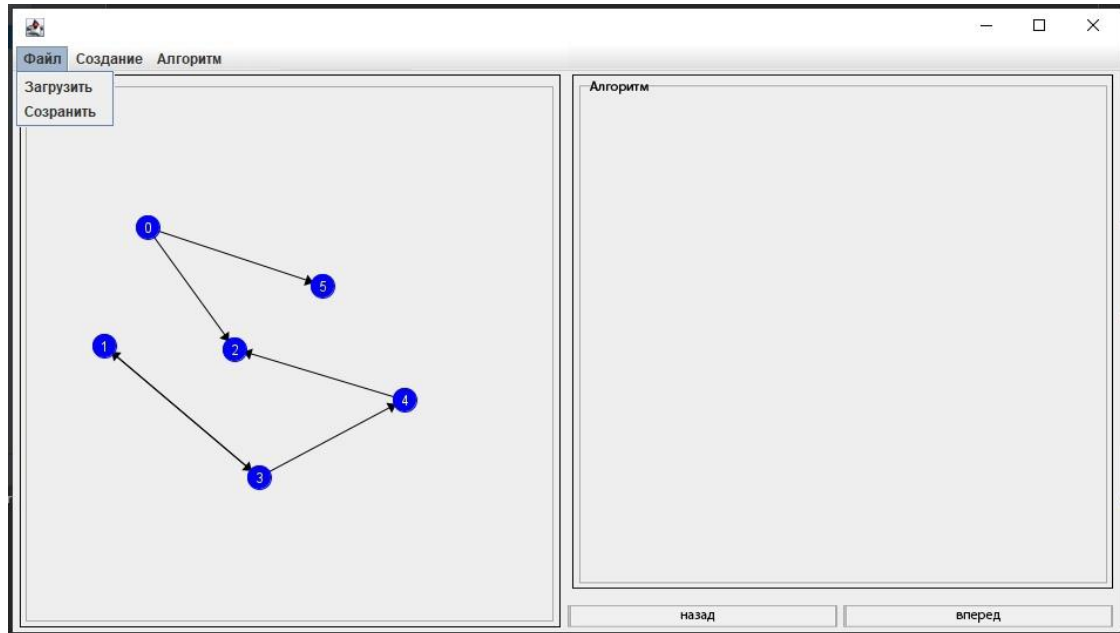


# Топологическая сортировка

## Спецификация

Прототип интерфейса.



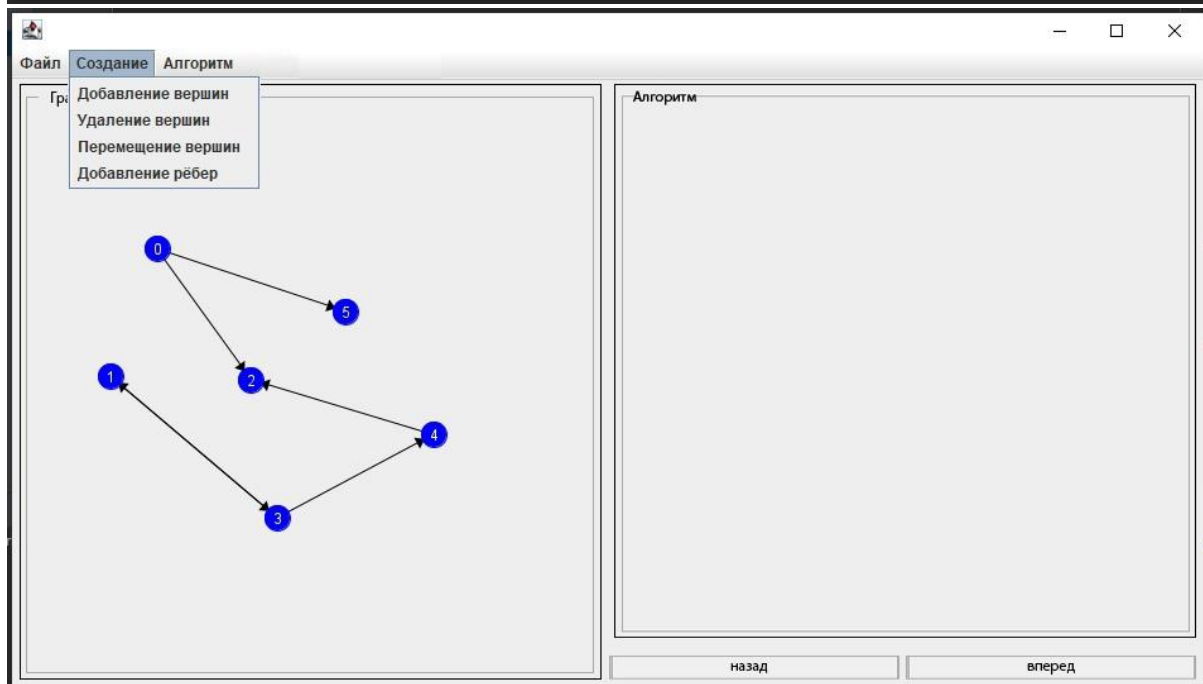
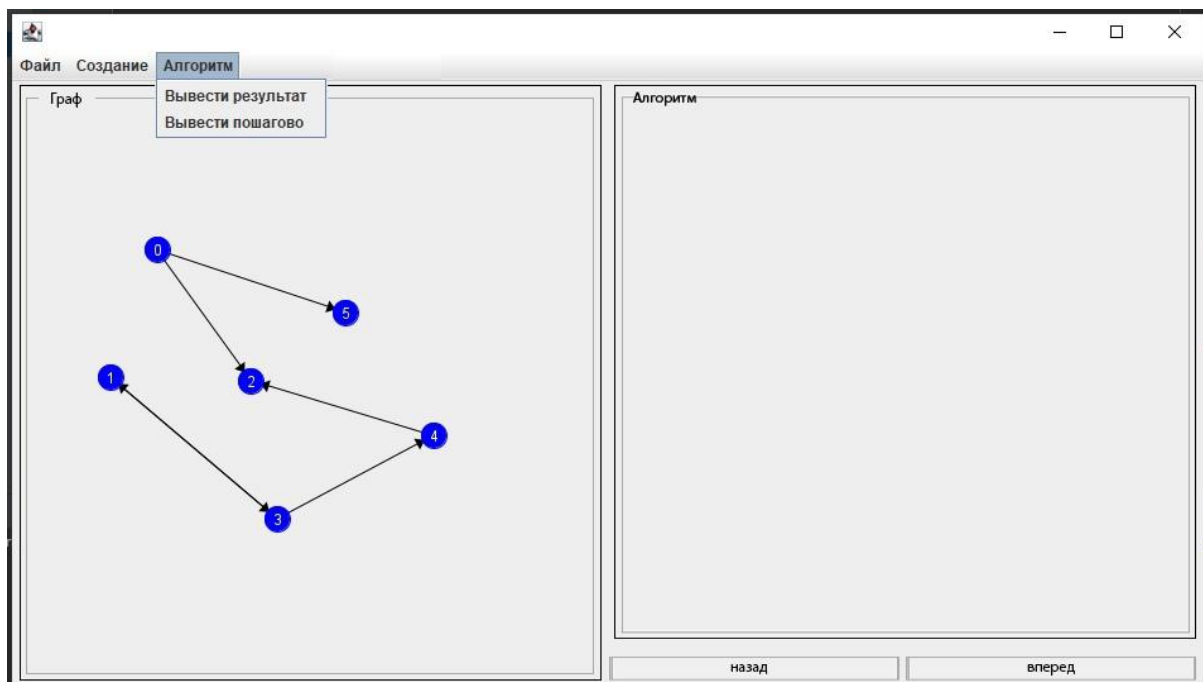


Диаграмма прецедентов.

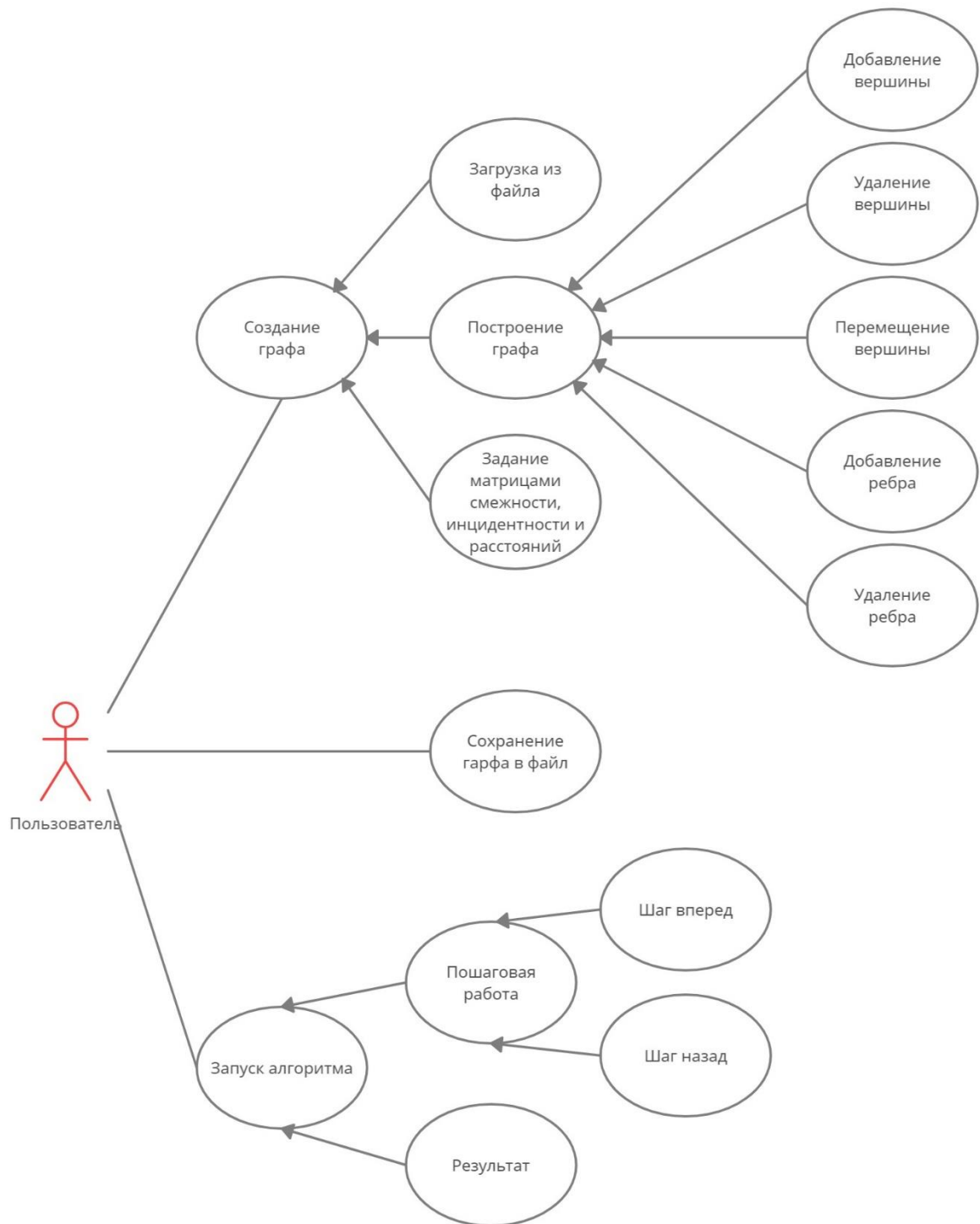
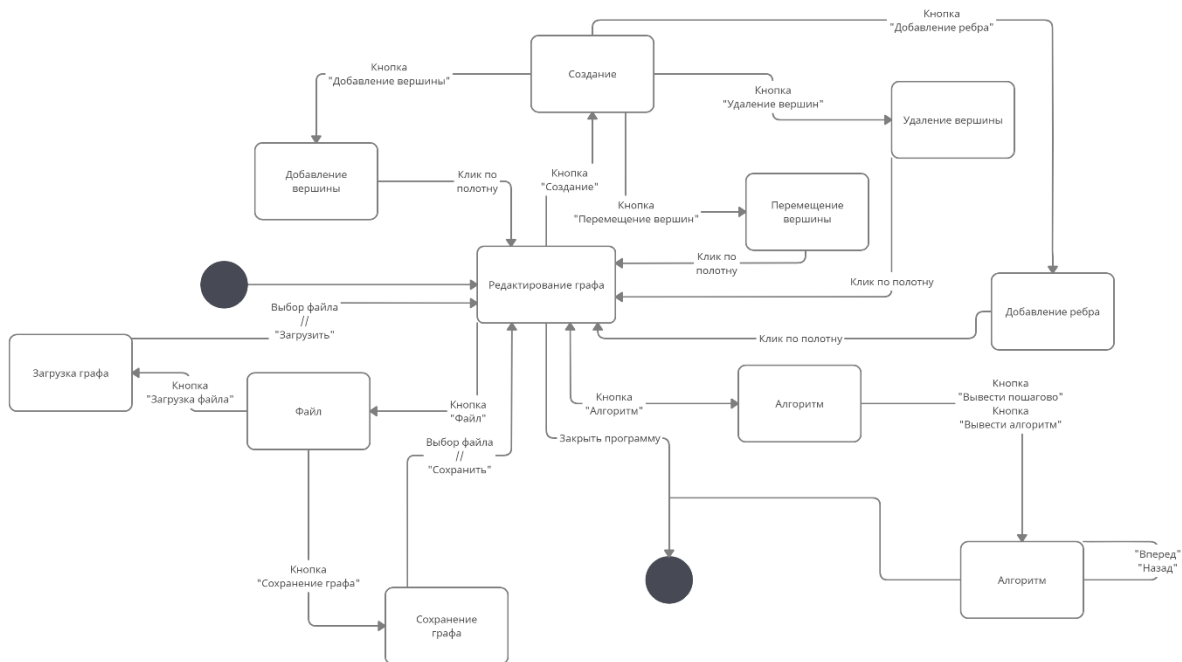


Диаграмма состояний.



## Пояснение к программе.

- После запуска программы нам нужно сделать граф для работы алгоритма. Для этого у нас есть интерфейс (кнопка “Создание”) работы с графом. Пояснение кнопок:
  - “Добавить вершину” - кнопка добавления вершин графа. После нажатия этой кнопки можно добавить нужное количество вершин. Для добавления вершин нужно нажимать мышью на левой панели.
  - “Добавить ребро” - кнопка добавления ребер графа. После нажатия этой кнопки можно соединить добавленные вершины. Для этого нужно зажать кнопку мыши на одну из вершин, которая находится на левой панели, и отпустить на вершине, с которой мы хотим ее соединить.
  - “Перемещение вершины” - кнопка для перемещения вершин. После нажатия этой кнопки можно переместить добавленные вершины. Для этого нужно зажать кнопку мыши на одну из вершин, которая находится на левой панели, и отпустить в месте, которое нужно пользователю.
  - “Удалить вершину” - кнопка удаления вершин графа. После нажатия этой кнопки можно удалить добавленные вершины. Для удаления вершин нужно нажимать мышью на вершины на левой панели.

- После создания графа можно применить алгоритм. Для этого есть кнопка “Алгоритм”, в которой есть кнопки “Вывести результат” и “Вывести пошагово”. Пояснение кнопок:
  - “Вывести результат” - кнопка вывода результата работы алгоритма. После нажатия этой кнопки, на правой панели выведется результат работы алгоритма.
  - “Вывести пошагово” - кнопка пошаговой работы алгоритма. После нажатия этой кнопки, будет продемонстрирована пошаговая работа алгоритма и результат работы алгоритма.
- Для возврата действий или отмены возврата действия в подменю “Редактировать” есть кнопки “Отменить” и “Возвратить” соответственно. Пояснение кнопок:
  - “Отменить” - кнопка отмены последнего изменения графа.
  - “Возврат” - кнопка возврата отмены последнего изменения графа.
- Для того, чтобы загрузить и сохранить граф есть подменю “Файл”, в котором есть кнопки “Загрузить граф” и “Сохранить граф”. Пояснение кнопок:
  - “Загрузить граф” - кнопка загрузки сохраненного графа. Для загрузки графа нужно выбрать имя файла и подтвердить загрузку. В начале файла будут в столбик перечислены имена всех вершин и их координаты, а ниже будет задан граф в виде матрицы смежности. Программа будет считывать построчно значения вида `<name posx posy> >`, где `name` - имя вершины, `posx`, `posy` - её позиция на экране, вызывать метод `Graph::addNode(String name, double x, double y)` - метод позволяет создать вершину в графе, после выполнения действий выше, программа считает значения матрицы смежности, которая располагается ниже и запишет их в двумерный массив, на основе которого, исходя из определения матрицы смежности и при помощи метода `Graph::addEdge(String firstNode, String secondNode)` - метод позволяет построить ребро между вершинами `firstNode` и `secondNode`, можно будет создать рёбра в графе. Значения индексов в массиве, будут соответствовать позициям вершин в списке вершин, т.е. в строке массива с индексом 0, будет находиться информация о соседях самой верхней вершины в файле.
  - “Сохранить граф” - кнопка сохранения графа. Для сохранения графа нужно выбрать файл сохраненного файла для его

перезаписи или написать новое название сохранения, а затем подтвердить сохранение графа. Программа будет в цикле `foreach()` обходить все вершины графа, записывать их в файл в следующей форме, `<name posx posy>`, где `name` - имя вершины, `posx`, `posy` - её позиция на экране. Параллельно с этим используя метод `Graph::getChilids(Node node)` - метод позволяет получить список соседей у указанной вершины `node`, будет создаваться массив, представляющий из себя матрицу смежности и затем он будет перенесен в конец файла.

### **Подробности пошаговой работы алгоритма.**

После запуска алгоритма на панель интерфейса появится таблица со значениями количества входящих в вершины ребер. Она будет заполняться каждый раз, когда мы будем удалять вершину в графе. Когда не останется ни одной вершины в графе, на панели интерфейса будет нарисован топологически отсортированный граф.

### **Распределение ролей:**

1. Николаев Александр (9381) - Разработка пользовательского интерфейса отвечающего за создание (Кнопки создания графа, редактирования, работы алгоритма, привязка к этим компонентам функционал). Визуализация графа.
2. Прашутинский Кирилл (9381) - Создание графа, как некоторой структуры данных и реализация самого алгоритма. Визуализация работы алгоритма.

### **План разработки:**

- Прототип (05.07.2021)
  - Распределение ролей
  - Составление спецификации
  - Создание прототипа пользовательского интерфейса
  - Разработка интерфейса позволяющего построить граф
- 1-я Итерация (07.07.2021)

- Реализация алгоритма и вывод результата работы алгоритма в приложение
- 2-я Итерация (09.07.2021)
  - Добавление возможность пошагового исполнения алгоритма
  - Добавление возможности возврата исполнения алгоритма к предыдущим шагам
  - Добавление возможности записи/чтения графа из файла
  - Тестирование приложения