Министерство образования и науки Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

«**Пермский национальный исследовательский политехнический университет**»

Кафедра «Информационные технологии и автоматизированные системы»

ОТЧЁТ

по творческой работе.

Вариант 15 (Калькулятор).

Вариант 21 (Задача Коммивояжёра).

Выполнил:

Студент группы ИВТ-20-2б

Галинов Олег Юрьевич

Проверила:

Доцент кафедры ИТАС

Полякова О. А.

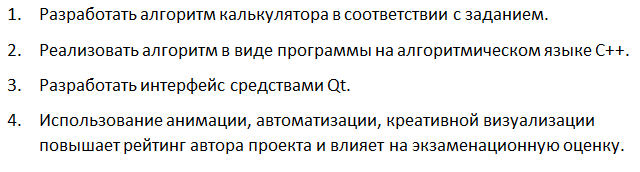
Пермь, 2021

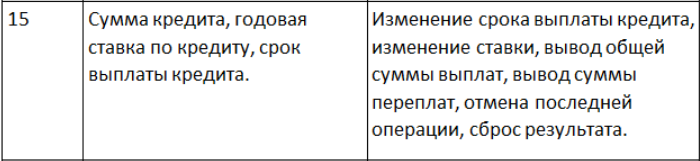
**Введение**

Для решения поставленной задачи, а именно разработка калькулятора и решение задачи Коммивояжёра, будет использована среда разработки QtCreator, позволяющая разработчику создавать графический интерфейс для своего приложения. Используемый язык программирования – C++.

**Разработка калькулятора**

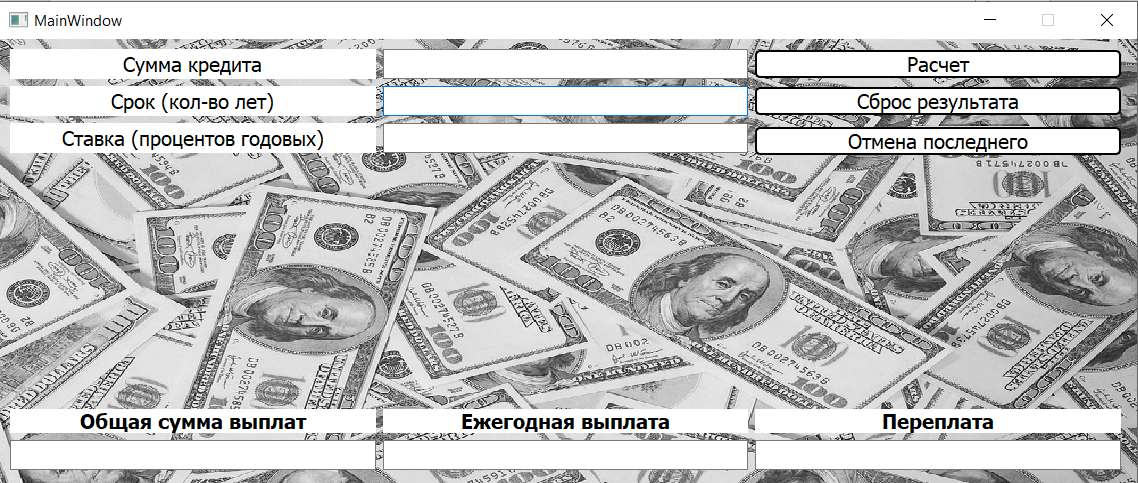
**Постановка задачи**



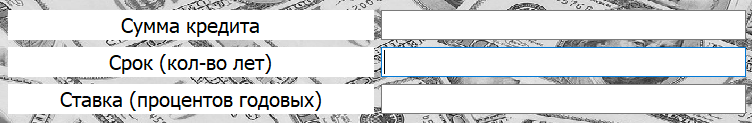


**Анализ задачи**

Интерфейс:

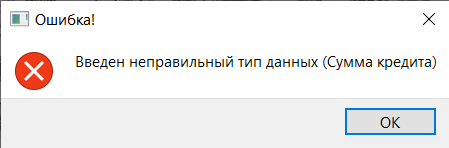


1. Описание алгоритма работы калькулятора.
   1. Ввод данных. Для получения программой пользовательских данных, то есть сведений по кредиту, на форме спроектированы 3 поля ввода (класс LineEdit), каждое из которых подписано с помощью меток (класс Label). Пользователь может ввести следующую информацию: сумму кредита, процентную ставку и срок.

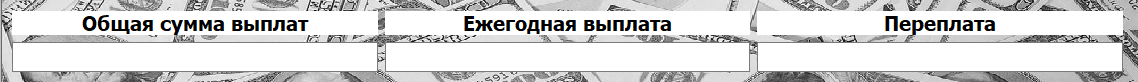


Предусмотрены проверка на ввод данных, не являющихся числом.

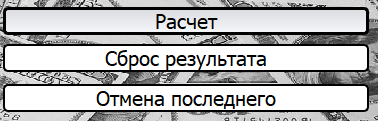
Кроме того указывается строчка, в которой ошибка была допущена.



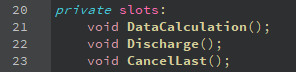
* 1. В полях вывода так же реализованы 3 строчки, в которые записываются результаты.



* 1. Кнопки на форме подсвечиваются при наведении курсора на них, это сделано для более понятного и простого восприятия.



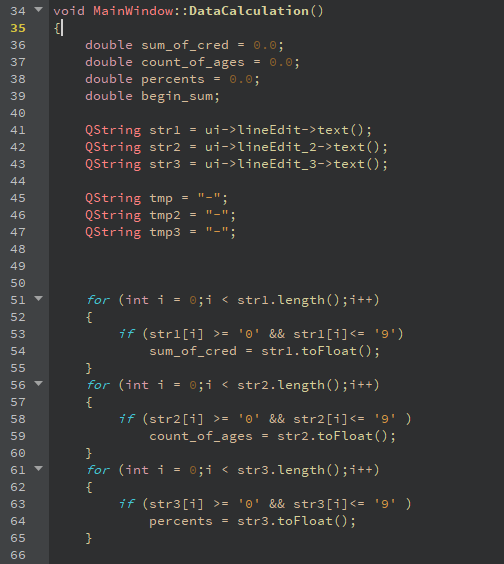
1. Описание структуры программы.
   1. В файле mainwindow.h объявлено 3 метода, каждый из которых является кнопкой.



* 1. В файле mainwindow.cpp кроме 3 методов создается 3 переменные, которые будут хранить данные для отмены последнего действия.



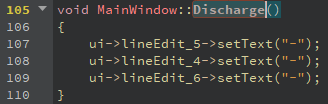
* 1. В методе DataCalculation сначала происходит проверка строки. Если в ней есть символы, которые не являются цифрами, то программа выдаст сообщение об ошибке. А если строка корректна, то запишет данные в переменные.



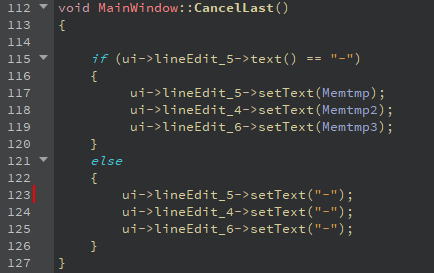
* 1. Далее программа проверяет изменились ли переменные, нужные для расчета. Если нет, то выдаются сообщения об ошибке, а если в них было что-то записано, то программа производит вычисления. Здесь же заполняются строчки вывода.



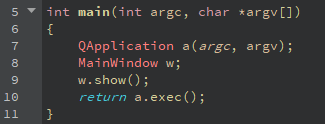
* 1. В методе Discharge происходит сброс всех результатов и строки для вывода делаются пустыми.



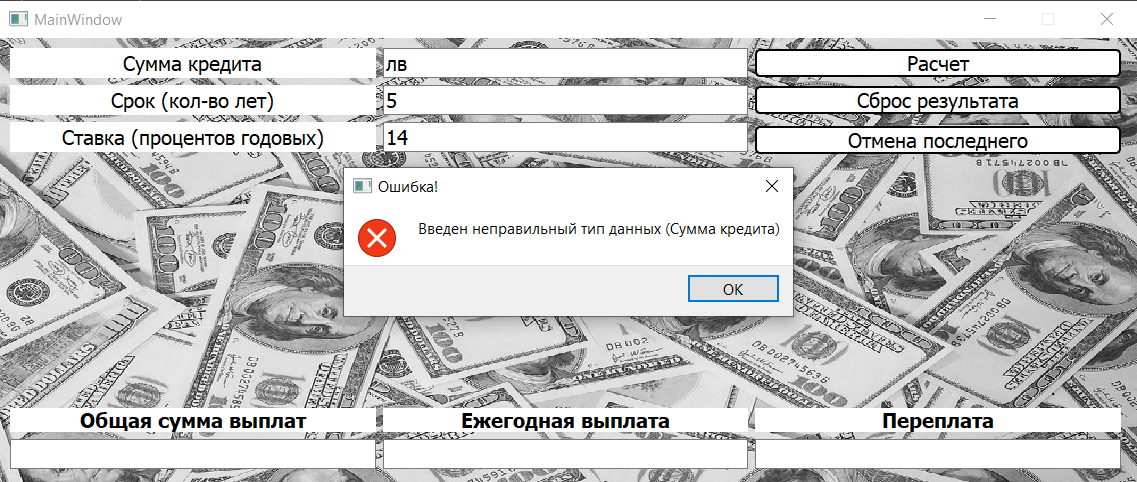
* 1. В методе CancelLast сначала идет проверка на строки вывода, если они пустые, то туда записываются прошлые рез-ты, а если в них что-то было, то строки становятся пустыми (или возвращаются на шаг назад).

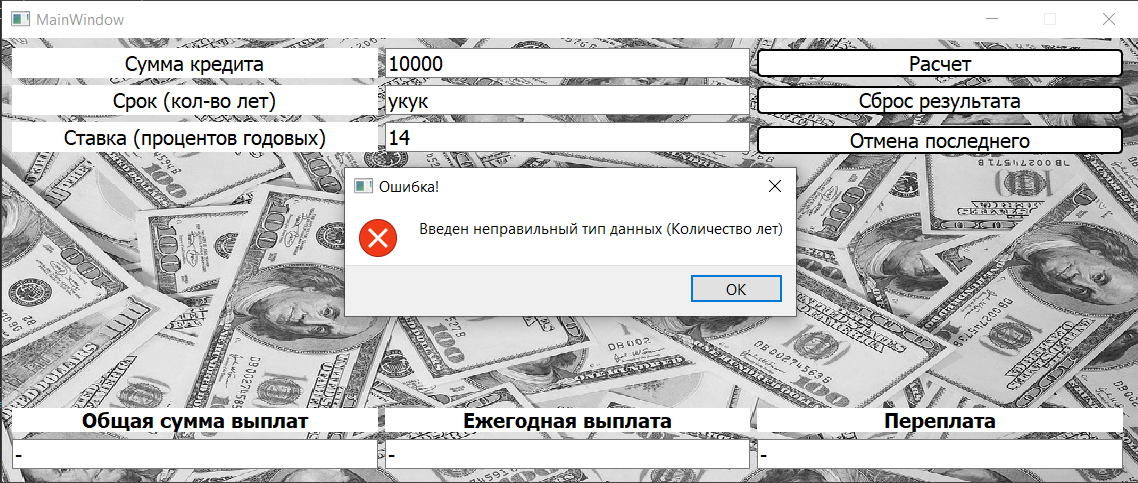


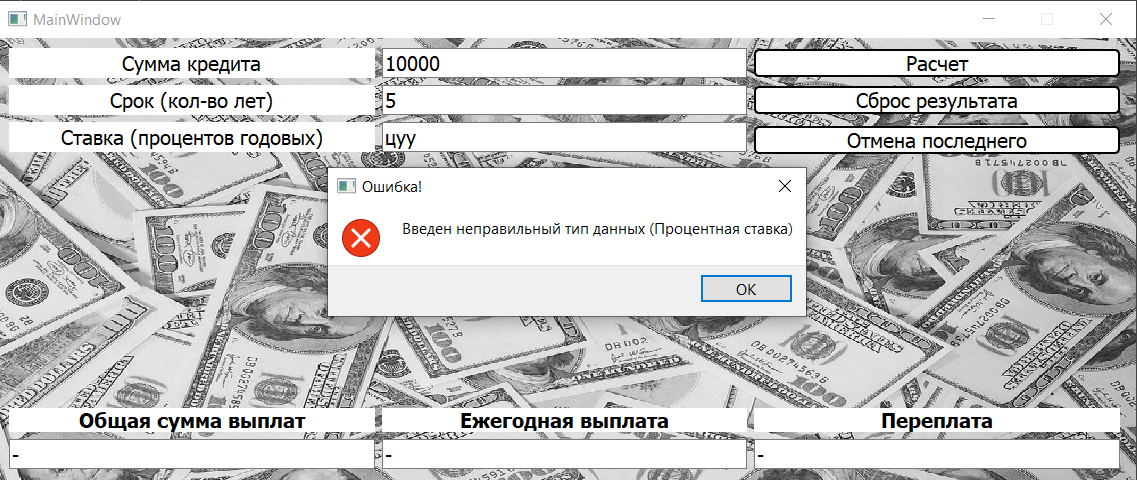
1. Функция main(). Нет отличий от того, что генерируется по умолчанию.



1. Тестирование работы калькулятора.
   1. В зависимости от места введения некорректных символов, будет выписана строка, в которой они есть, чтобы пользователь мог исправить ее и ввести верные данные.



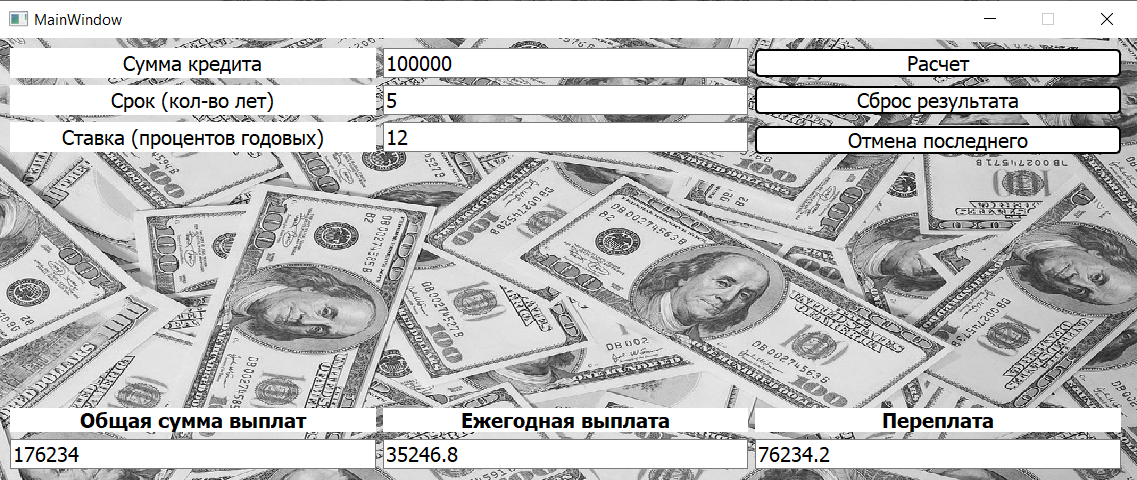




1. Объяснения результатов работы калькулятора. Далее будет продемонстрирована работа калькулятора.
   1. Калькулятор перед вводом данных.



* 1. Введем сумму кредита, срок и процентную ставку.



Как видим программа отработала и верно вывела все данные.

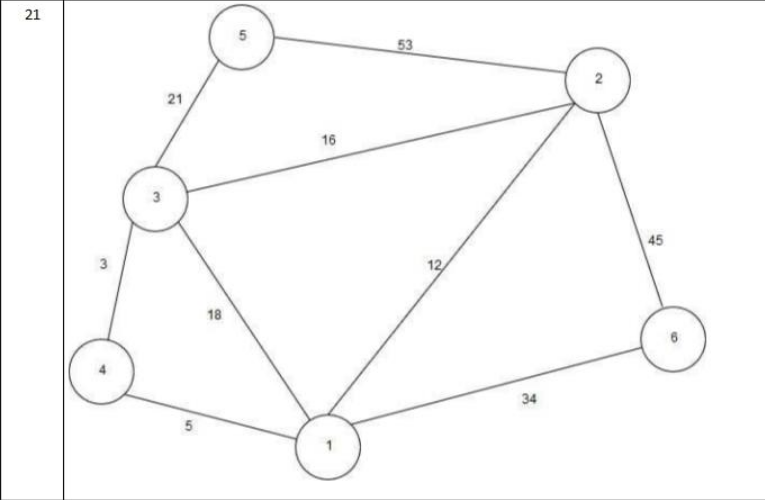
**UML-диаграмма**

В программе не описываются дополнительные классы.

**Задача Коммивояжёра**

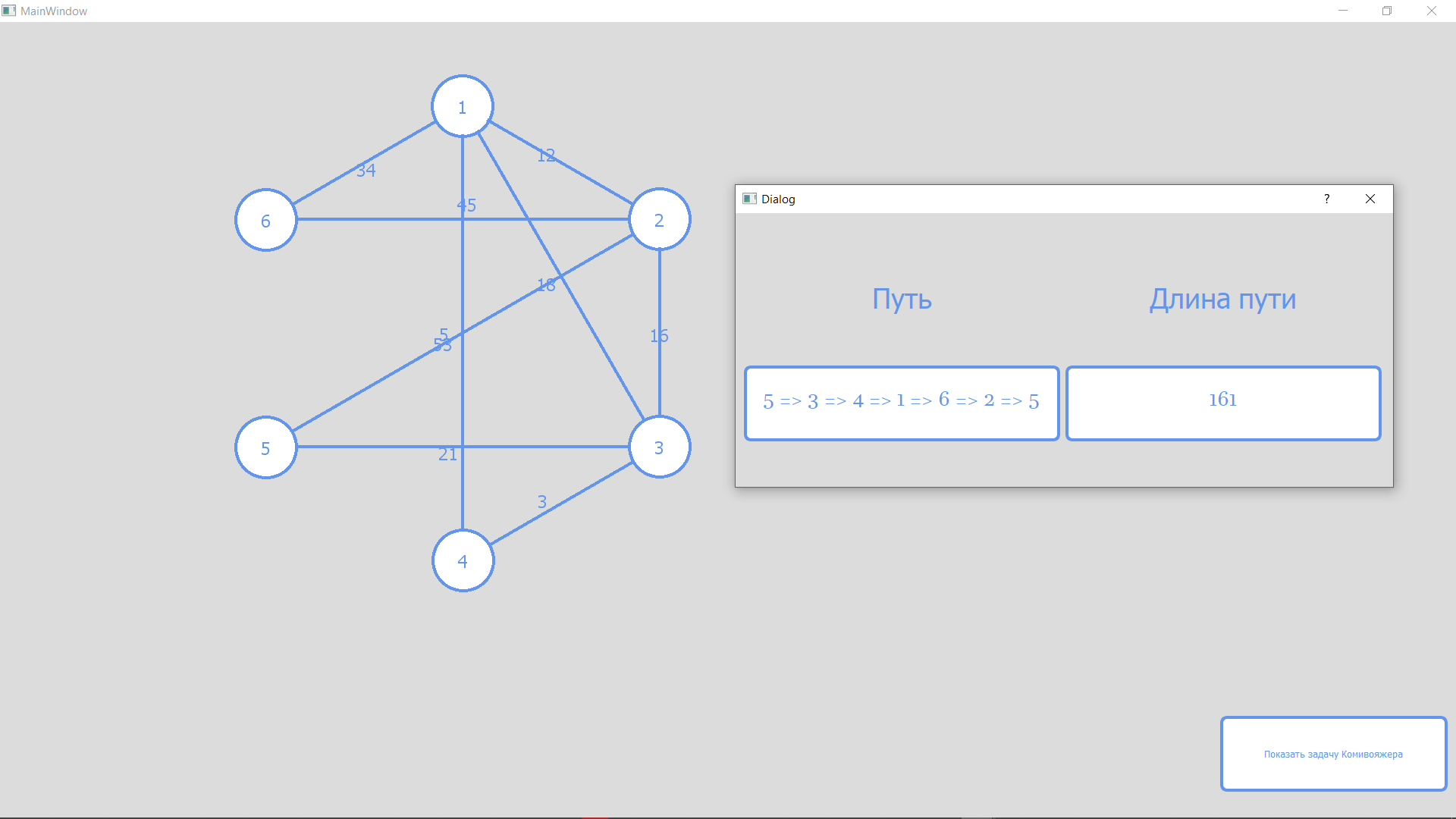
**Постановка задачи**

Решить задачу Коммивояжёра на данном графе.

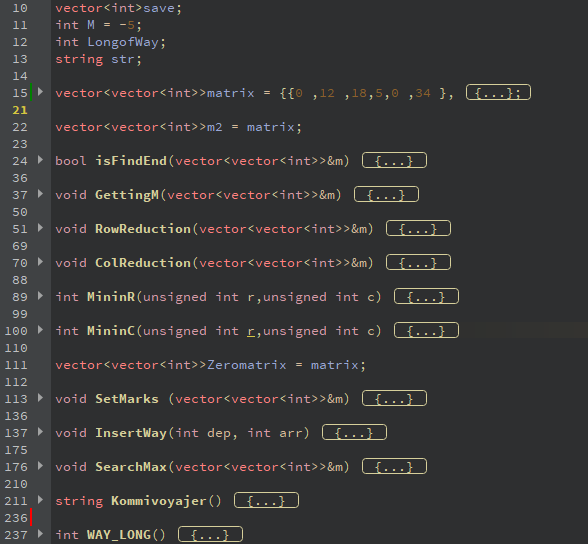


**Анализ задачи**

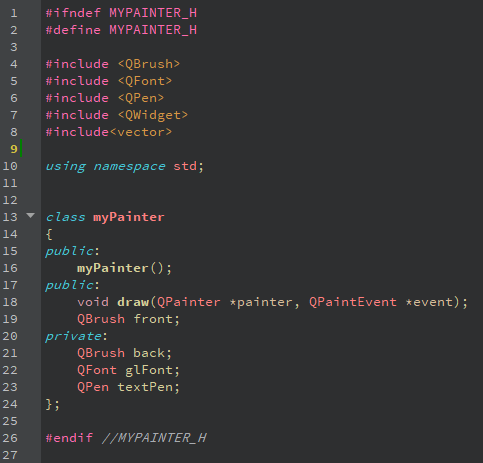
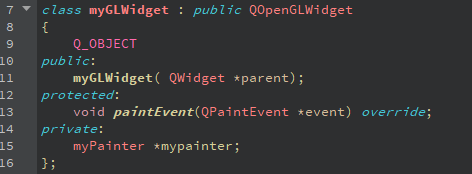
Интерфейс:



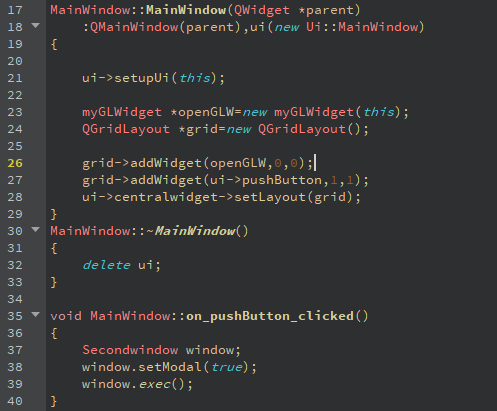
1. Описание алгоритма работы.
   1. Отрисовка. Узлы графа располагаются на окружности. Между узлами рисуются дороги.
   2. Решение задачи Коммивояжёра выводится в дополнительном окне.
2. Описание структуры программы.
   1. Файл graph.h. В нем написан метод нахождения решения задачи Коммивояжёра.



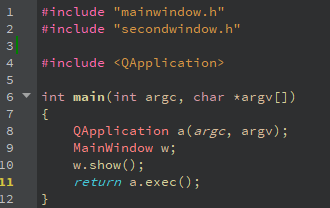
* 1. Класс mypainter. Наследуется от класса myglwidget. В нем описаны методы для отрисовки графа.

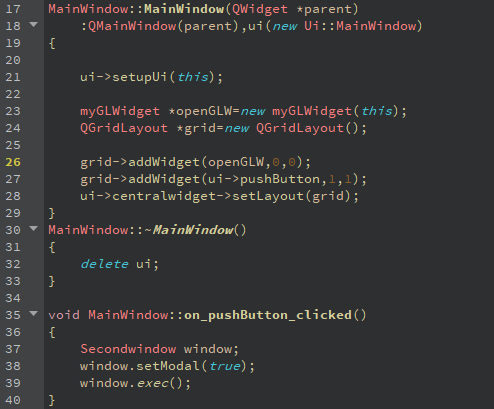
* 1. Класс MainWindow описывает окно. Одним из полей является ссылка на экземпляр класса myglwidget.



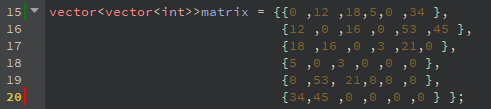
1. Функция main(). Нет отличий от того, что генерируется по умолчанию.



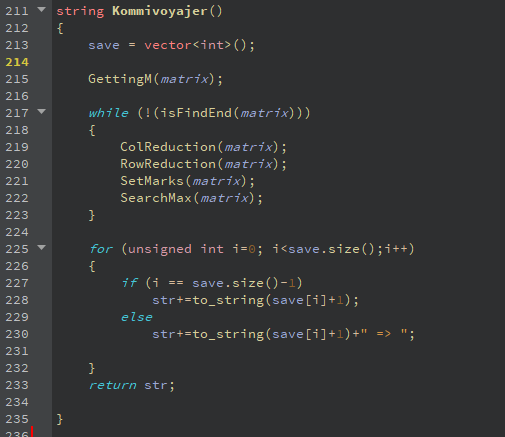
1. Объяснения результатов работы программы.
   1. В конструкторе MainWindow происходит создание экземпляра класса myglwidget и последующий вызов методов для отрисовки графа.



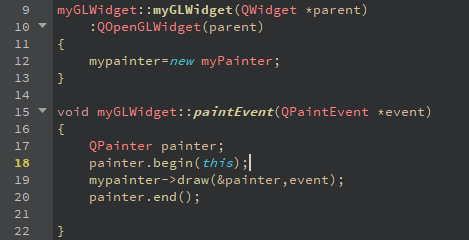
* 1. В файле Graph по умолчанию прописывается матрица смежности.



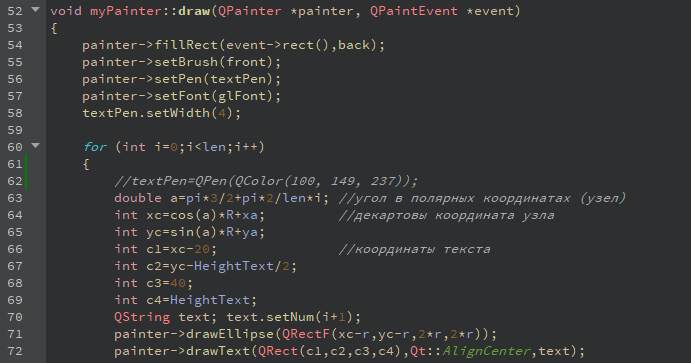
* 1. Вызов метода решения задачи Коммивояжёра.



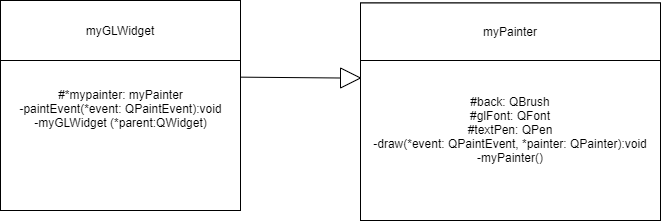
* 1. Вызов метода отрисовки.



* 1. Пример отрисовки узлов, вызванный в draw.



**UML-диаграмма**



**Заключение**

Ссылка на видео:

https://www.youtube.com/watch?v=Jt2tKl\_X-cE&t=87s

Ссылки на GitHub:

https://github.com/galileg/Calculator https://github.com/galileg/KomivTask