2. Планирование и анализ задачи

2.1. Спецификация проекта

2.1.1. Структура классов

В рамках данной работы будет использоваться структура классов, схожая с MVC-парадигмой. Моделью в данной работе выступает граф, реализованный с помощью отдельного класса-контейнера. Этот класс содержит в себе информацию о вершинах в виде списка инцидентности и должен поддерживать операцию транспонирования. Контроллером в данной схеме является класс, реализующий исследуемый алгоритм. Этот класс должен реагировать на события графического интерфейса пользователя и реализовывать алгоритм Косорайю как в пошаговом, так и в автоматическом режиме. Видом в данной структуре будет являться класс окна, содержащий в себе все элементы графического интерфейса пользователя, включая поле графического вывода графа. На рисунке 1 представлена UML-диаграмма проекта.

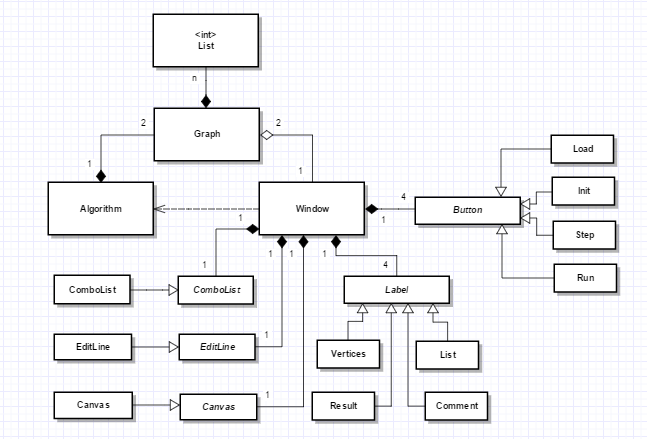


Рисунок 1. UML-диаграмма проекта

2.1.2. Описание интерфейса

Графический интерфейс пользователя состоит из 8 основных компонентов (см. рисунок 2).

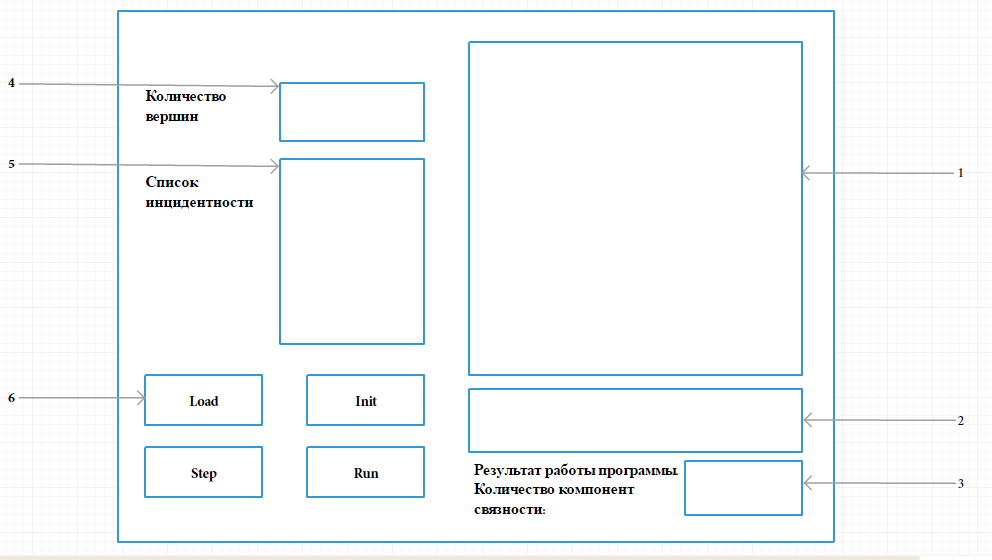


Рисунок 2. Схема графического интерфейса пользователя

Программа может поддерживать как ручной ввод пользователем, так и автоматический ввод из файла. Для того, чтобы ввести граф из файла, необходимо нажать кнопку Load в блоке 6 на рисунке 2. Если пользователь хочет ввести граф самостоятельно, то он должен ввести количество вершин графа в поле 4, после чего ввести список инцидентности в список 5. Для того, чтобы начать выполнение программы, необходимо нажать на кнопку Init в блоке 6, загрузив введенный граф в память. После этого кнопки Step и Run позволяют выполнить один шаг алгоритма и весь алгоритм полностью соответственно. Во время выполнения программы в окне 1 будет выводиться графическая интерпретация работы алгоритма с кратким пояснением в надписи 2; также в надписи 3 выводится итог работы алгоритма в виде количества найденных компонент сильной связности.

2.1.3. Описание входных и выходных данных

Входные данные:

Граф, записанный в следующем формате: в первой строке количество вершин (целое число), каждая последующая *i-*я строка содержит последовательность вершин, инцидентных к (*i-2*)-й вершине. Отсчет вершин начинается с нуля.

Выходные данные:

Визуализация работы алгоритма Косорайю с краткими пояснениями.

2.2. План разработки

Роли в бригаде были распределены следующим образом:

* архитектор-аналитик – Якушкин А.Э.;
* разработчик – Гайдук М.А.;
* тестировщик – Федорова М.Д.

Календарный план разработки приведен в таблице 1.

Таблица 1. Календарный план разработки

|  |  |  |
| --- | --- | --- |
| Дата | Этап работы | Ответственное лицо |
| 21 июня | Согласование темы проекта;  подготовка к работе (создание репозитория, настройка окружения и т.д.);  изучение теоретического материала. | Совместно |
| 22 июня | Составление спецификации проекта и планирование цикла разработки | Якушкин А.Э. |
| 23 июня | Согласование спецификации с руководителем практики и корректировка плана разработки | Якушкин А.Э. |
| 24-25 июня | Разработка графического интерфейса и основных классов | Гайдук М.А. |
| 25 июня | Тестирование и сборка прототипа | Федорова М.Д. |
| 26 июня | Согласование прототипа с руководителем практики | Якушкин А.Э. |
| 27 июня | Завершение реализации класса алгоритма и графа | Гайдук М.А. |
| 27 июня | Тестирование вывода графа на экран | Федорова М.Д. |
| 28 июня | Согласование рабочей версии проекта с руководителем практики | Якушкин А.Э. |
| 29 июня | Отладка проекта | Федорова М.Д. |
| 30 июня | Демонстрация и защита финальной версии | Совместно |