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 № 2
по дисциплине
«Основы алгоритмизации и программирования»
на тему
«Задача Коммивояжера» и «Разработка автоматизированного рабочего места для оценщика недвижимости»

Выполнил студент гр. ИВТ-23-1б
Попова Мария Вячеславовна

Проверил:
доцент кафедры ИТАС
Яруллин Денис Владимирович

(оценка) (подпись)

(дата)

г. Пермь, 2024

Автоматизированное рабочее место для оценщика недвижимости (АРМ)

Постановка задачи

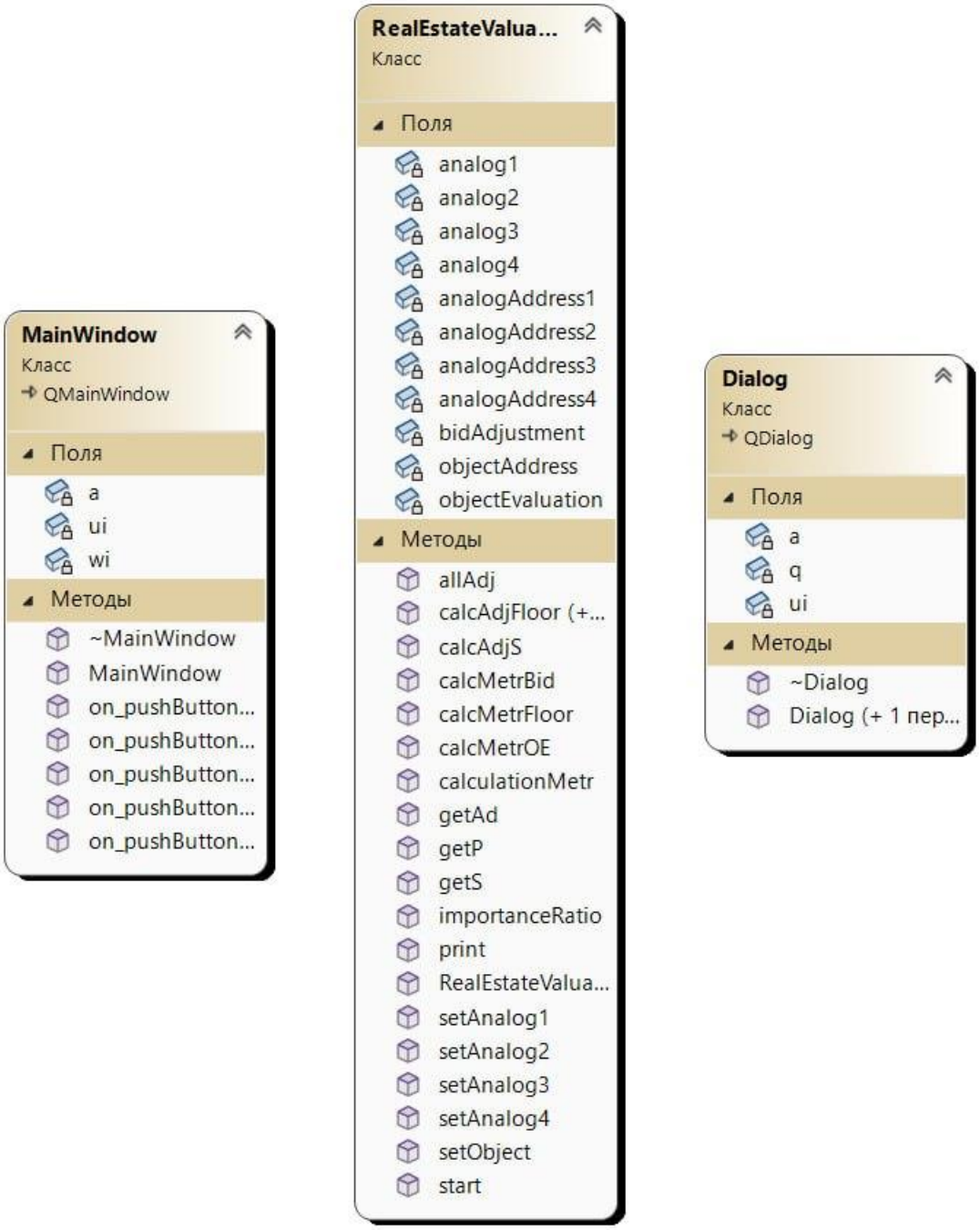
Разработка приложения, которое помогает оценщику недвижимости посчитать стоимость объекта на основе аналогов.

1. Проанализировать тему
2. Составить логический план по оценке недвижимости
3. Оформить работу в QT

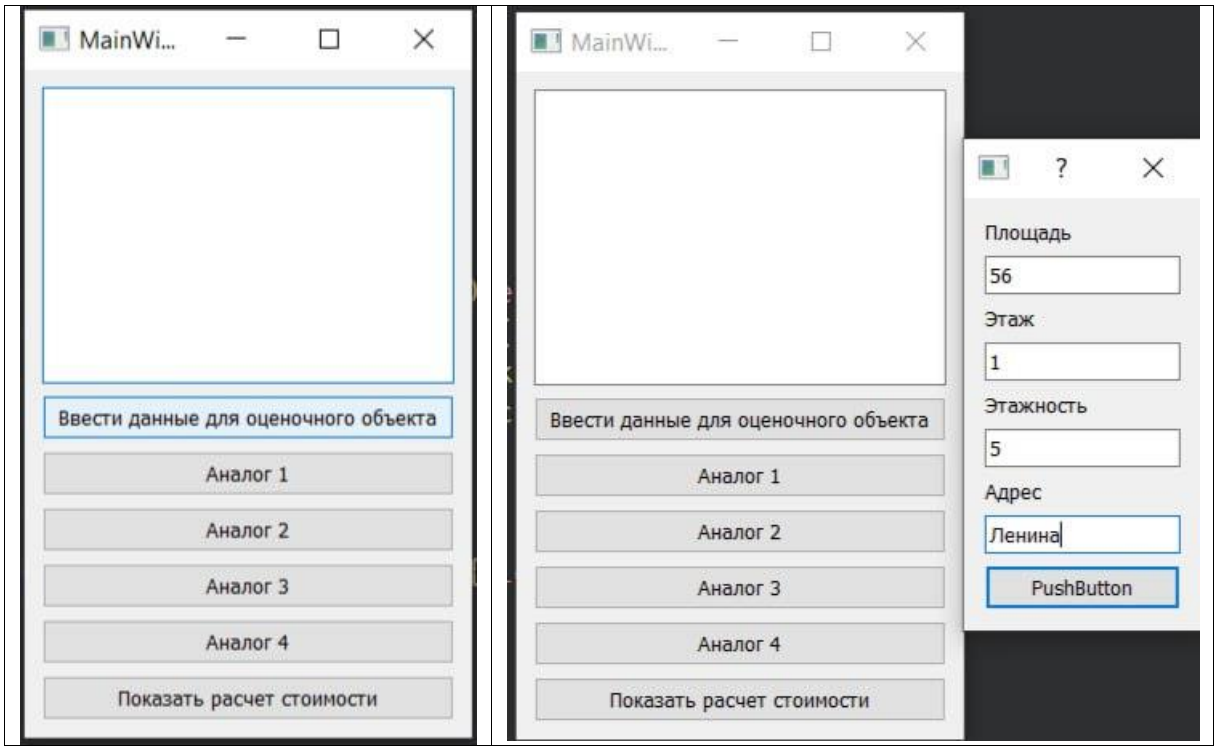
Анализ Задач

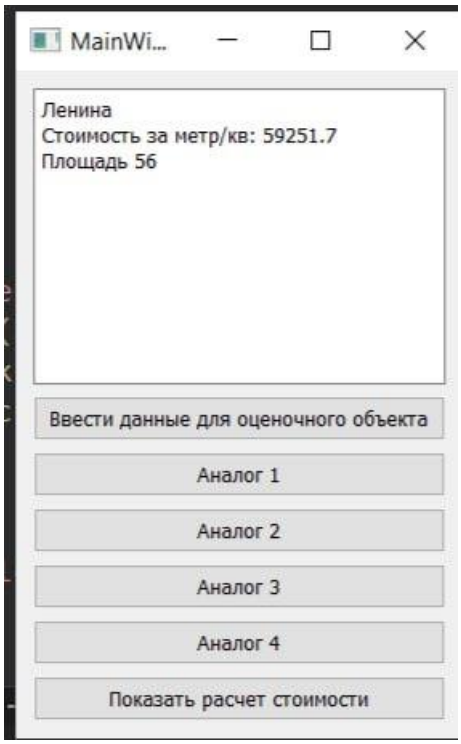
1. Проконсультироваться с оценщиком и проанализировать его работу
2. Собрать формулу расчета
3. Написать алгоритм для оценки объекта, основываясь на формулах и 4 аналогах. Составляющее:
    3.1. Разделение анализа на шаги
    3.2. На каждом шаге корректируется стоимость объекта
    3.3. Получение результат в виде стоимости оценочного объекта на основе 4 аналогов
4. Прописать логическую часть программы
5. Написать визуализацию

UML-диаграмма



Демонстрация работы





Задача Коммивояжера

Постановка задачи

1. В качестве варианта для демонстрации работы программы взять свой вариант задания из лабораторной работы «ГРАФЫ» (не менее 6 вершин, двунаправленный граф). Модифицировать граф таким образом, чтобы для этого графа можно было решить задачу Коммивояжера. Разработать программу, которая будет универсальной на любом наборе исходных данных.

2. Проработать визуализирующую часть в программе средствами QT. Интересные дизайнерские и конструкторские решения в интерфейсе применить: добавление новых узлов, перемещение узлов, установка связей между узлами, разрыв.
3. Исходные данные должны приниматься с консоли, либо через графический интерфейс с помощью Qt.

4. Задокументировать программу диаграммой классов UML.

5. Подготовить общий отчет

Анализ Задач

Graph Class: Класс, представляющий граф с методами для добавления/удаления узлов, установки весов ребер, поиска кратчайших путей и других операций. Включает матрицы смежности mat и matrix (и транспонированную матрицу matrixT).

Node Class: Класс, представляющий вершину в дереве решений. Вершина содержит текущий город, список посещенных городов, оценки нижней и верхней границ, а также методы для генерации следующих вершин.

Container Class: Класс, который управляет списком вершин и выполняет операции добавления, удаления и обновления текущего состояния.

Основные этапы алгоритма
Инициализация: Создается объект класса Graph и заполняется матрица смежности вершин с помощью метода generateMatrix.

Создание корневой вершины: Создается начальная вершина Node с пустым списком посещенных вершин, указывающим на начальную вершину (обычно вершина 0).

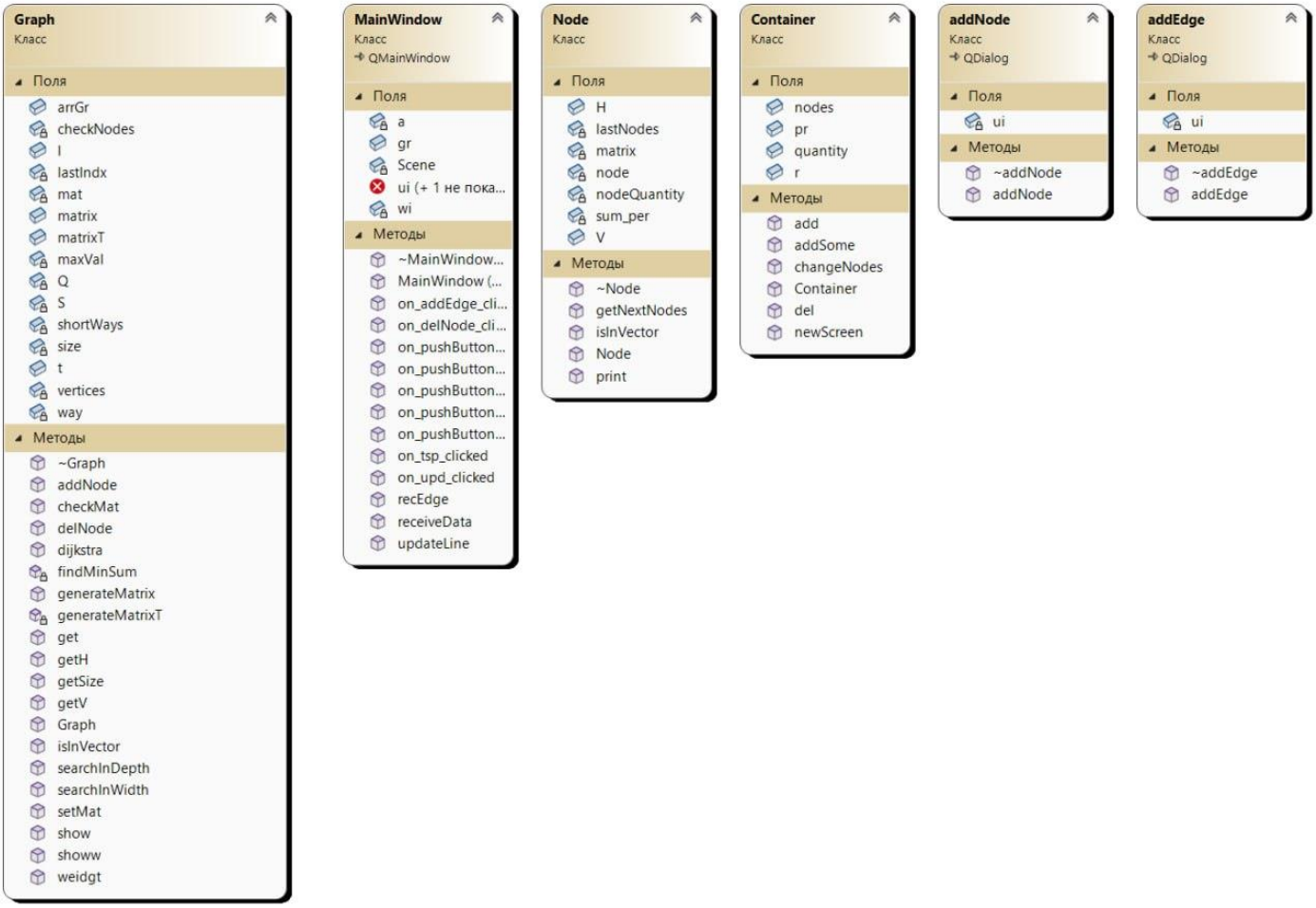
Расширение вершин:
Для текущей вершины генерируются все возможные следующие вершины (вершины, которые еще не посещены).
Для каждой следующей вершины вычисляются оценки верхней и нижней границ.
Эти вершины добавляются в контейнер Container.

Выборка:
Вершины с верхней оценкой, превышающей текущий рекорд, отбрасываются.
Таким образом, дерево решений не расширяется в тех направлениях, которые явно не могут привести к улучшению текущего наилучшего решения.

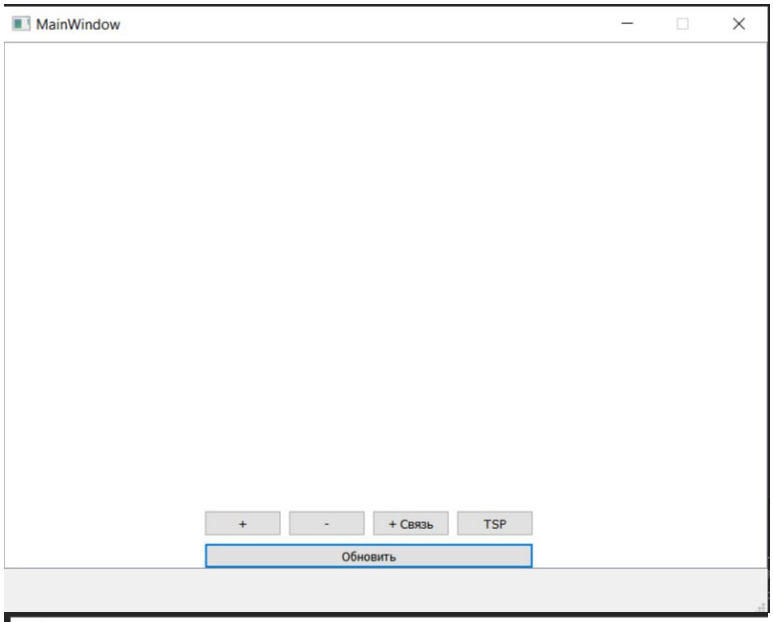
Выбор следующей вершины:
Среди всех вершин в контейнере выбирается вершина с наименьшей нижней оценкой.
Эта вершина становится текущей, и процесс расширения повторяется.

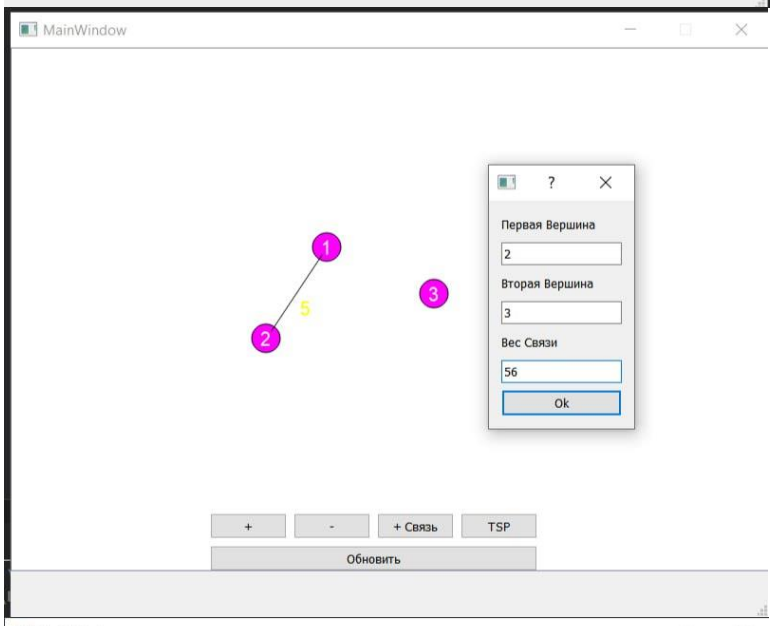
Завершение:
Алгоритм продолжается до тех пор, пока не будут рассмотрены все возможные вершины.
Рекордный путь (наименьшая верхняя оценка) будет содержать оптимальный маршрут для коммивояжера.

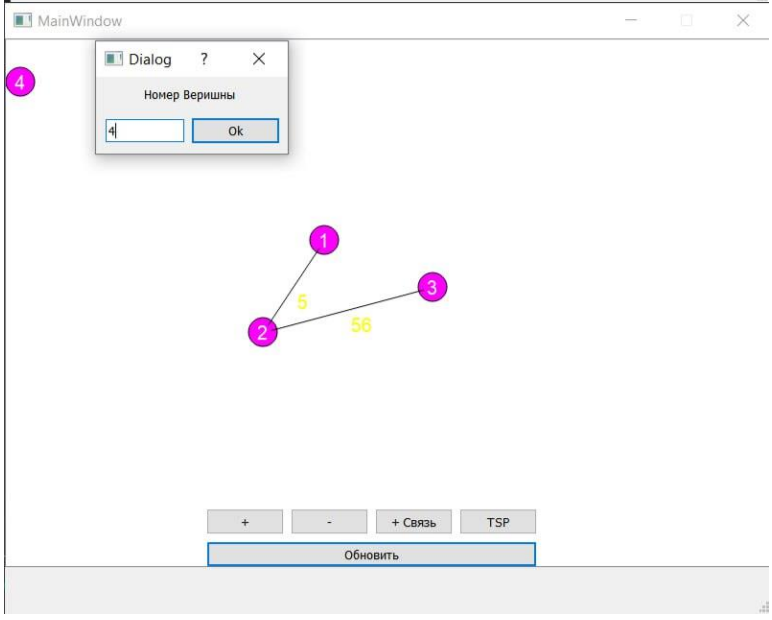
UML - диаграммы

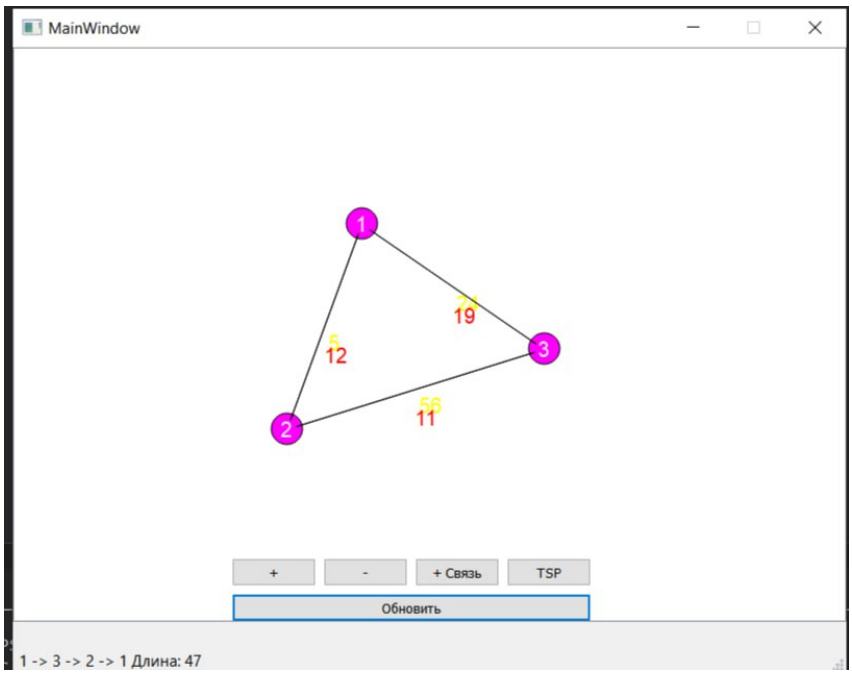


Демонстрация работы








MainWindow
1
2
3
19
12
11
+
-
+ Связь
TSP
Обновить
1 -> 3 -> 2 -> 1 Длина: 47

GitHub

https://github.com/RuFo2/Labs_PSTU_2023/tree/main/Sem_2.gitkeep/creative%20works.gitkeep

YouTube

https://www.youtube.com/watch?v=zo0XTpR8_qo