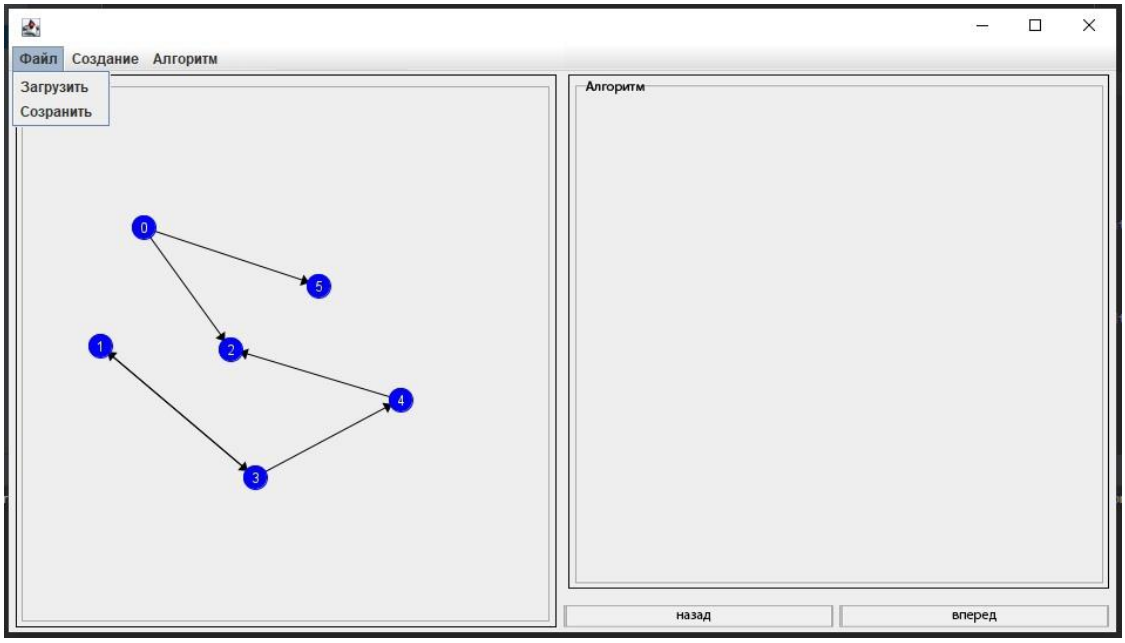
Топологическая сортировка

Спецификация

Прототип интерфейса.



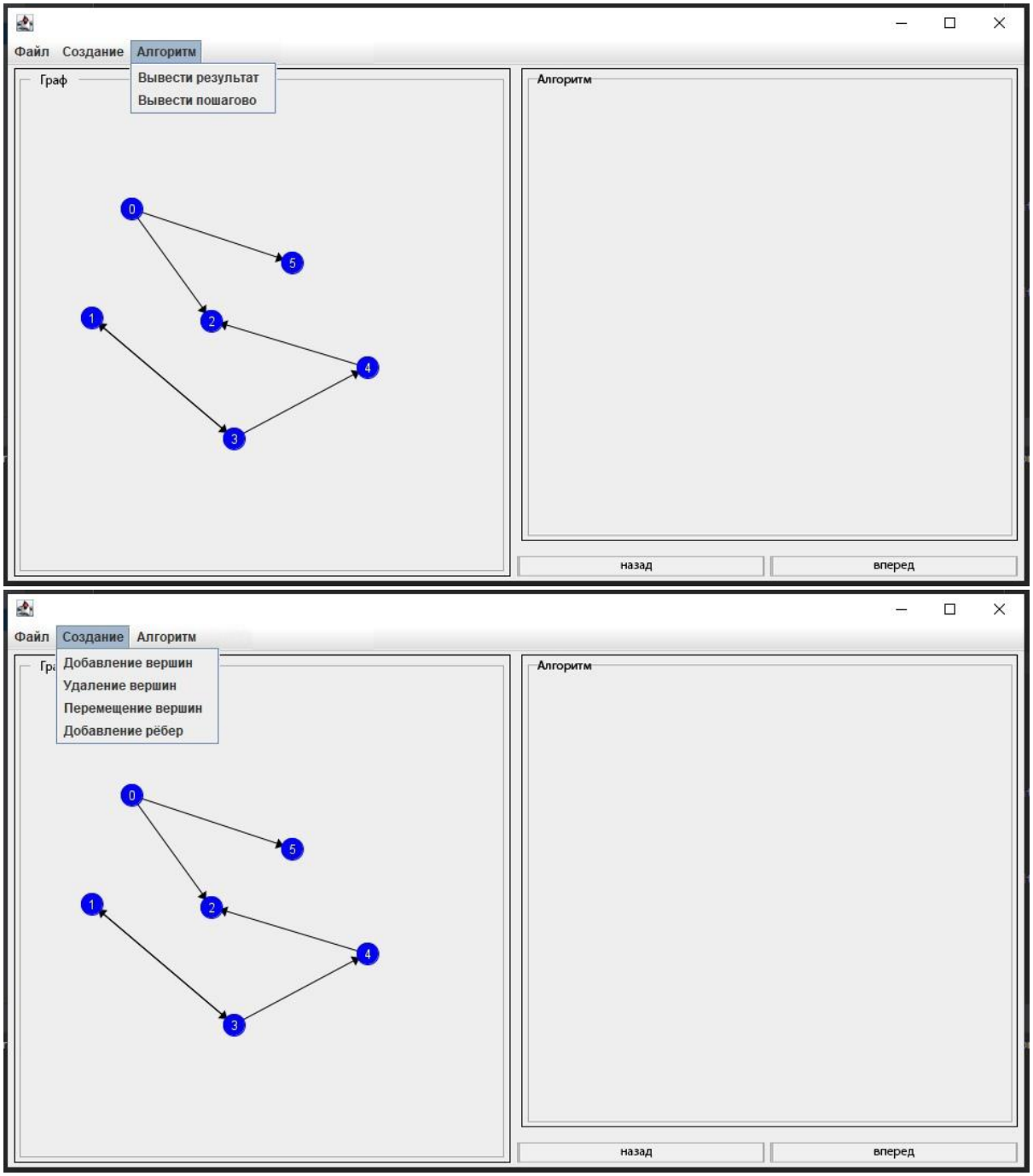


Диаграмма прецедентов.

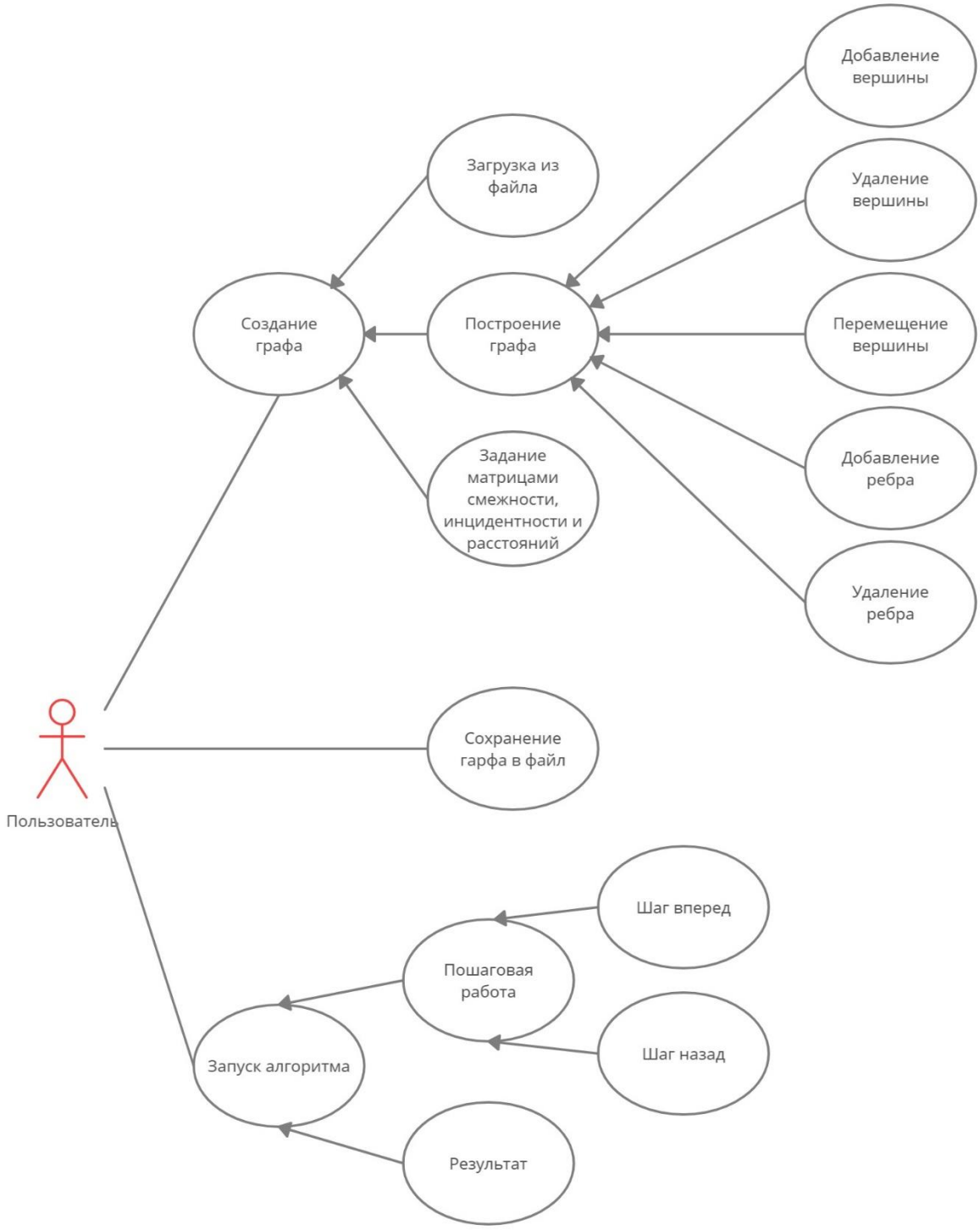
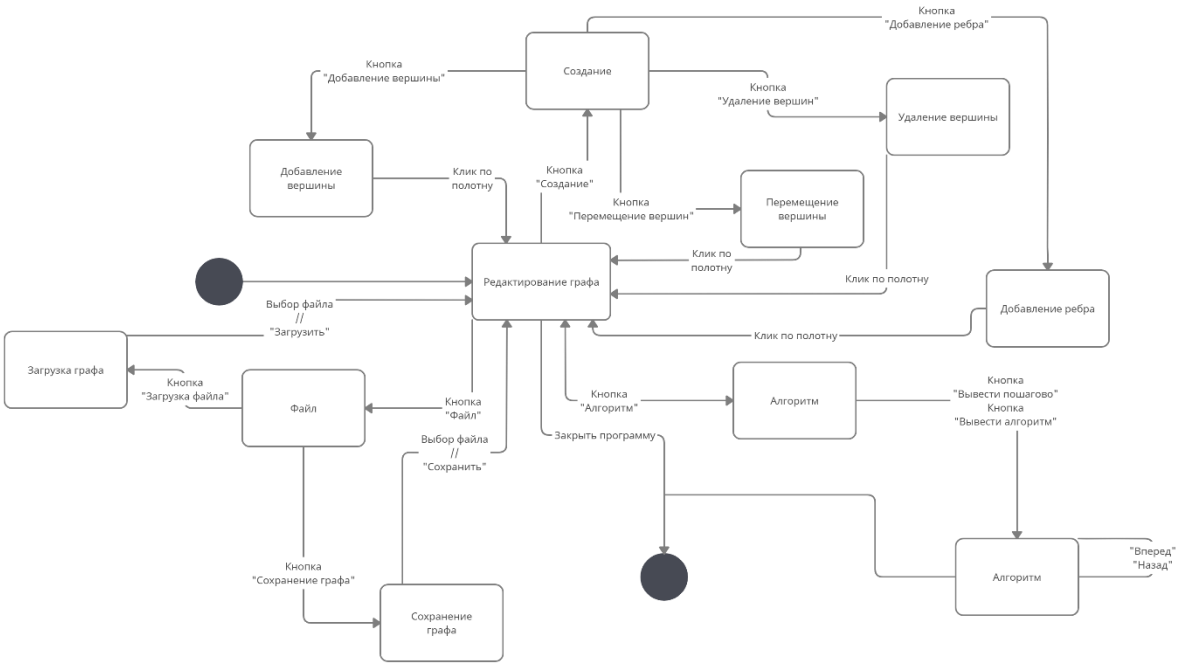


Диаграмма состояний.



Пояснение к программе.

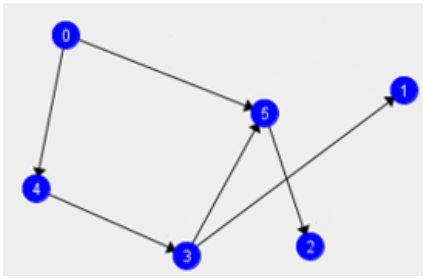
- После запуска программы нам нужно сделать граф для работы алгоритма. Для этого у нас есть интерфейс (кнопка “Создание”) работы с графом. Пояснение кнопок:
    - “Добавить вершину” - кнопка добавления вершин графа. После нажатия этой кнопки можно добавить нужное количество вершин. Для добавления вершин 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 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. 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 добавления. Значения индексов в массиве, будут соответствовать позицям 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 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 получения списка соседей у указанной вершины. Так 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 с нулевым вхождением ребер в этот граф вместе со смежными ребрами. Когда не останется ни одной вершины в графе, на панели интерфейса будет нарисован топологически отсортированный граф.

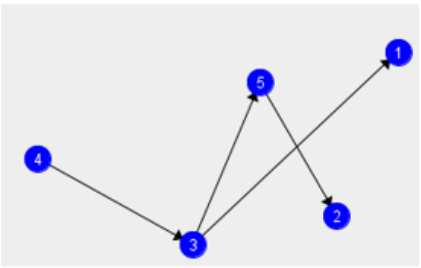
Шаг 1. Заполняем таблицу количеством входящих ребер



Этим цветом помечен количество вершин, входящих в вершину

шаг\название вершин	0	1	2	3	4	5
1	0	1	1	1	1	2
2						
3						
4						
5						
6						
7						

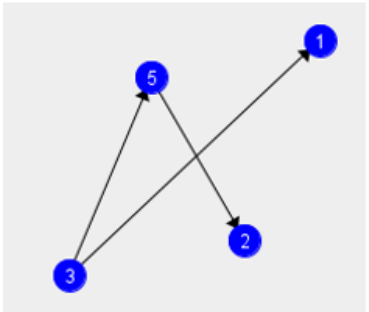
Шаг 2. Удаляем вершину «0» т.к. у нее нулевое количество вхождений. Уменьшаем значения количества вхождений у вершин с названиями «4» и «5».



Этим цветом помечен количество вершин, входящих в вершину

шаг\название вершин	0	1	2	3	4	5
1	0	1	1	1	1	2
2	-	1	1	1	0	1
3						
4						
5						
6						
7						

Шаг 3. Удаляем вершину «4» т.к. у нее нулевое количество вхождений. Уменьшаем значение количества вхождений у вершины с названием «5».



Этим цветом помечен количество вершин, входящих в вершину

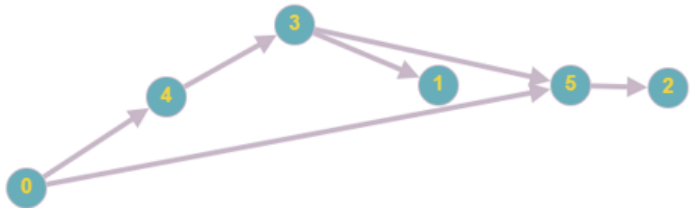
шаг\название вершин	0	1	2	3	4	5
1	0	1	1	1	1	2
2	-	1	1	1	0	1
3	-	1	1	0	-	1
4						
5						
6						
7						

Шаг 4. Заполняем всю оставшуюся таблицу по тому же принципу. Если будет несколько вершин с нулевым количеством вхождений, то удаляем любую из «нулевых» вершин

Этим цветом помечен количество вершин, входящих в вершину

шаг\название вершин	0	1	2	3	4	5
1	0	1	1	1	1	2
2	-	1	1	1	0	1
3	-	1	1	0	-	1
4	-	0	1	-	-	0
5	-	-	1	-	-	0
6	-	-	0	-	-	-
7	-	-	-	-	-	-

Результат:



Пояснения в ходе выполнения алгоритма.

1. Будут пояснения почему мы удаляем конкретную вершину.
2. Будет подсвечиваться вершина, которая будет удалена следующей.
3. Если граф невозможно будет отсортировать, то будет выведена причина, по которой это невозможно сделать.

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:

- Согласование
  - Распределение ролей
  - Составление спецификации
- Прототип (06.07.2021)

    - 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