Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
Кафедра вычислительной техники

ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

проекта «GraphHelper»

Преподаватель _____ Васильев В.С.
подпись

Студент КИ19-08Б, 031940024 _____ Звягин С.А.
подпись

Красноярск 2020

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Приложение «GraphHelper» предназначено для визуализации теории графов, помощи её освоения и выполнения ряда сопутствующих задач. Таким образом оконное приложение «GraphHelper» должно обеспечивать достоверное визуальное представление ключевых положений теории графов, наглядно предоставлять пользователю результат операций над графом(графами).

1 Функционал приложения

На основании означенной в введении проблематики, оконное приложение «GraphHelper» должно отвечать следующему функционалу:

- создание и удаление вершин, рёбер, петель, дуг графа;
- взвешивание рёбер графа;
- слияние вершин простого графа;
- стягивание ребра простого графа;
- дополнение простого графа;
- объединение графов;
- пересечение графов;
- выполнение алгоритмов над графами;
- сохранение графа;
- возможность отключения текстовых пометок на области холста;
- определение типа графа;
- составление матриц, ассоциативных с графом.

2 Данные

Приложение «GraphHelper» оперирует над графом – множеством вершин и набором линий(рёбер, петлей, дуг), соединяющих эти вершины.

Организация входных и выходных данных представлена на рисунке 1:

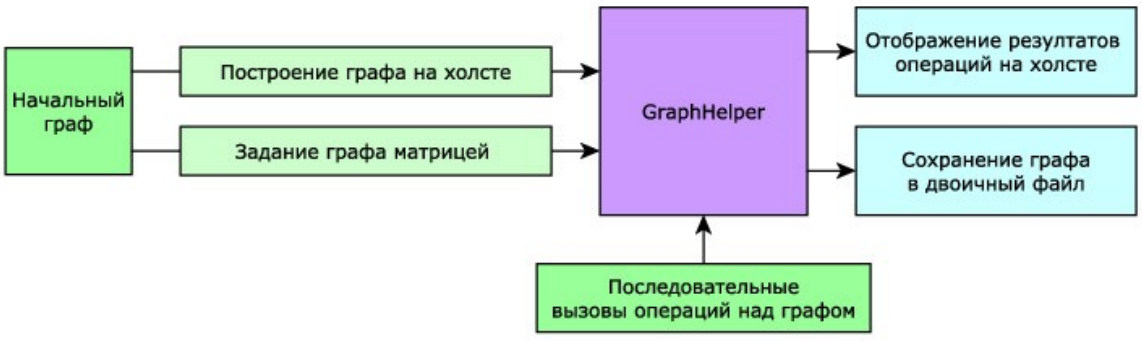

Рисунок 1 – Организация входных и выходных данных

В качестве входных данных выступает либо построение графа на области холста, с помощью инструментария программы, либо задание графа с помощью матрицы инцидентности, в таком случае программа сама распределит необходимые объекты по области холста.

В качестве выходных данные будет вывод результата операции, запрошенной пользователем, на области холста, либо при необходимости продолжения работ с графом в дальнейшем – сохранение графа в двоичный файл, в таком случае программой будет создан двочный файл. Структура двоичного файла представлена на рисунке 2:

Рисунок 2 – Структура двоичного файла

3 Объектно-ориентированный подход

Основываясь на входных данных целесообразно использование объектно-ориентированного подхода. Предполагается хранение внутри программы следующих объектов:

- граф;
- вершина;
- линия.

На рисунке 3 приведена объектно-ориентированная модель данных для мультиграфа:

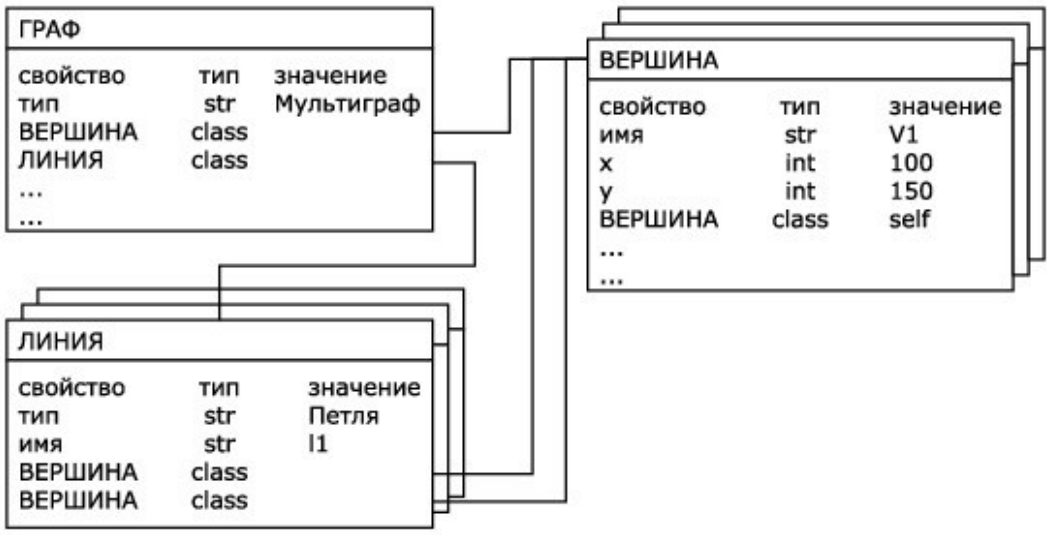

Рисунок 3 – Объектно-ориентированная модель данных

Объект класса «Линия» содержит два поля для хранения имён(указателей) объектов типа «Вершина» таким образом, что в случае, если граф имеет тип «Простой», то расстановка вершин по этим двум полям не имеет существенное роли, но если граф имеет тип «Орграф», то вершина стоящая в первом поле

означает начало, а стоящая на втором соответственно конец, в таком случае объект класса «Линия» получит тип «Дуга».

Каждая вершина хранит своё имя, свои координаты и список смежных с ней вершин.

На рисунке 4 изображена зависимость типа линий от типа графа:

Рисунок 4 – Типы линий от типа графа