Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Пермский национальный** **исследовательский политехнический университет»**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

направление подготовки: 09.03.01– «Информатика и вычислительная техника»

**Творческая работа**

по дисциплине

**«Теория алгоритмов и структуры данных»**

Семестр 2

Выполнил студент гр. ИВТ-21-1б

Русинов Александр Станиславович

Проверил:

ст. пр. каф. ИТАС

Яруллин Денис Владимирович

(оценка) (подпись)

(дата)

Пермь 2021

**Постановка задачи**

Создать приложения калькулятора по нахождению коэффициентов системы уравнений и по решению задачи коммивояжера в соответствие с вариантами.

**Описание решения задачи коммивояжёра**

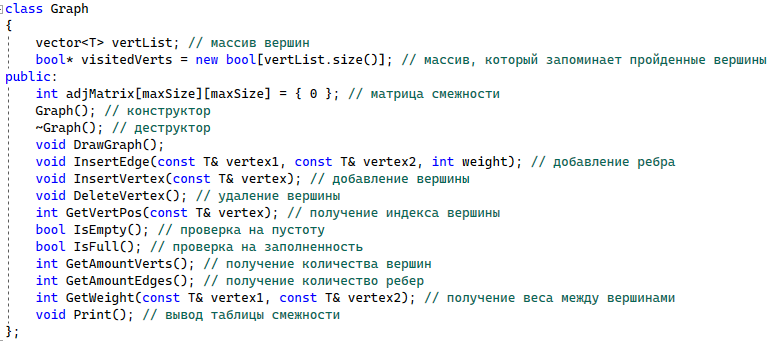
1. Для задачи коммивояжёра был применен метод «ветвей и границ», так как этого требует условие.
2. Написание кода было выполнено на языке программирования «С++»

визуализации идеи использована платформа «OpenGl». Данная платформа выбрана автором из-за доступности обучающего материала в сети интернет.

**Описание основных частей в коде**

**Схема решения**

На рисунке 1 представлен класс Graph – в public поле методы, которые позволяют производить действия над графом.



Русинок 1 – класс Graph

На рисунке 2 представлен функция, которая 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.

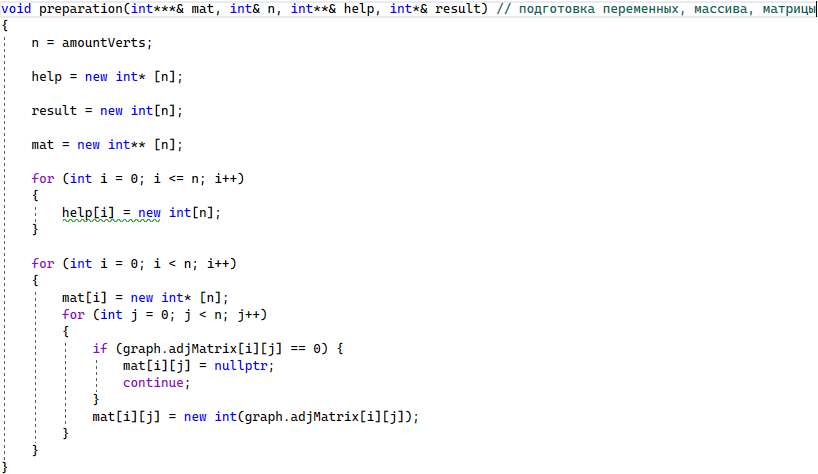


Рисунок 2 – функция preparation

На рисунке 3 функция answer – так же принимает параметры, как и первая, но последний указатель не результат, а на путь. Далее идет редукция по строкам, то есть, через цикл for – происходит перебор строк с проверками на больше, меньше минимума (min. 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.

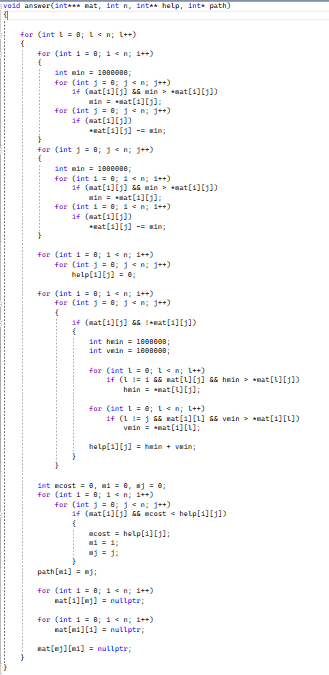


Рисунок 3 – функция answer

На рисунке 4 представлена функция tsalesman. В данной функции 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.

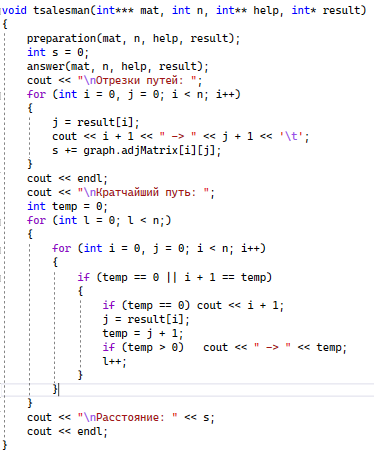


Рисунок 4 – функция tsalesman

На рисунках 4, 5, 6 представлена реализация графа. Сначала обработка создания путей и дорог, а также обработка протяженности дорог. В конце возвращаем объект класса graph (рис. 4). Функция (рис. 5), 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 (рис. 6). На рисунке 7 показаны 3 функции, они отвечают за изображение текста, изображение кругов (через обращение к функции, которая создает круг), и изображение линий для путей.

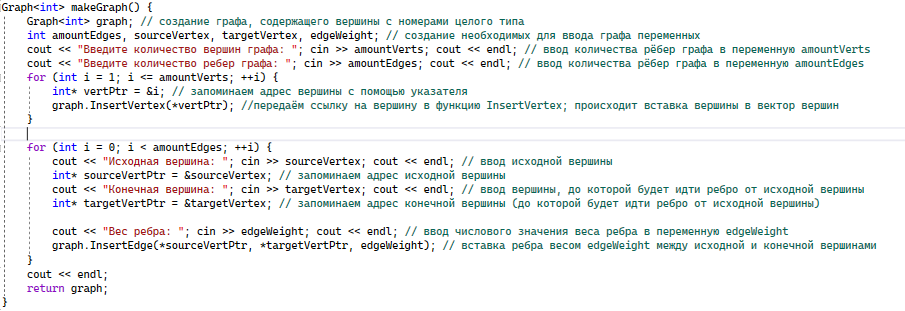


Рисунок 4

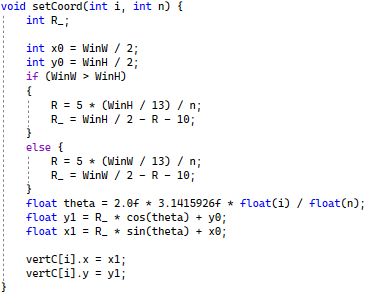


Рисунок 5

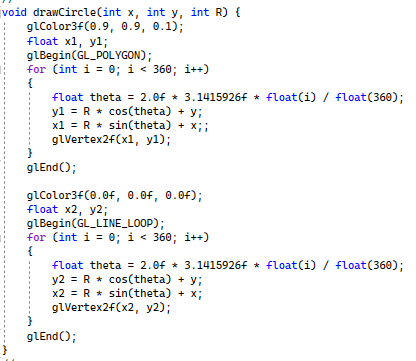


Рисунок 6

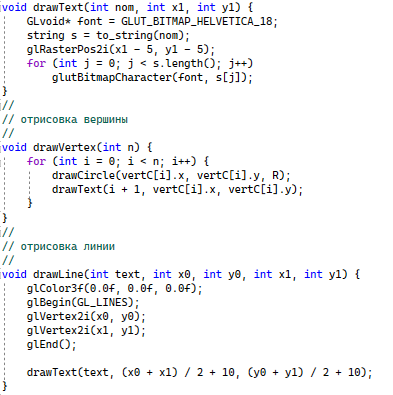


Рисунок 7

Ну рисунке 8 представлена 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.

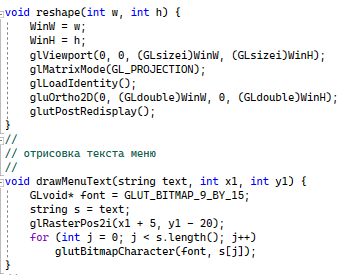


Рисунок 8

В данном блоке кода происходит обработка «меню», изображение заднего плана «меню», а уже после изображение панелей кнопок на меню (рис. 9).

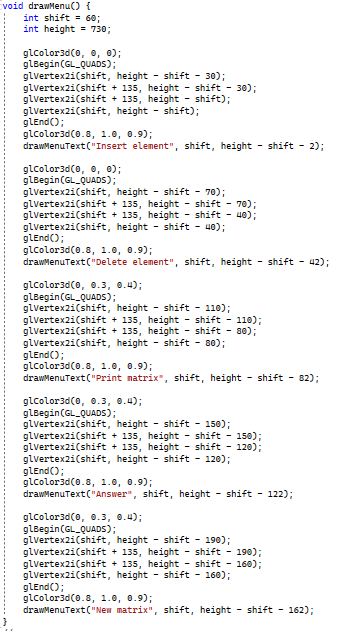


Рисунок 9

Функция отображения на экран (рис. 10).

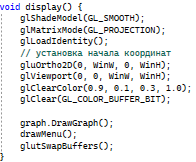


Рисунок 10

В main () по стандарту вызываются основные функции, которые в свою очередь будут вызывать функции из основной части кода. Также в main настраивается размер окна (рис. 11).

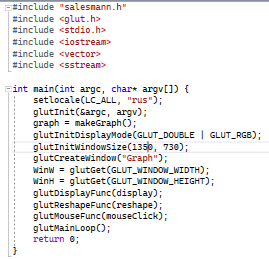


Рисунок 11

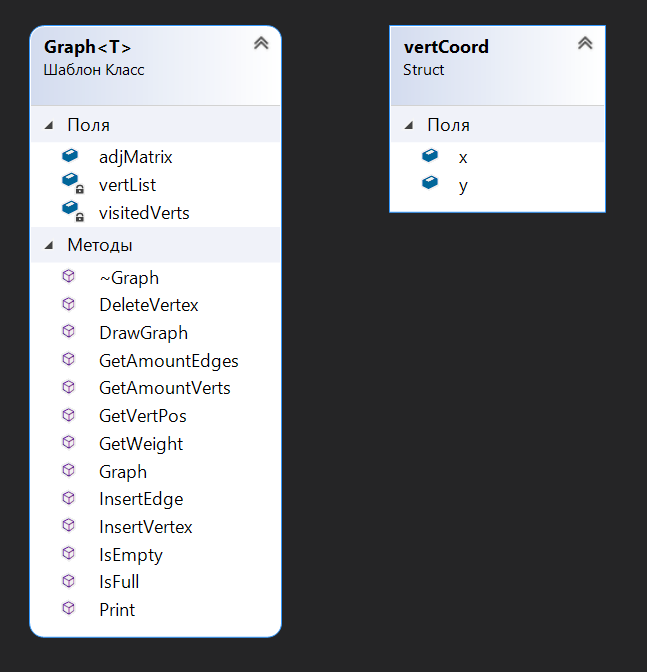
**Диаграмма классов (рис. 12)**

Рисунок 12

**Пример работы и результаты выполнения (рис. 13, 14, 15, 16)**



Рисунок 13

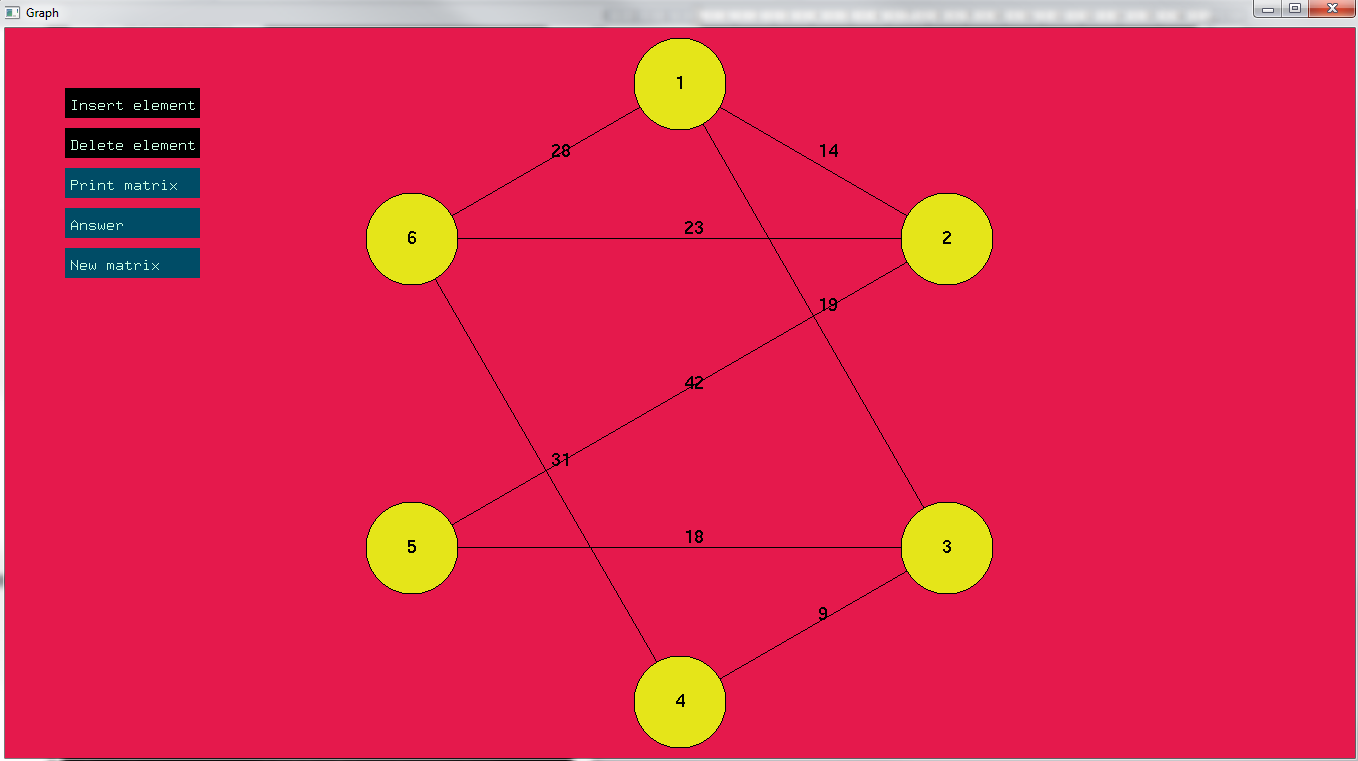


Рисунок 14

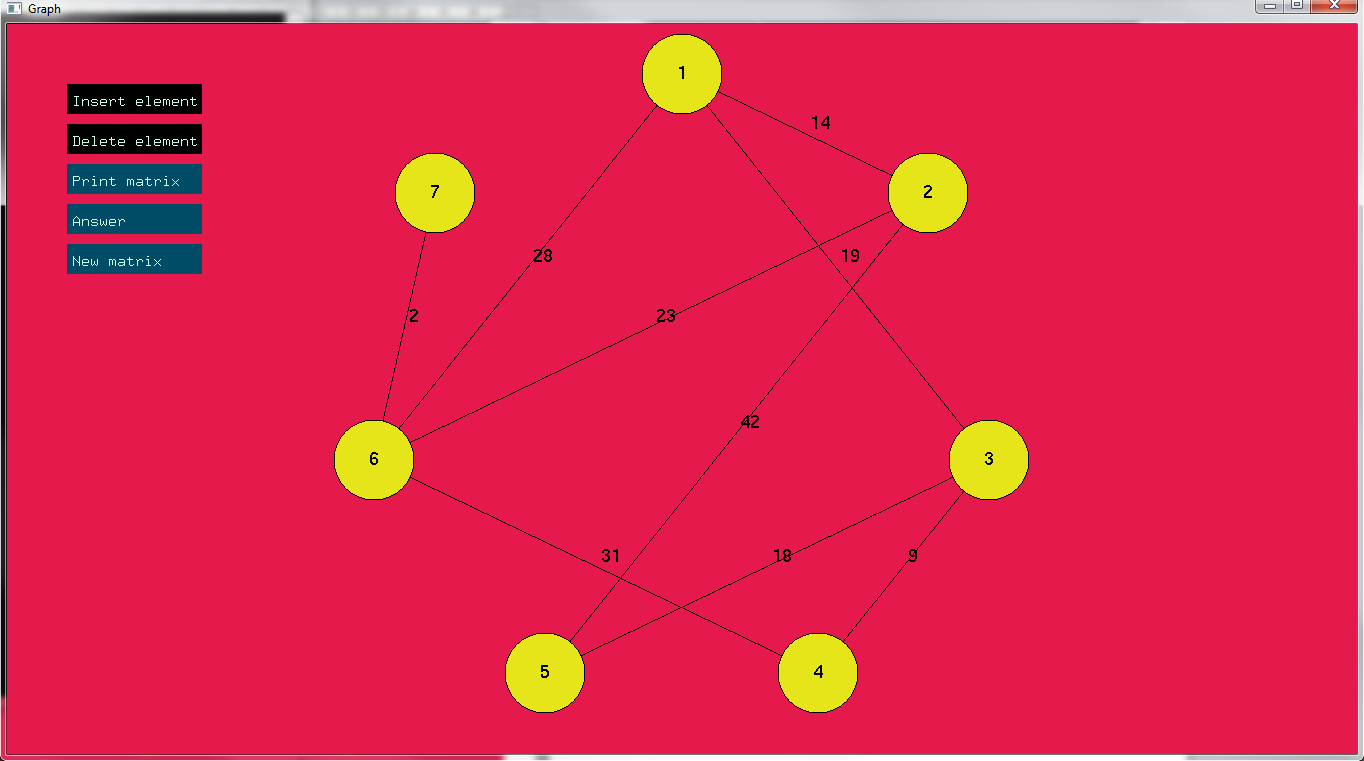


Рисунок 15

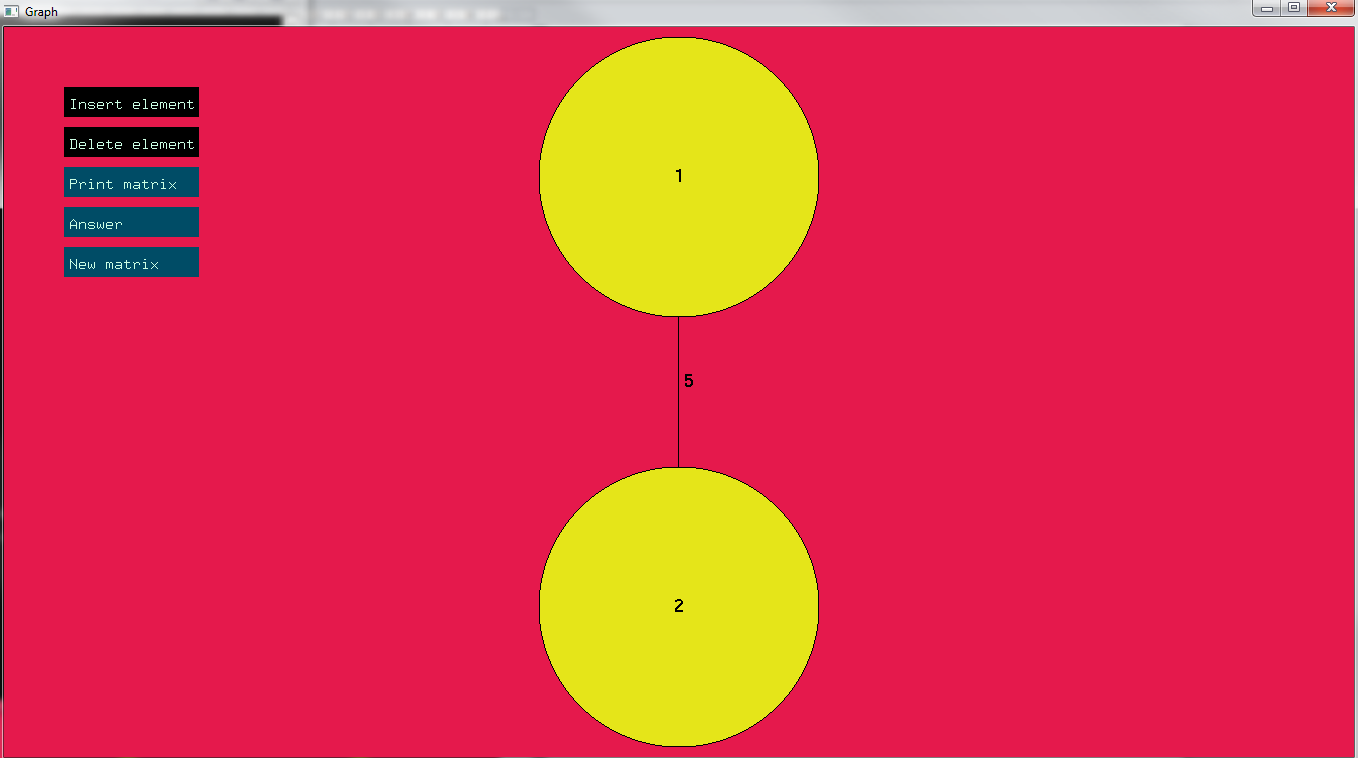


Рисунок 16

**Описание решения задачи калькулятор**

1. Написание кода было выполнено на языке программирования «С++»

визуализации идей использованы платформы «Windows Forms». Данная платформа выбрана автором из-за доступности обучающего материала в сети интернет.

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, реализация такова. Нельзя не отметить проверки на случаи неправильного ввода, что убережет пользователя от негативных эмоций в периоде эксплуатации, а компилятор от ошибок.

**Описание основных частей в коде**

На рисунке 17 представлено событие-метод result\_Click, которое происходит по нажатию на кнопку «Вычислить». В нем происходит сам подсчет коэффициентов и сразу же проверка на некорректные входные данные.

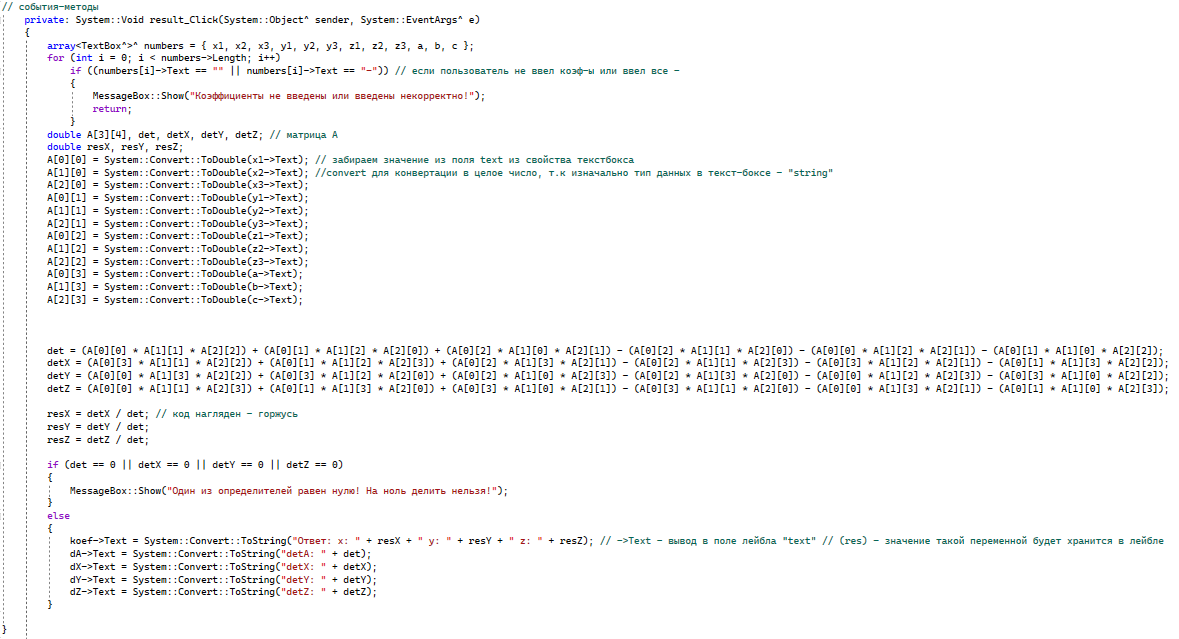


Рисунок 17 – событие-метод result\_Click

На рисунке 18 представлено событие-метод reset\_Click, которое происходит по нажатию на кнопку «Очистить поля». В нем происходит очистка полей по методу возвращения значения полей к начальным - пустым.

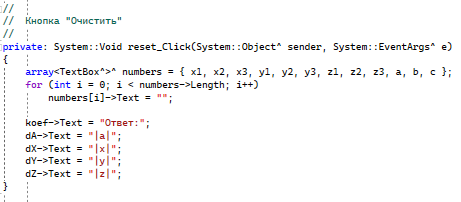


Рисунок 18 – событие-метод reset\_Click

На рисунке 19 представлено событие-метод x1\_KeyPress, которое происходит по нажатию на клавиши клавиатуры. В нем происходит проверка на ввод данных в поля самой программмы. В исключения ввода данных включены цифры, точка, клавиша «BackSpace» и знак минус. Аналогично со всеми остальными полями – текст-боксами.

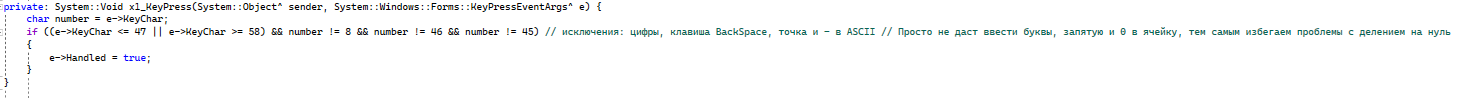


Рисунок 19

**Диграмма классов (рис. 20)**

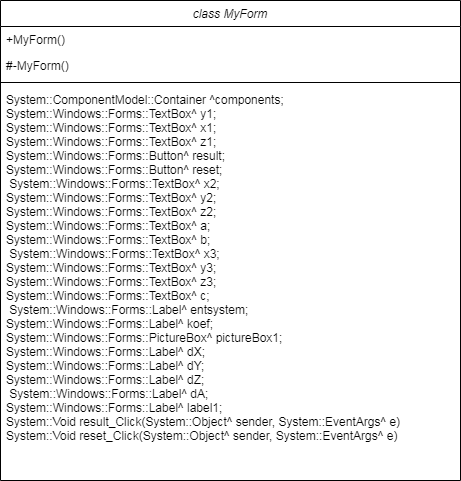


Рисунок 20

**Пример работы и результаты выполнения (рис. 21, 22, 23, 24)**

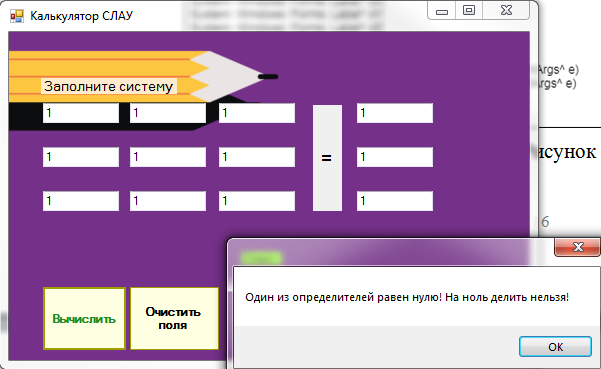


Рисунок 21

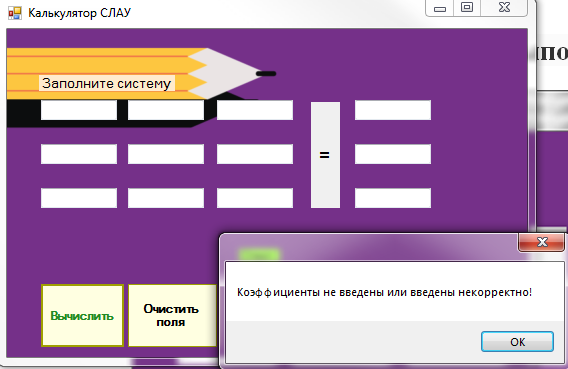


Рисунок 22

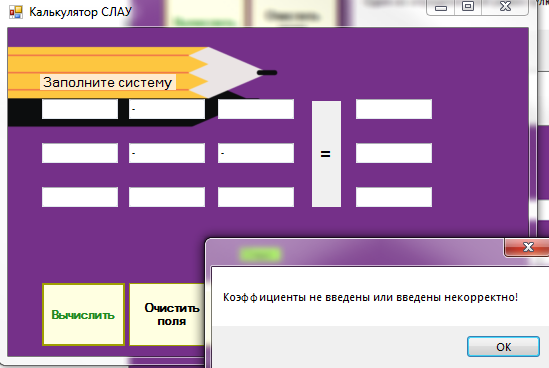


Рисунок 23

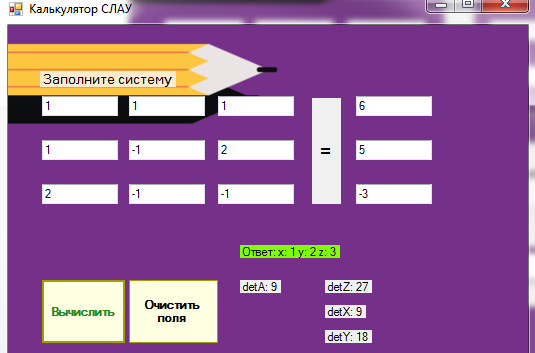


Рисунок 24