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»

**Отчёт по творческой работе “Коммивояжер и Калькулятор градусов”**

**по дисциплине**

**«Основы алгоритмизации и программирования»**

Выполнил студент гр. ИВТ-21-1б

Волков Роман Алексеевич

Проверил:

(подпись) (оценка)

г. Пермь-2022

**Коммивояжер:**

**Постановка задачи:**

1. Реализовать ввод матрицы смежности.
2. Реализовать функции для решения задачи методом ветвей и границ.
3. Реализовать нахождение наикратчайшего пути.
4. Реализовать отрисовку графа с помощью OpenGL.
5. **Постановка задачи:**

Переменные:

SourceVertex - Начальная вершина.

TargetVertex - Конечная вершина.

EdgeWeight - Вес ребра.

AmountVerts - Кол-во вершин.

AdjMatrix - Матрица смежности для вывода и счёта пути.

Inter - Матрица смежности для алгоритма Дейкстры.

Helpinter - Вспомогательная матрица.

Path - Матрица для хранения пути.

VertC - переменная для хранения координат вершин.

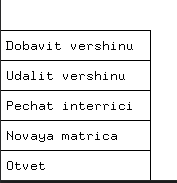
String Text - для введения текста в вершинах и меню.

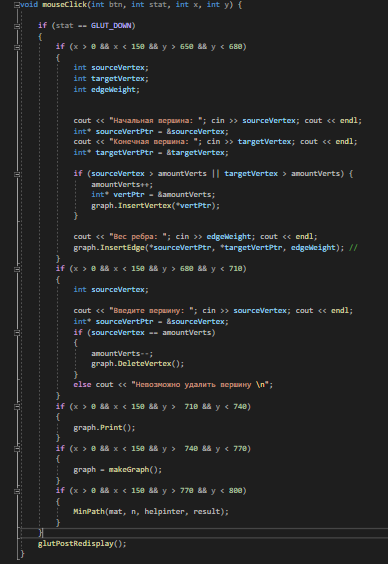
Также есть подключённая библиотека OpenGL <glut.h> дял возможности отрисовки графа.

**2) Решение задачи:**

1. Выполнить отрисовку кнопок и привязка к ним функций

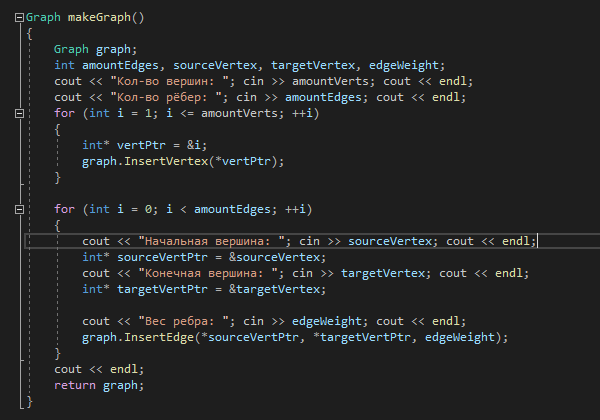
Реализовано координатами положения мыши и функции Glut\_down считывающей нажатие.



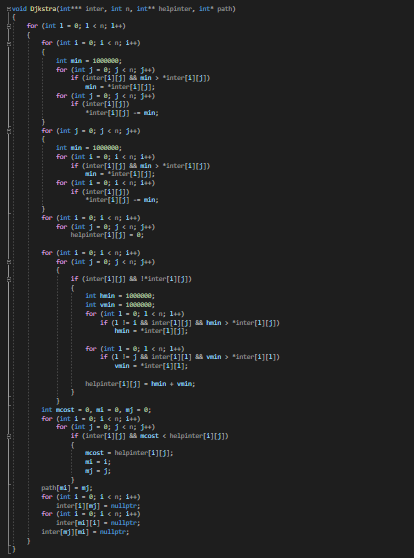


1. Реализовать ввод матрицы смежности

На ввод идёт кол-во вершин и рёбер, а дальше происходит их связывание.



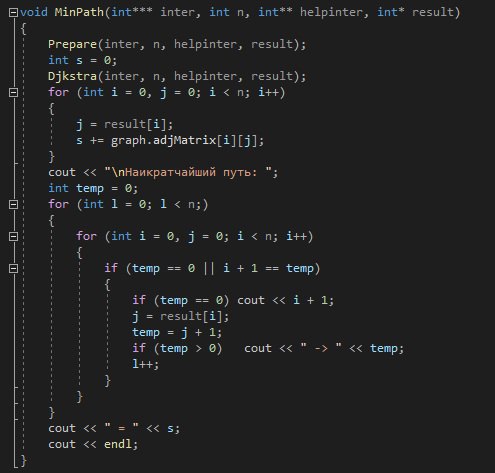
3) Реализовать алгоритм Дейкстры



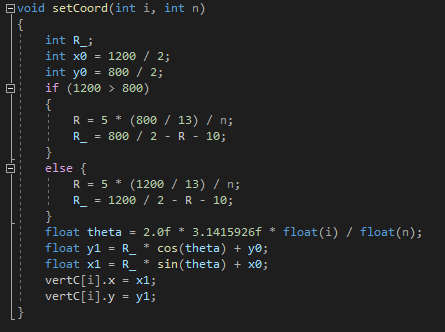
1. Функция нахождения и вывода наикратчайшего пути

Переменная s хранит значение минимального пути

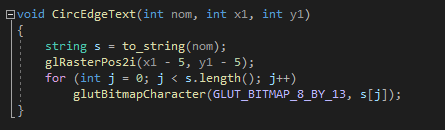
Путь выводится в формате “Номер начального пункта” - > “Номер конечного пункта”

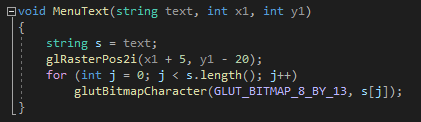


1. Функция координат вершин и их радиуса



1. Функции отрисовки текста в вершинах и меню

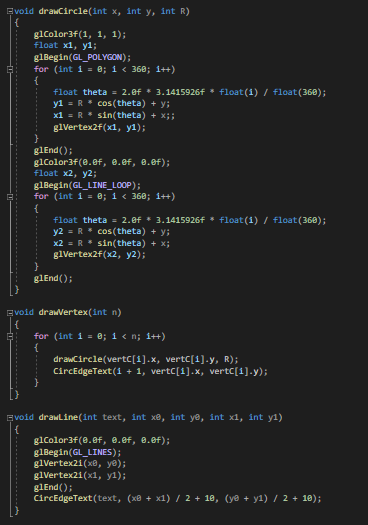




1. Функции отрисовки вершин с координатами и связывающих их рёбер

Круг рисуется в зависимости от кол-ва вершин, и чем больше вершин, тем меньше радиус.

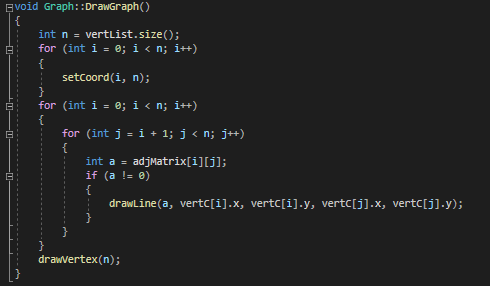
Отрисовка линий и вершин происходят сразу с текстом.



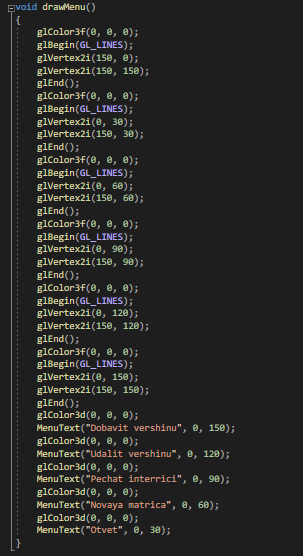
1. Функция отрисовки графа

Граф рисуется круговым расположением, где вершины являются вершинами правильного вписанного многоугольника с n вершинами.

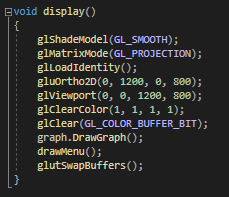
Линии отрисовываются раньше вершин, поскольку нужно чтобы вершины накладывались на них, а не наоборот



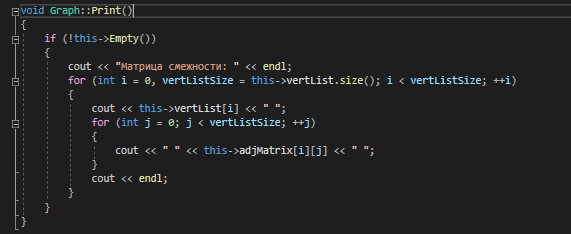
1. Функция отрисовки меню в левом нижнем углу



1. Функция отрисовки окна с параметрами и графом

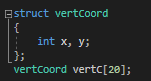


1. Функция вывода матрицы смежности

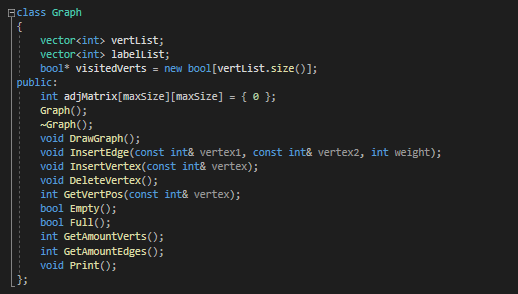


Если граф не пустой, то происходит отрисовка

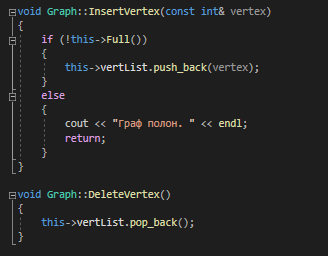
11) Структура для координат вершин



12) Класс Графа

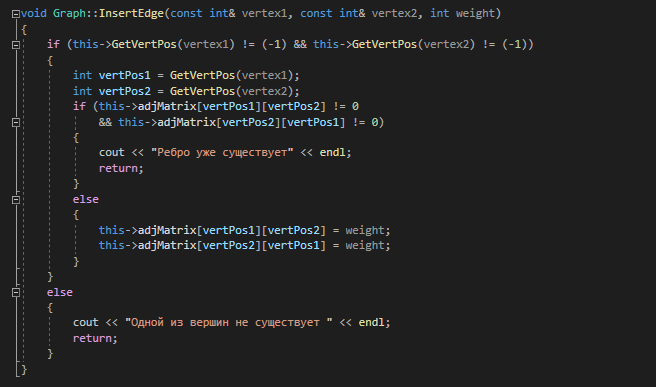


13) Функции вставки и удаления вершин через библиотеку <vector>

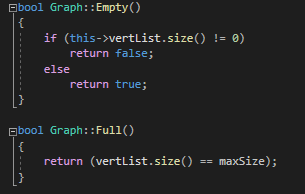


14) Функция вставки ребра

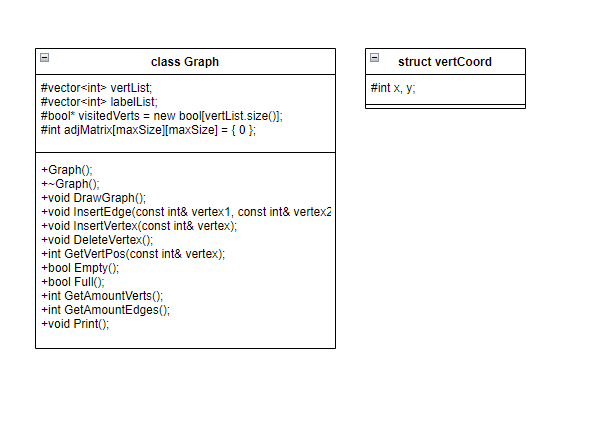
Вставка происходит если ребро существует или ребра ещё нет и оно не имеет значение



15) Функции проверки пустоты или наполненности графа



**3) UML-диаграмма**



**Калькулятор градусов:**

**1)Постановка задачи:**

1. Создать в конструкторе оформление калькулятора.
2. Прописать функции для действий.
3. Проверить правильность выполнения всех действий.

**2)Решение задачи:**

Переменные:

A,b,c,d,f,g - хранение частей чисел first и second.

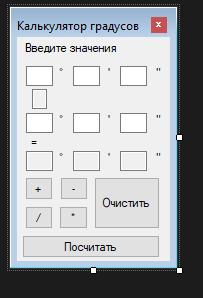
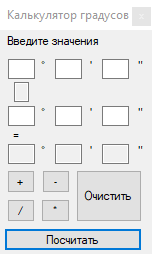
First,second,third - переменные для храения первого, второго числа и результата.

Var - переменная для хранения действий.

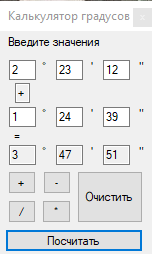
Plus,minus,mult,divide - хранение знаков действий для преобразования.

Решение было реализовано с помощью WindowsForm.

1) Внешний вид запущенного калькулятора и в конструкторе.

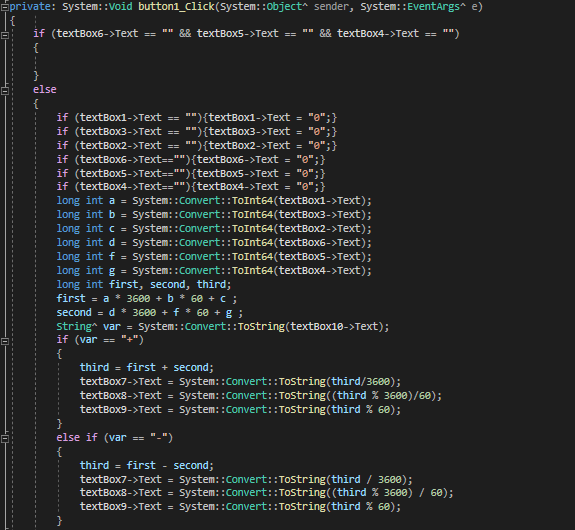


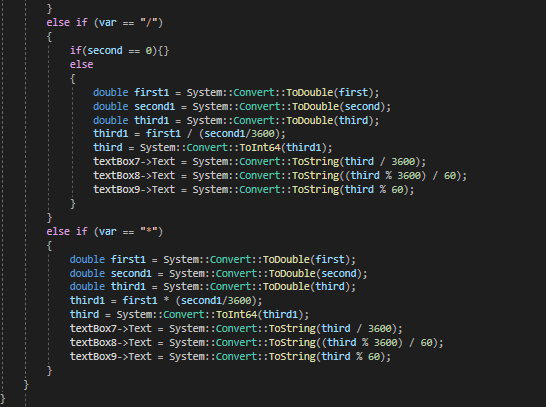
2) С введёнными данными и полученным результатом.



3) Функции сложения, вычитания, деления и умножения. Здесь сразу производится проверка на пустоту в поле 2-го числа. Проверка на пустоту полей после проверки на пустоту полей 2-го числа с заменой их на 0, если они пустые, т.к. программа не будет работать без этого.

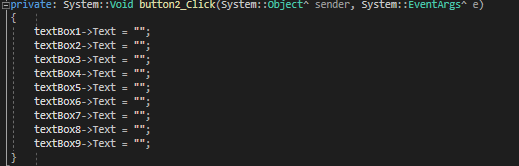
Дальше представлен алгоритм перевода и вычисления итогового числа с помощью функций преобразования System::Convert::Тип\_Данных из string в int



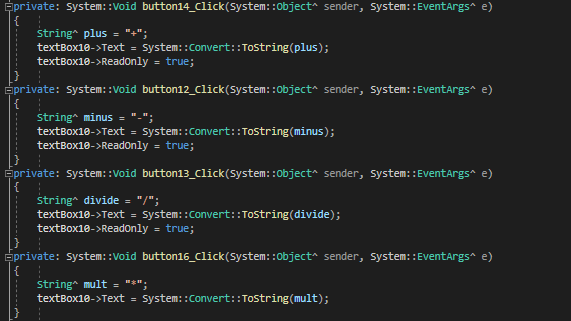


4) Функция очистки полей

После нажатия на кнопку все поля со значениями очищаются



5) Функции ввода для поля знаков действий, самих знаков действий



**3) UML-диаграмма**

