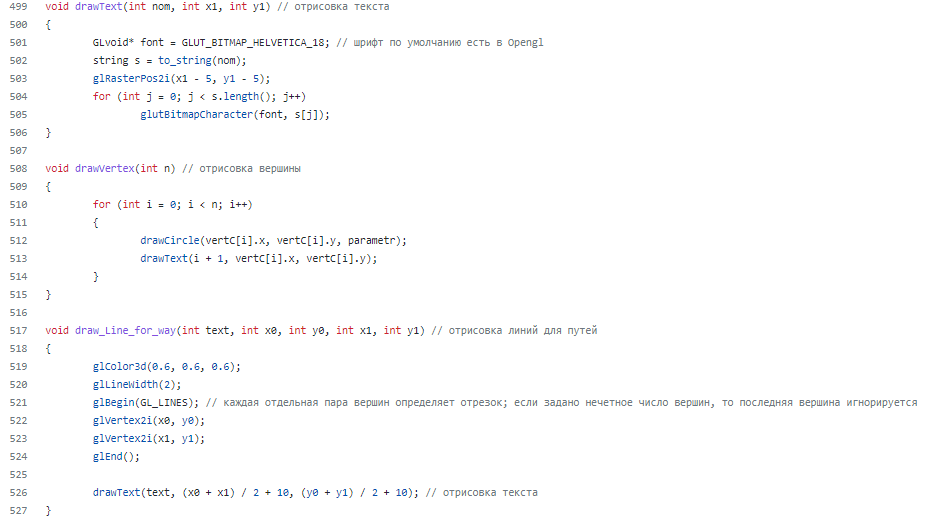
Функция, устанавливающая координаты для смещения окна, в параметр R\_ - после этого блока программы будет присвоено какое-то значение, которое нам понадобиться для отображения кругов. Также формула theta – для точности изображения, и формулы с (cos и sin), они нужны для правильного отображения вершин (кругов) – относительно друг друга.



Функция OpenGl – изображение круга. Также применяется формула для точности краев круга, а также формулы углов. Здесь же задаем цвет – glColor3f, далее происходит обработка прямых – которые соединяют круги. Для них также высчитывается точность и углы – с помощью которых происходит направление прямой.

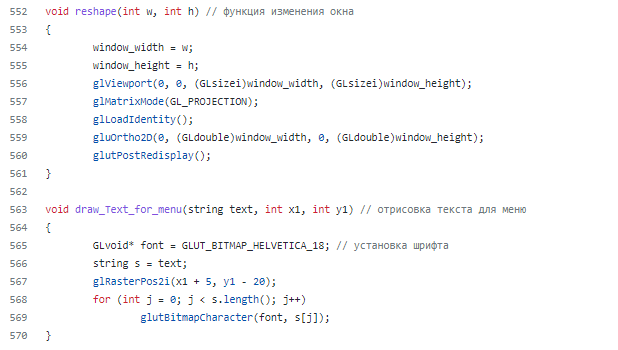


Здесь показаны 3 функции, они отвечают за изображение текста, изображение кругов (через обращение к функции, которая создает круг), и изображение линий для путей.

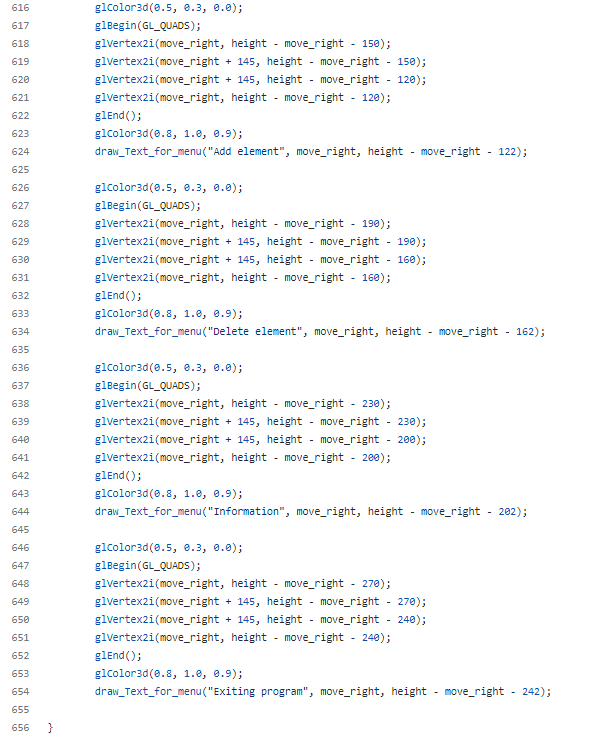
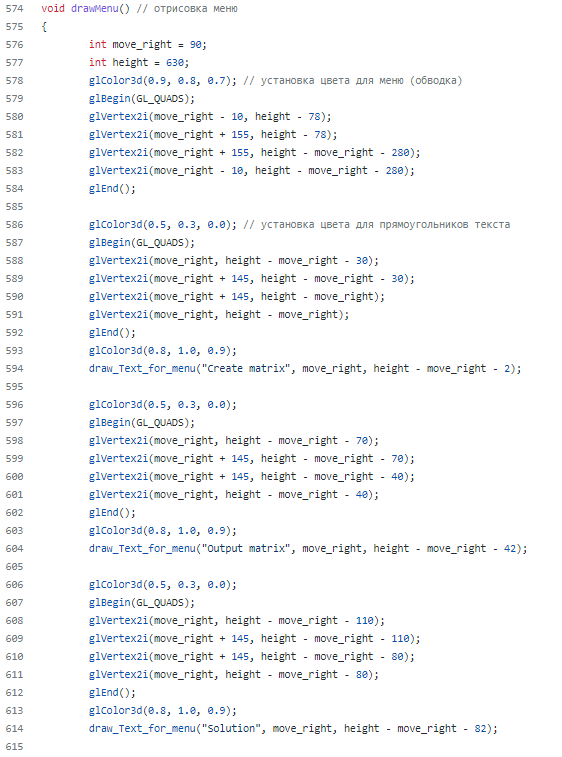


Функция reshape – принимает параметры высоту и ширину окна. Эта функция примечательна тем, что при изменении окна, граф остается в поле видимости пользователя.

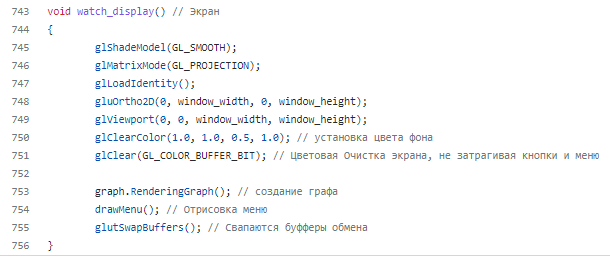
Далее, функция изображения текста в меню, шрифт взял по умолчанию. Создается переменная строкового типа данных – s, в которую и присваивается наш text.



В данном блоке кода происходит обработка «меню», изображение заднего плана «меню», а уже после изображение панелей «кнопок» на этом самом «меню».



Далее в кода идет функция реализации кнопок, и функция отображения на экран:



В main () по стандарту вызываются основные функции, которые в свою очередь будут вызывать функции из основной части кода. Также в main настраивается размер окна.

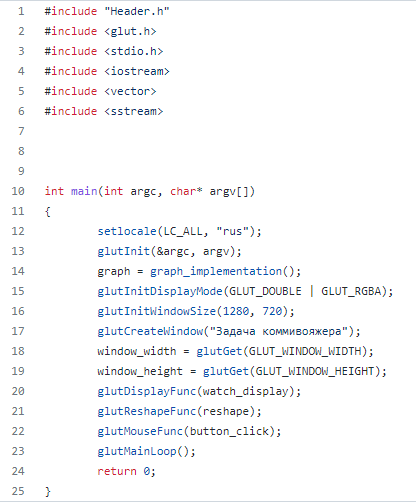
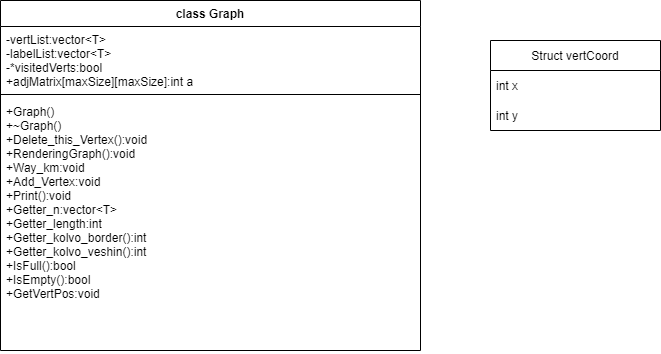
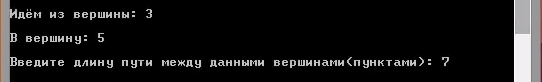
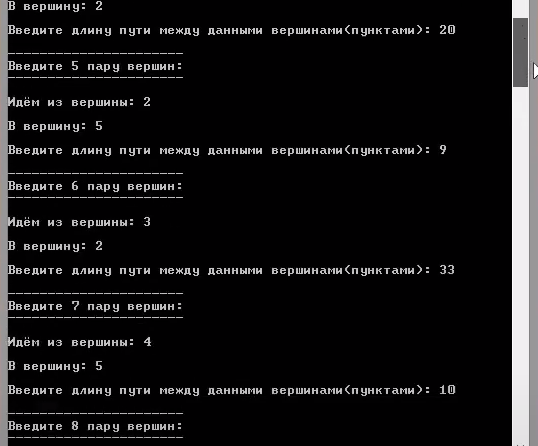
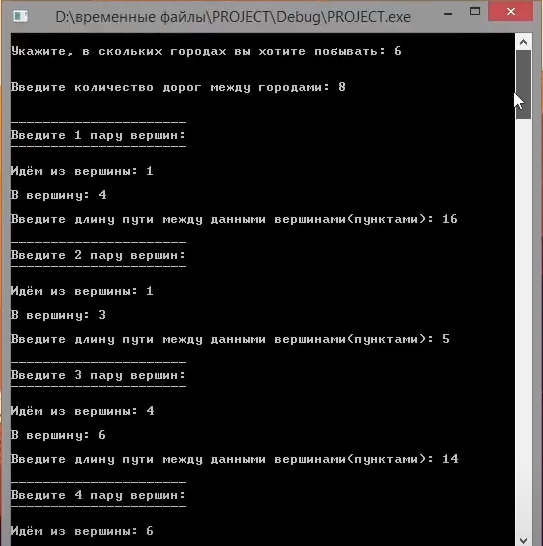
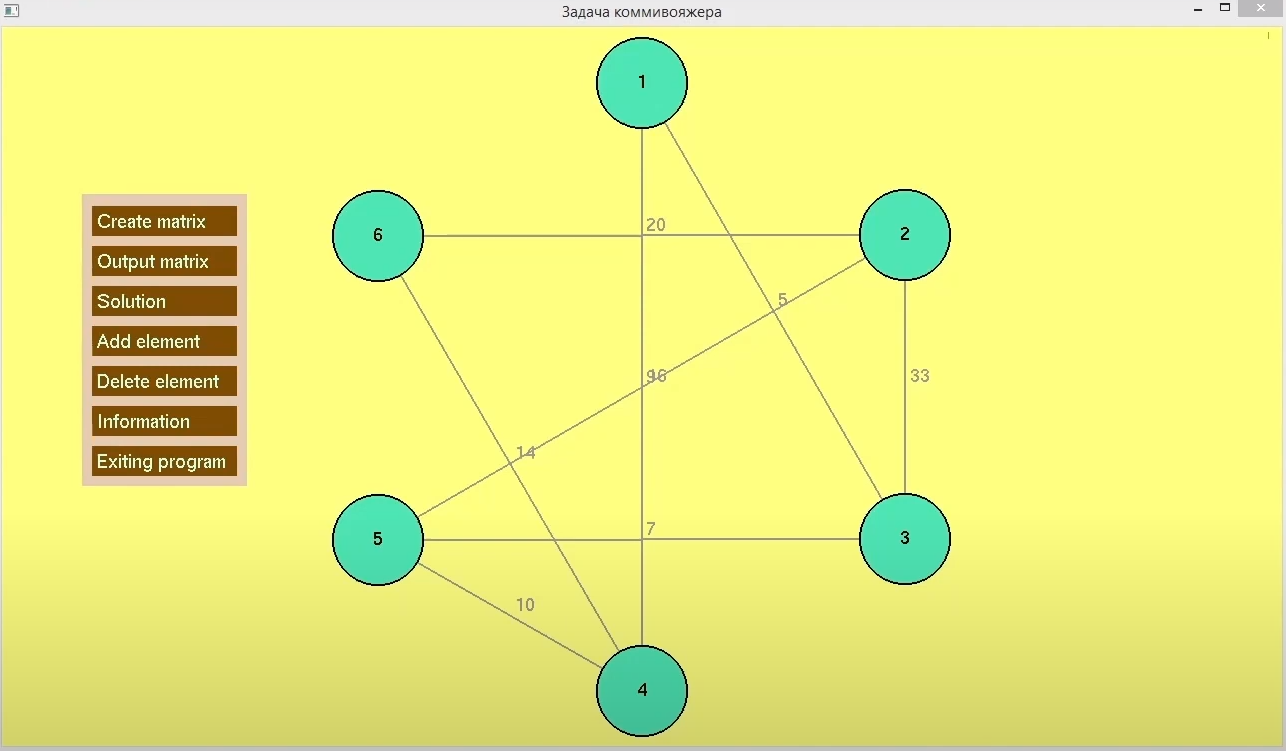


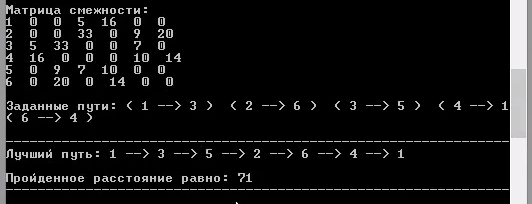
Диаграмма классов:

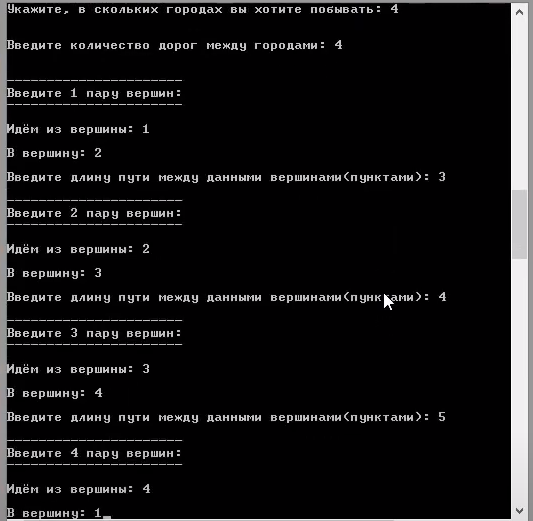


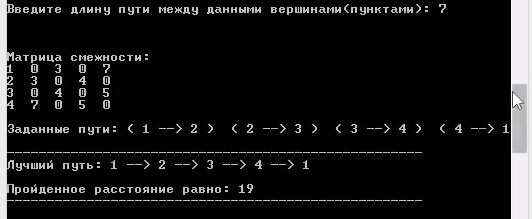
Пример работы (данные и визуализация):

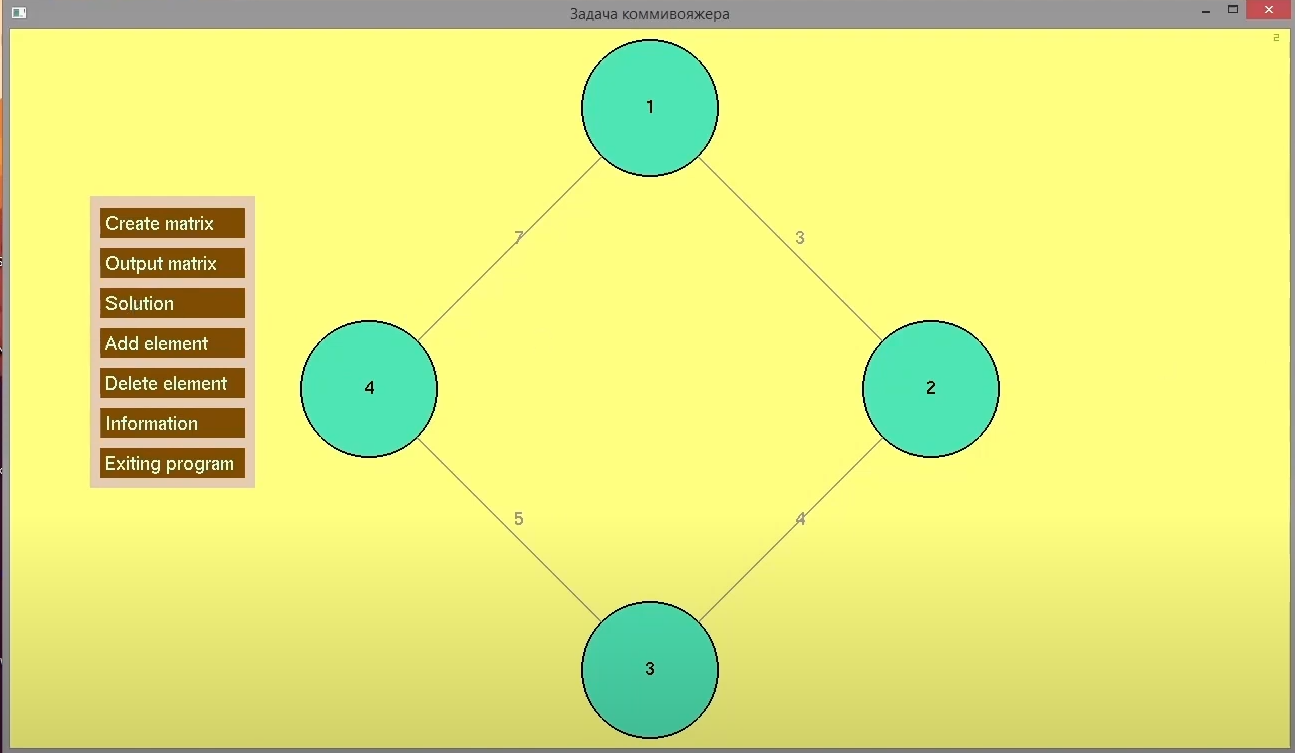


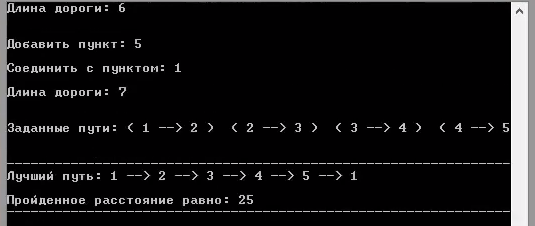
Изображение после ввода данных:

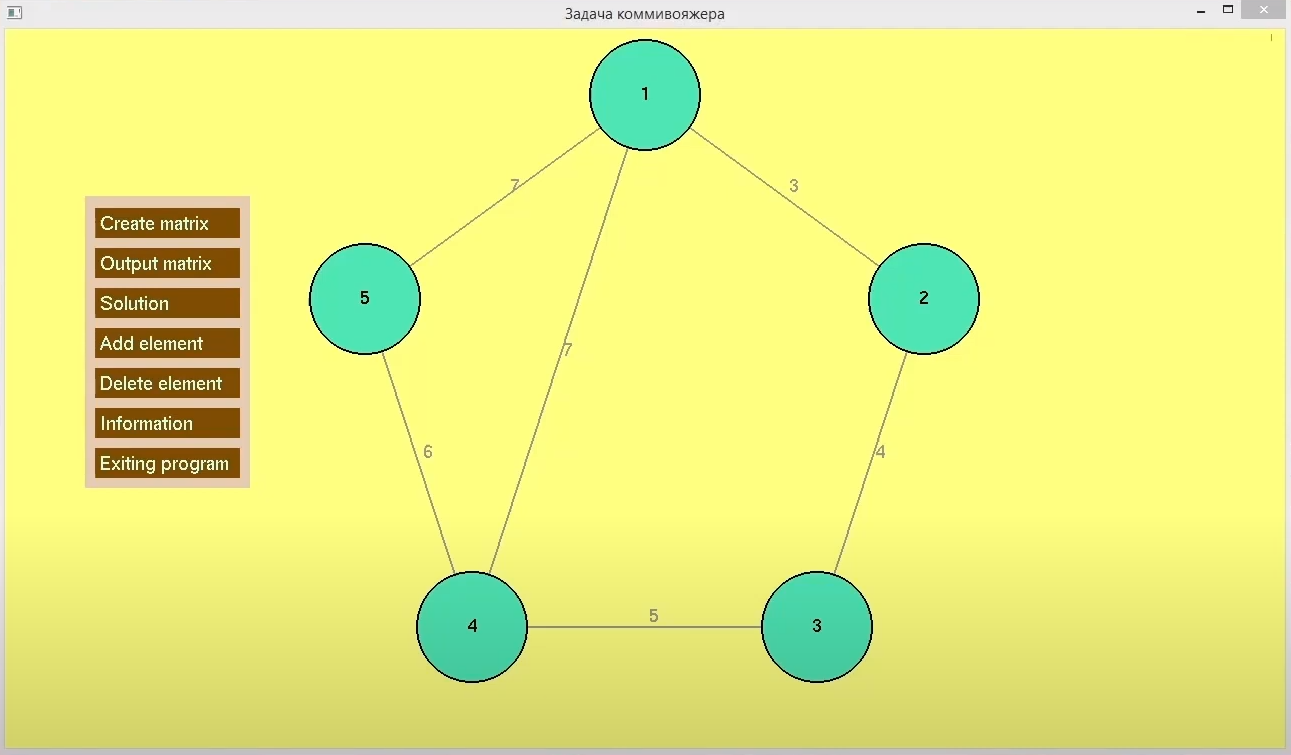
Матрица смежности для данного графа и длина кратчайшего расстояния:

Создаем новую матрицу через соответствующую кнопку в «меню»:

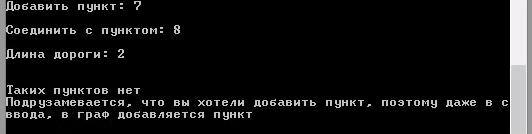
После ввода данных новой матрицы – выводим ее матрицу и результат:

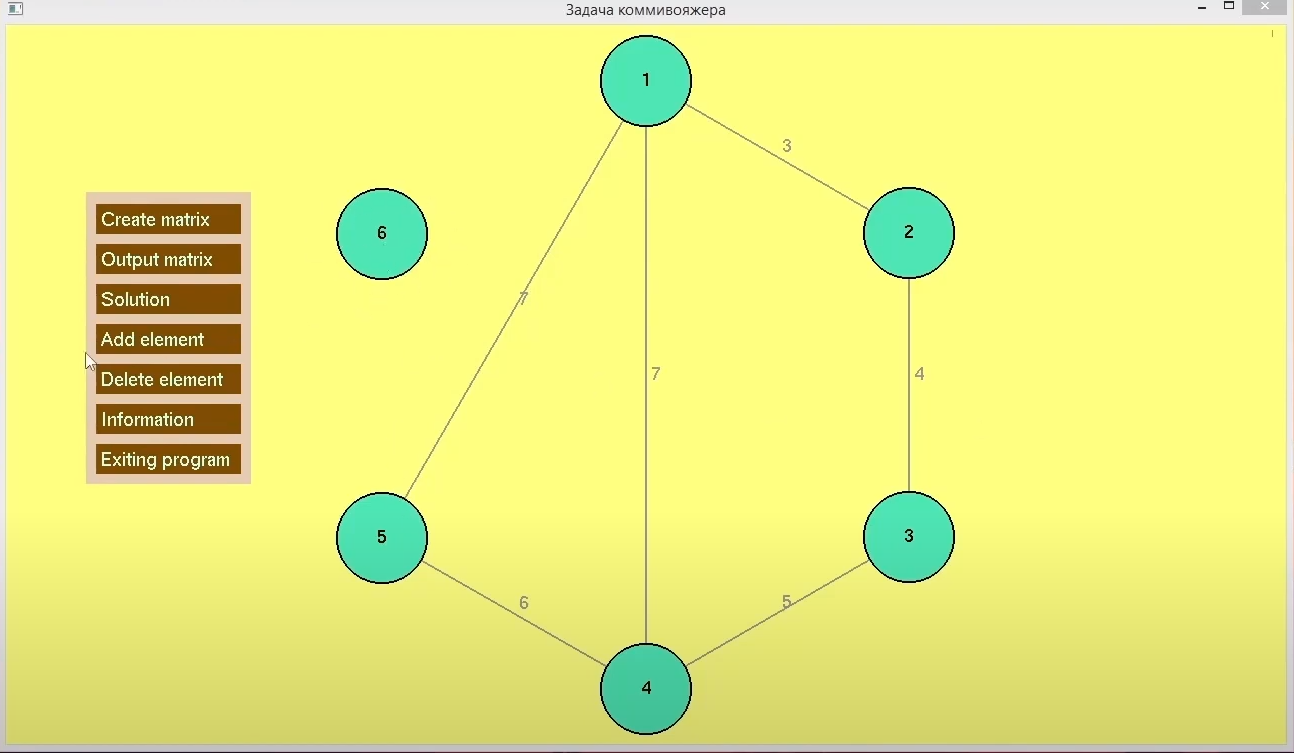
Вот граф для данной матрицы:

Добавим пункт в граф и покажем решение:

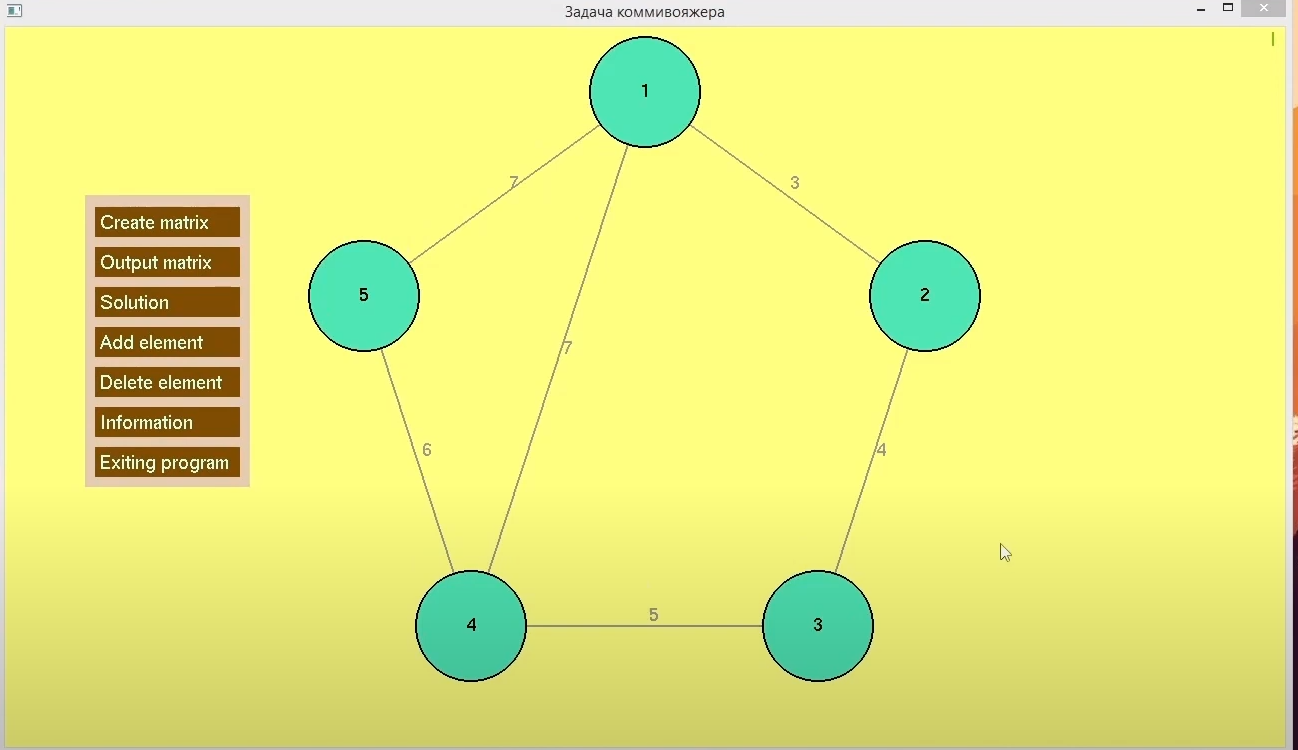
После добавления вершины:

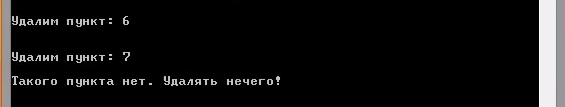
Добавим пункт, которого нет, после удалим дополнительный пункт:





Удалился:



Выдает ошибку, если удалять не существующий путь:

Кнопка информация – выдает данные обо мне:



------------------------------------------

Ссылка на GitHub: [GitHub](https://github.com/BaldinAlexey/Tvorcheskoe_zadanie_2022)

Ссылка на видео: [Видео на ютубе](https://www.youtube.com/watch?v=wiQh8Y7ZUk0)

------------------------------------------

**Описание решения задачи калькулятор**

1. В калькуляторе нет определенного метода решения
2. Написание кода было выполнено на языке программирование «С++»

Также для визуализации идеи воспользовался программой «Windows Forms», работал с данной программой, потому что она проста в понимании, у неё удобный интерфейс, и по ней есть множество обучающих видео.

1. a) Возникли трудности с работой в самой программе (Windows Forms) – так как не разу не работал в подобных программах, поэтому сначала 2-дня осваивался. Позже, понял какие блоки за что отвечают.

б) Также возникли трудности в реализации проверок арифметических действий, так как первый раз столкнулся с таким количеством кода, и, соответственно, начал путаться в коде, но все-же после просмотра всего кода, разобрался в нём.

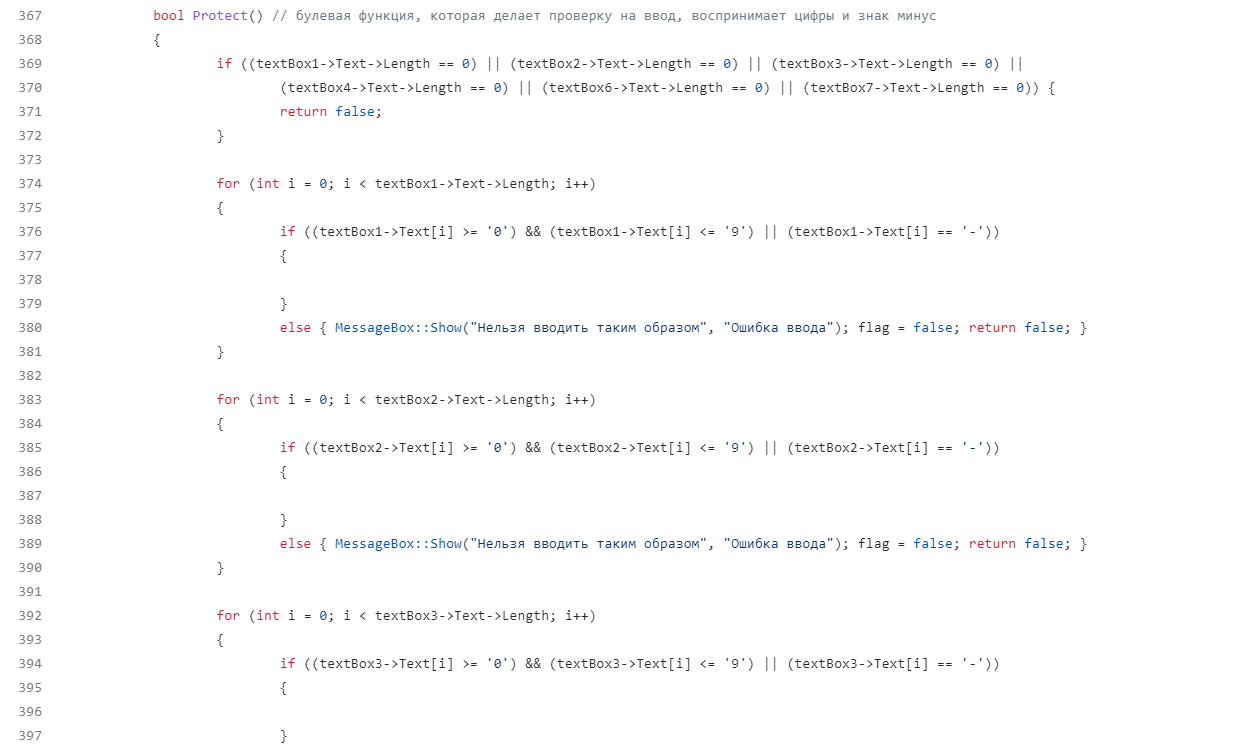
1. Автор гордиться тем, что интерфейс калькулятора сделан очень понятно, любой может зайти в приложение и понять куда какие данные нужно ввести. Также можно отметить, что калькулятор выполнен в стиле минимализм – ничего лишнего, только то, что нужно для калькулятора и дополнительная информация. Считаю, что калькулятор не должен быть излишку цветным и громоздким – ведь это приложение нужно для быстрого получения ответа, собственно, поэтому я и реализовал его таким образом. Нельзя не отметить многочисленные проверки на случаи неправильного ввода и проверки на все случаи дробей – поэтому могу сказать, что мой калькулятор считает абсолютно все случаи дробей как с целой частью, так и без неё. Также автор гордиться дополнительным функционалом программы, а именно информацией об авторе приложения, информацией о самой программе, и проверка по выходу на крестик.

**Описание основных моментов в коде**

Разберу основные моменты кода, которые писал лично я, так как в приложении Windows Forms, когда создаешь кнопку, и хочешь написать ее реализацию, нужно нажать на нее 2-раза, и сразу же в коде появляется нужный блок кода, который реализует ее обработку.

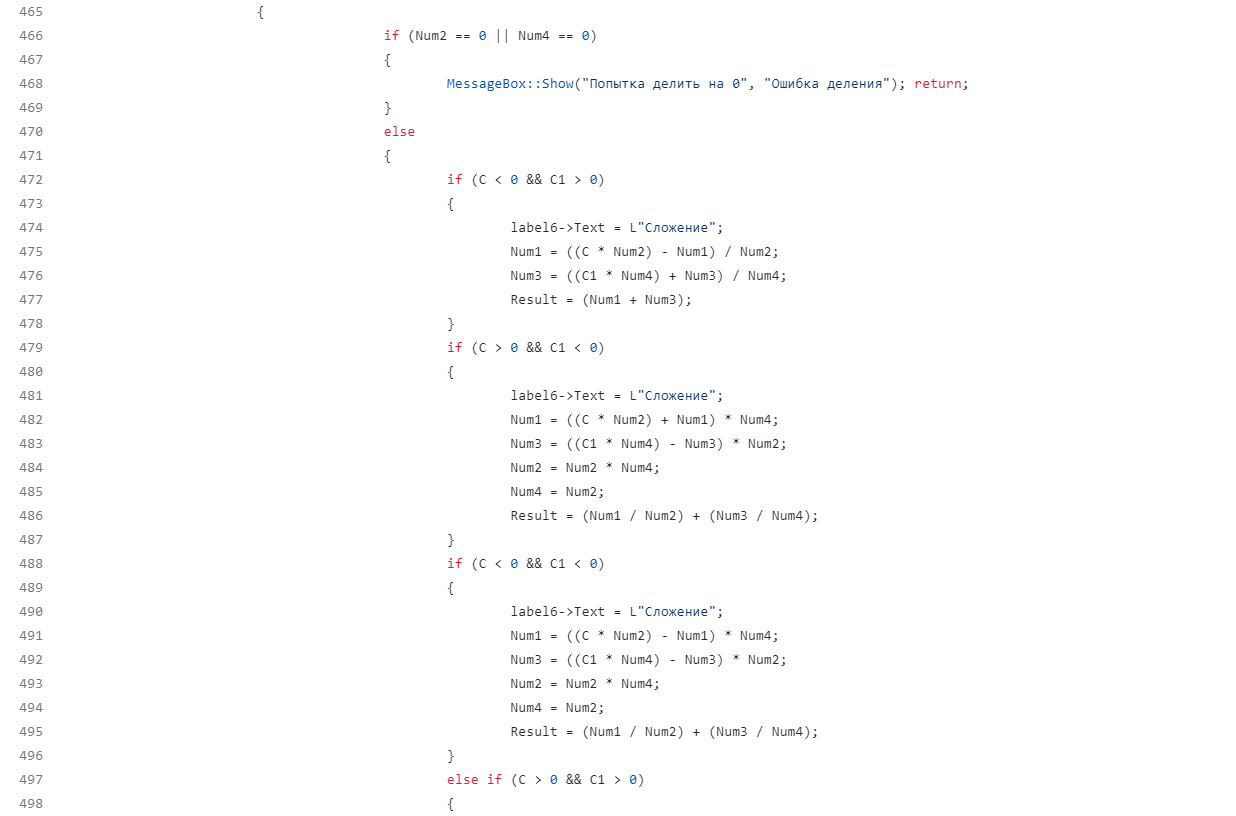
Здесь создаю булевою переменную, которая будет находиться в цикле, и, если этот цикл будет выдавать ошибку, соответственно я данную переменную делаю – false, и после на экран выводиться окно с ошибкой. То есть переменная предназначена для обработки исключительных ситуаций.

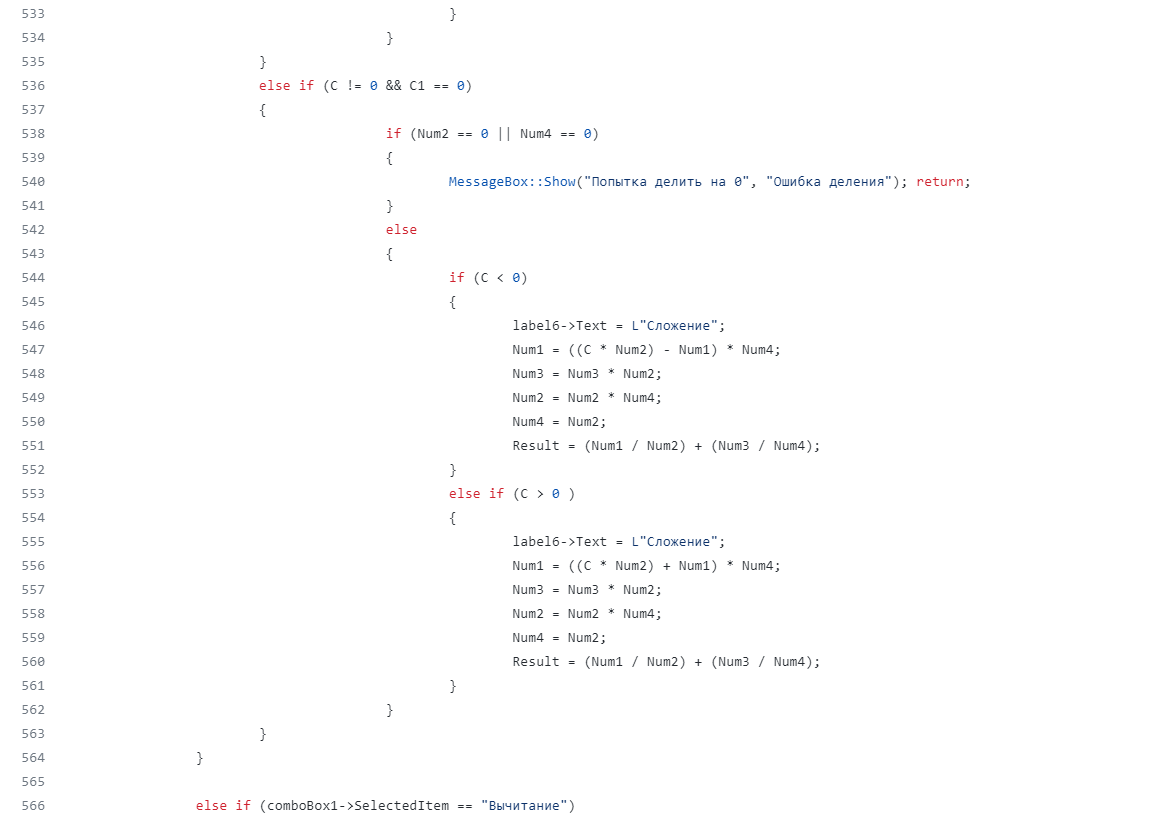
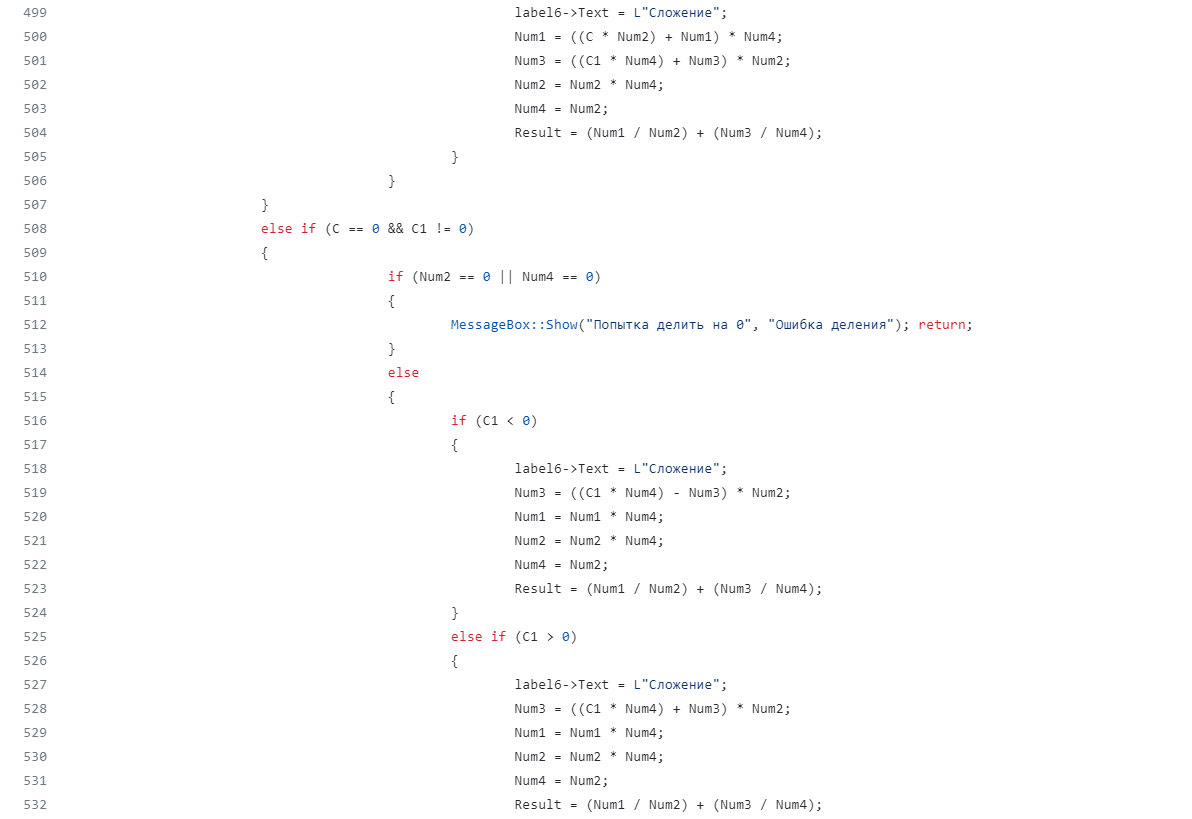


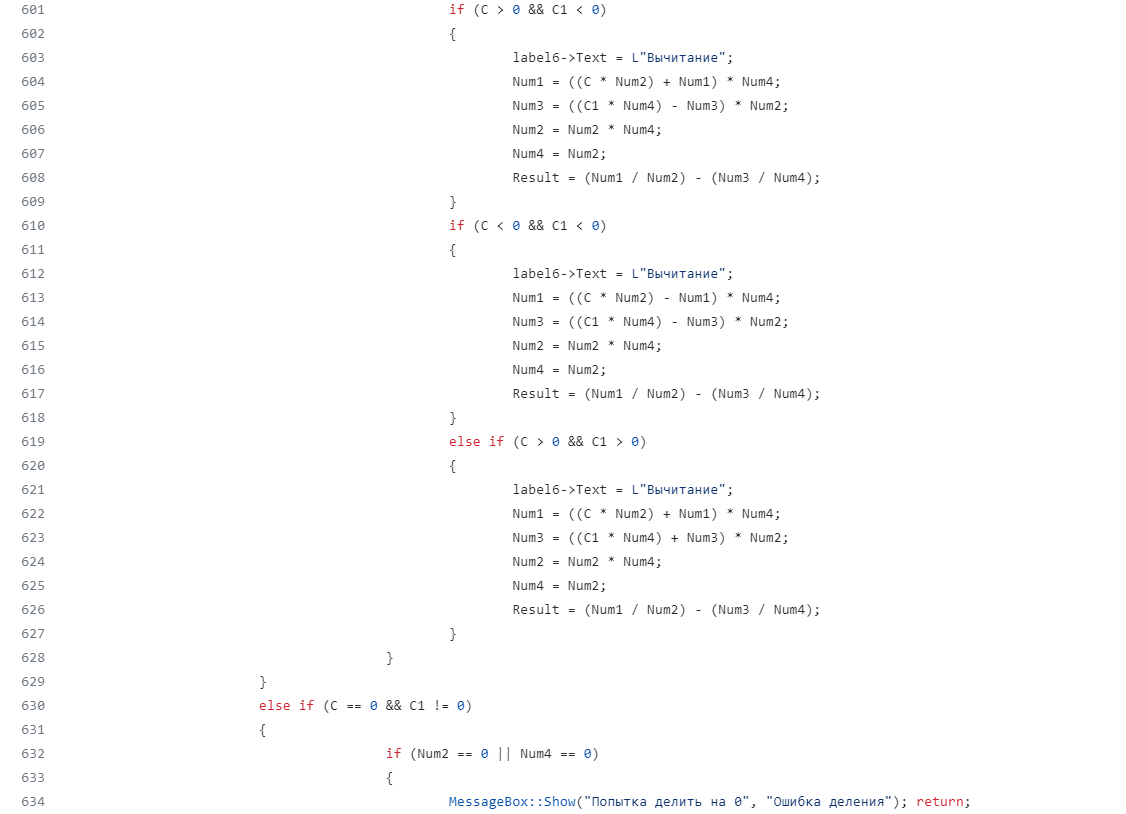
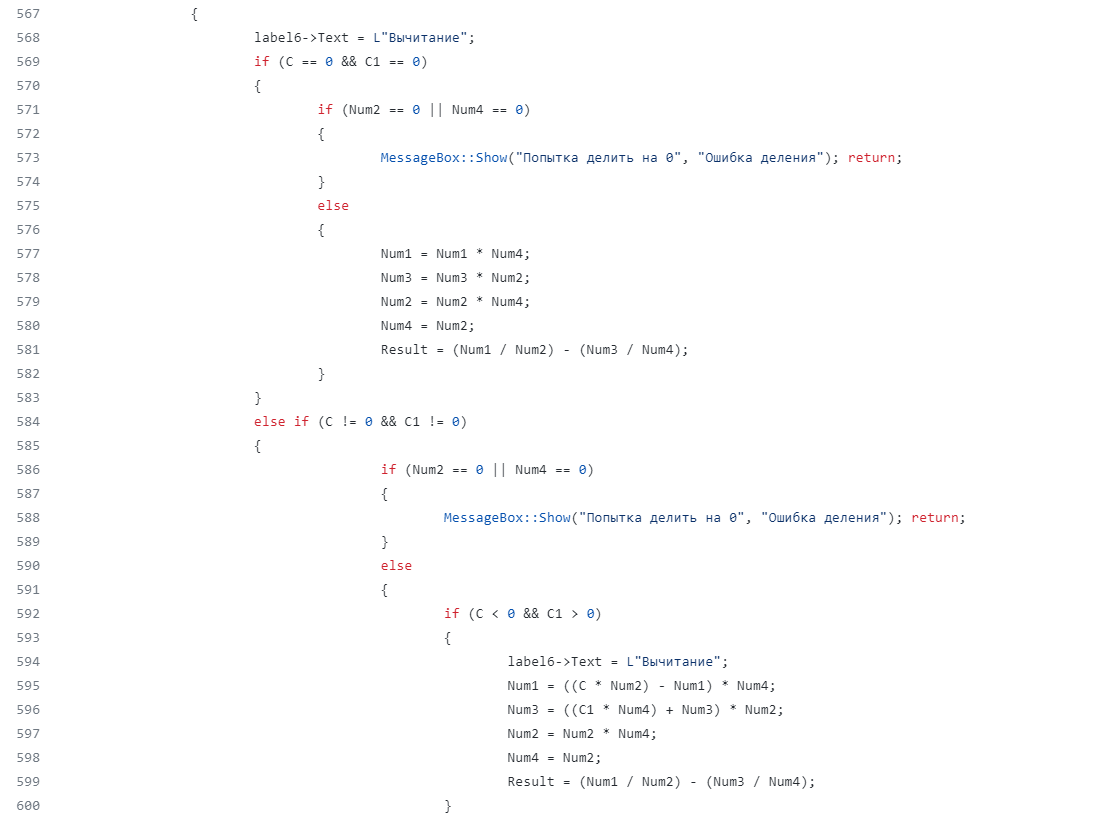
В данном блоке кода представлена функция Protect – также является булевой, она предназначена для проверки введенных значений, а именно можно вводить цифры и знак минус, все остальное будет выводить ошибку на экран.

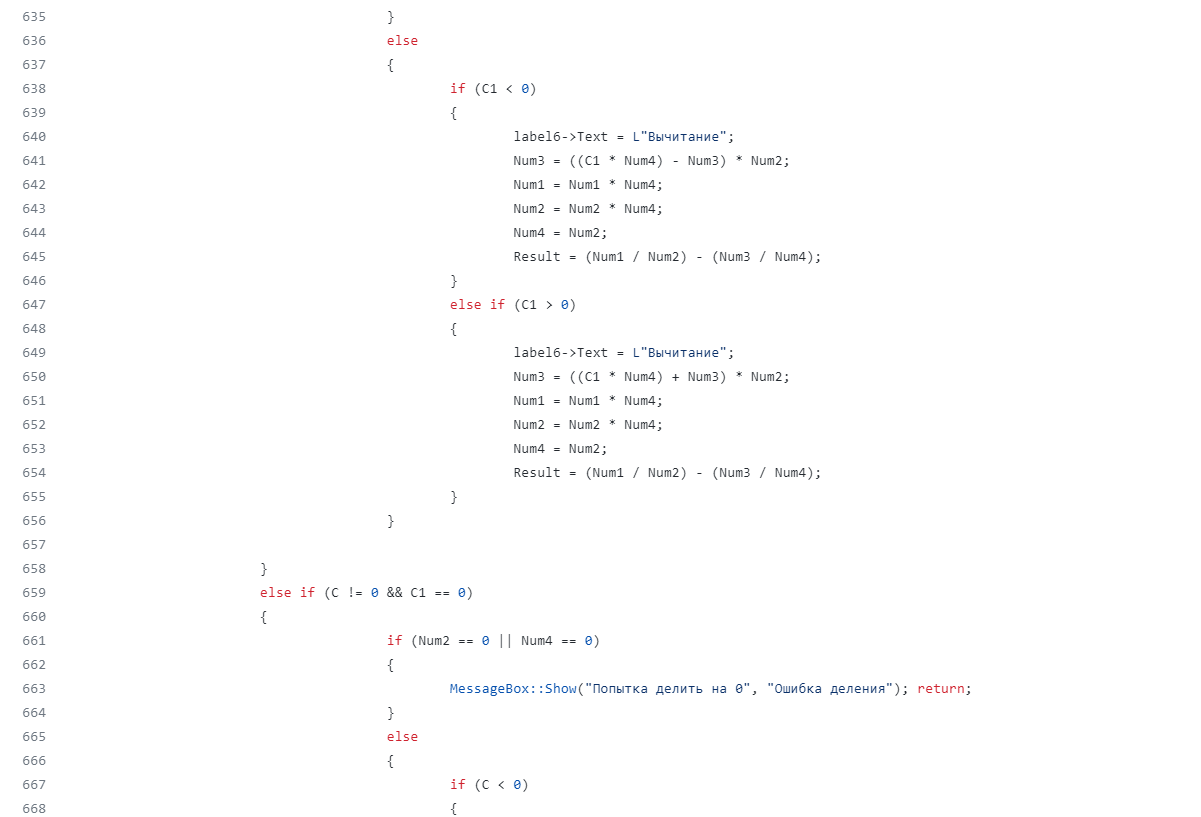


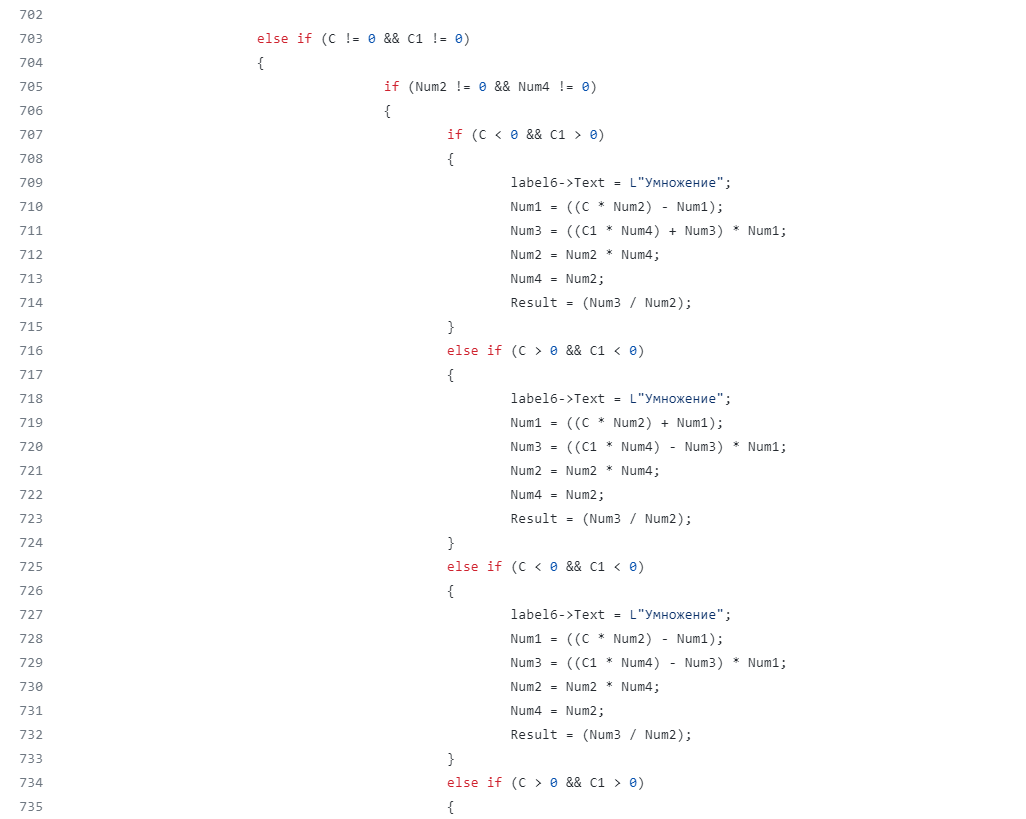
В данном большом блоке кода находятся те самые многочисленные проверки на корректность выполнения арифметических действий, конечно, действия производятся по всем правилам обыкновенных и смешанных дробей. Итак, через проверку Protect, создаются переменные Num1, Num2, Num3, Num4 – это наши дроби, далее создаются переменные С, С1 – это наши целые части, и переменная Result – которая будет хранить в себе результат, переменные chisl, znam – ничего не делают в моей программе, эти переменные случайно остались в данном блоке кода. Далее, происходит проверки на все случаи дробей со сложением, после, идут проверки на дроби с вычитанием, далее, проверки происходят на умножение и деление, конечно, во всех проверках присутствуют исключительные ситуации деление на 0, или неполный ввод данных.

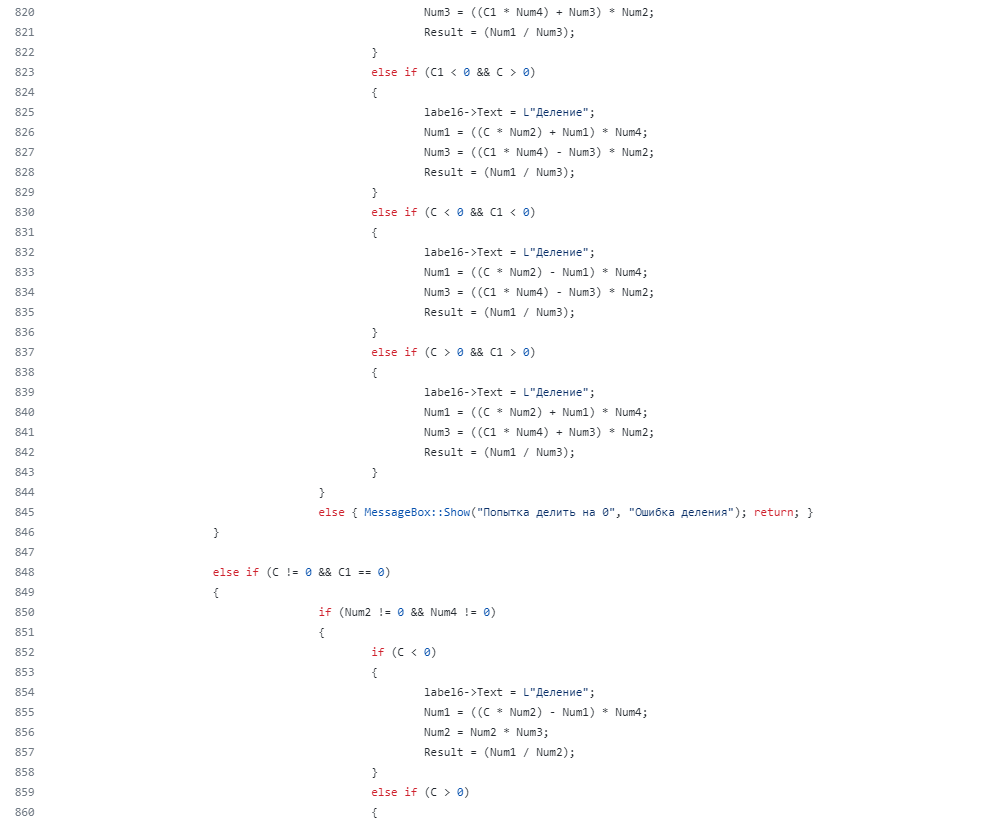






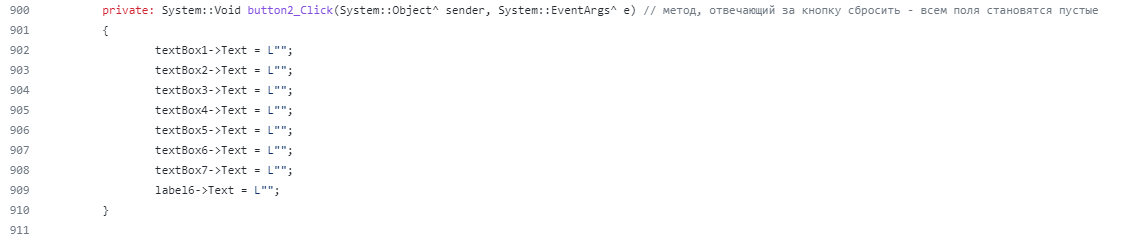






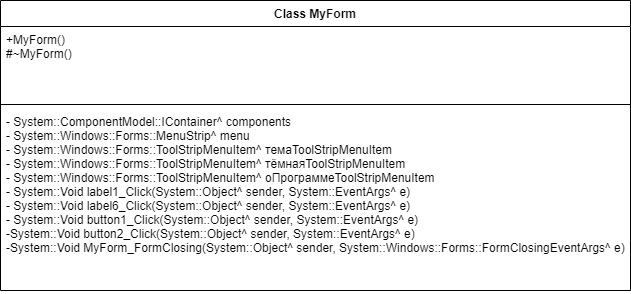
Данный метод – э то реализация кнопки сбросить, то есть все поля сбрасываются на пустоту:



Также методы, реализующие такие кнопки как: Справка, О программе, и выход по крестику:

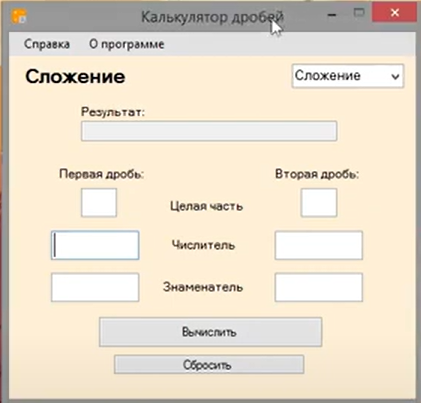


Диаграмма классов:



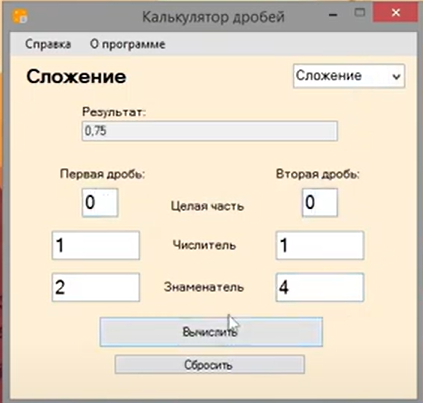
Пример работы (данные и визуализация):

Вот так выглядит само приложение калькулятор:

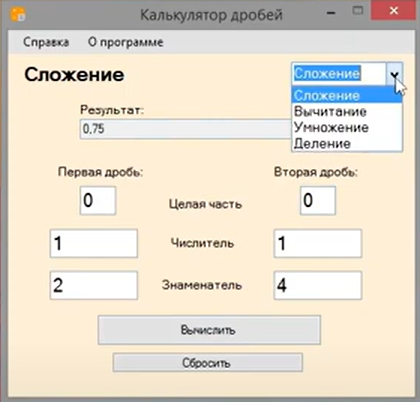


Далее, сделаю проверку арифметических действий.

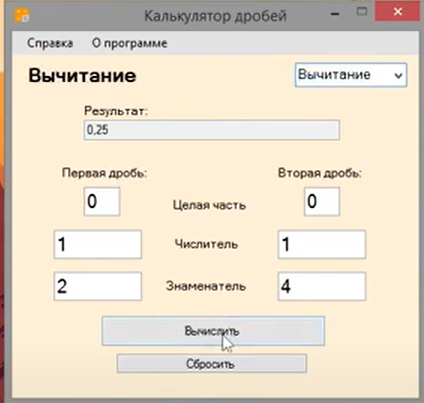
Проверили действие - сложение



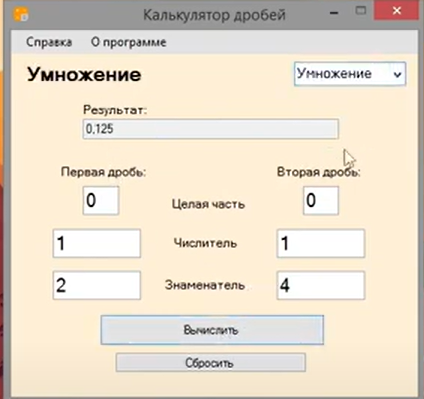
Выбирать действия можно через данный блок «меню»



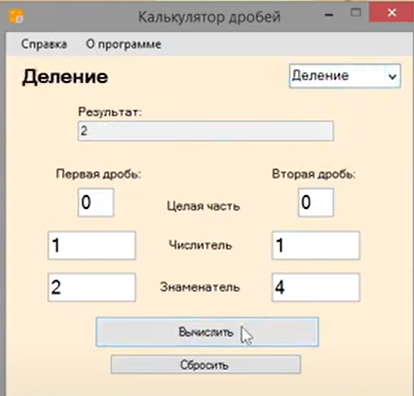
Теперь проверили действие – вычитание. Хочу отметить, что при выборе того или иного пункта меню, меняется надпись выбранного действия:



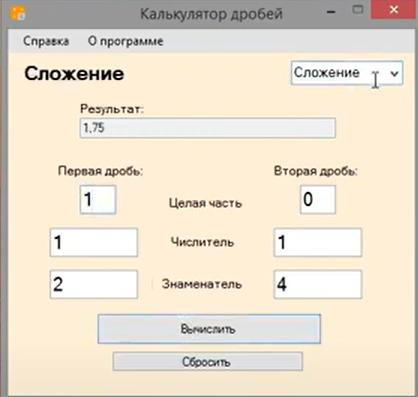
Проверили действие - умножение



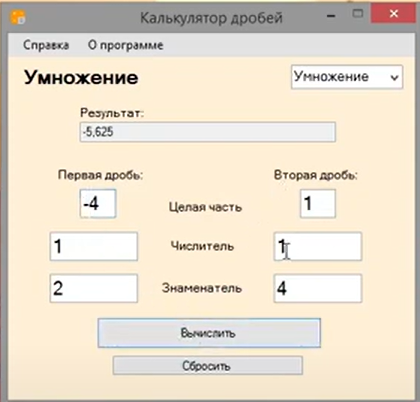
Проверили действие - деление



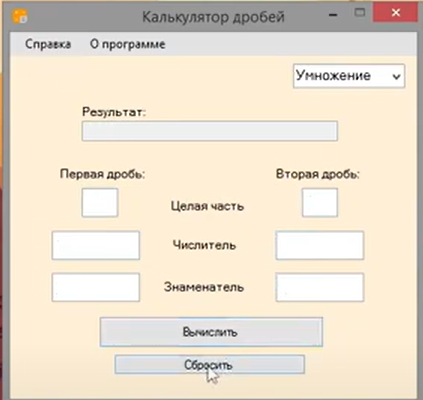
Проверка работы с целой частью



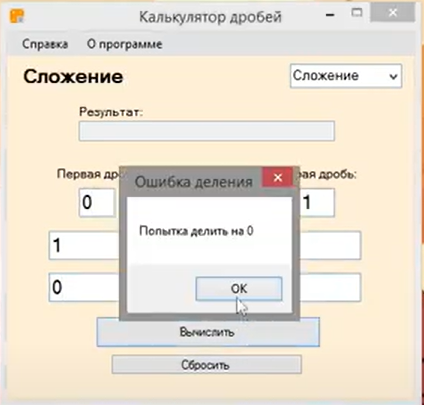
У обеих есть целая часть:



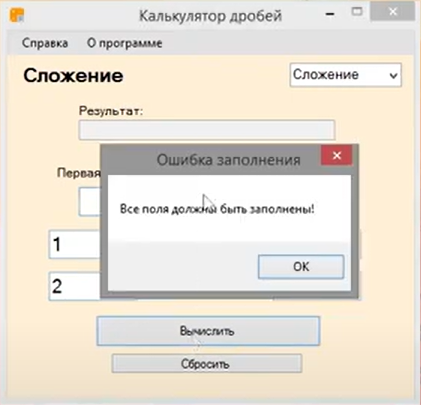
Проверка кнопки - сбросить



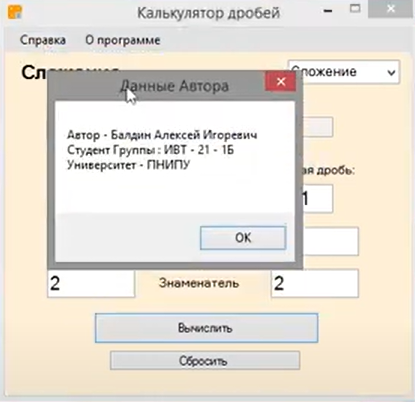
Проверка деления на 0:



Проверка на пустые поля:

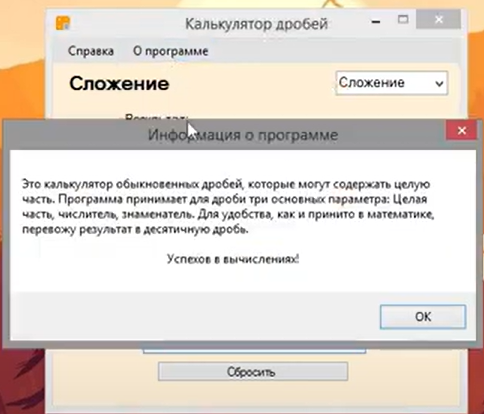


Проверка кнопки - Справка

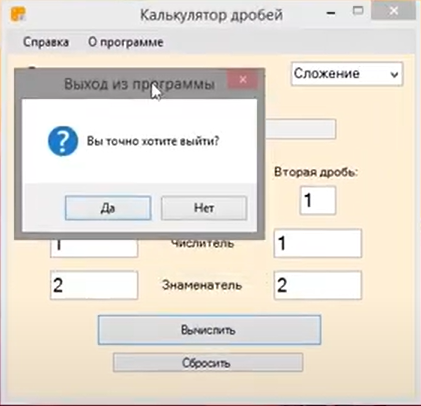


Проверка кнопки – О программе

Здесь и объясняется, почему я вывожу результат в десятичной дроби, потому что принято в математике после работы с обыкновенными дробями – переводят для конкретного результат в десятичную дробь:



Проверка на выход по крестику:



**Дублирую ссылки на Видео и GitHub:**

------------------------------------------

Ссылка на GitHub: [GitHub](https://github.com/BaldinAlexey/Tvorcheskoe_zadanie_2022)

Ссылка на видео: [Видео на ютубе](https://www.youtube.com/watch?v=wiQh8Y7ZUk0)

------------------------------------------