

Задача

Реализовать на языке программирования Java программу, представляющую собой визуализатор алгоритма нахождения компонент сильной связности орграфа.

Описание алгоритма

Используется алгоритм Kosaraju (Косарайю).

Алгоритм:

- 1) обходом в глубину рассчитывается порядок выхода вершин исходного графа.
- 2) исходный граф транспонируется.
- 3) выполняется обход в глубину в транспонированном графе в обратном порядке выхода вершин.
- 4) в течение обхода в глубину формируются компоненты сильной связности.

1. ТРЕБОВАНИЯ К ПРОГРАММЕ

1.1. Исходные требования к программе

1.1.1 Требования к визуализации

Библиотека: JavaFX

Элементы интерфейса: Холст для графа, панель управления, лог с текстовым выводом шагов алгоритма.

Граф: Граф строго ориентирован, веса не используются, кратные ребра не создаются, максимальное количество вершин 15.

Элементы управления холстом:

Кнопки:

- «Загрузить граф» — загружает граф из файла формата JSON.
- «Скачать граф» — скачивает граф в файл формата JSON
- «Добавить вершину» — добавляет вершину на холст нажатием левой кнопки мыши; вершины автоматически нумеруются при создании.
- «Добавить ребро» — добавляет ориентированное ребро на холст нажатием левой кнопки мыши — сначала по начальной вершины, потом по конечной.
- «Очистить граф» — удаляет полностью граф с холста.
- «Удалить элемент» — удаляет с холста элемент, выбранный нажатием левой кнопки мыши.

Элементы управления алгоритмом:

Кнопки:

- «СТАРТ / ПАУЗА» — если алгоритм еще не запущен, запускает визуализацию работы алгоритма и вывода поясняющего текста в отдельной области, в ином случае приостанавливает визуализацию алгоритма, повторное нажатие возобновляет работу.

- «Перейти к результату» — сразу показывает результат полной работы алгоритма (без пошаговой визуализации алгоритма).

Визуализация алгоритма:

Пройденные ребра отмечаются зеленым цветом. Выход из вершины помечается номером в стеке.

В конце работы вершины компоненты сильной связности будут выделены соответствующим номером.

Примерный вид интерфейса представлен на рис.1:

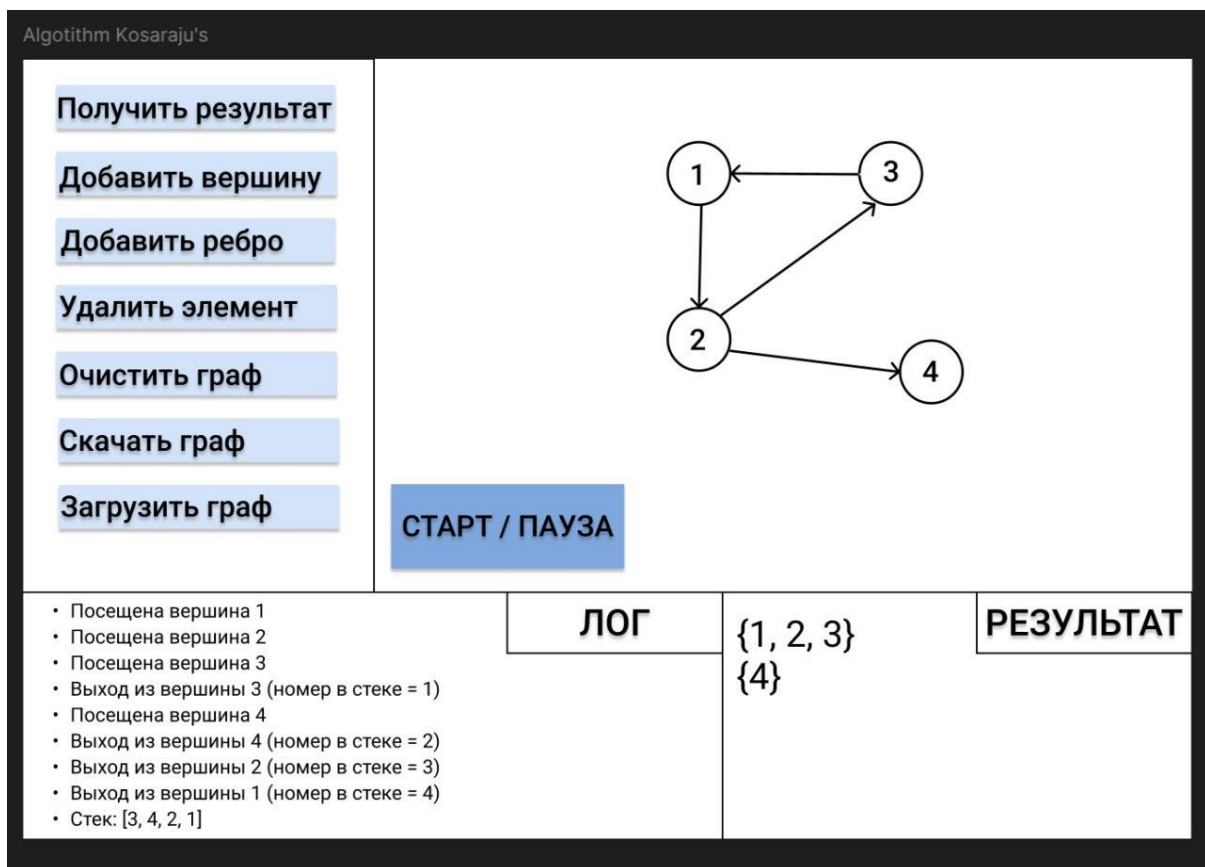


Рисунок 1: Примерный дизайн интерфейса

1.1.2 План тестирования

Обработка ошибок:

- Работа с файлом: некорректный файл, файл не формата JSON.

- Работа алгоритма: пустой граф.
- Работа интерфейса: добавление вершин больше допустимого значения, создание нескольких одинаковых ребер из одной вершины в другую

Проверка корректности работы интерфейса: Нажатие кнопок, расширение экрана, работа холста, работа текстового поля.

Проверка работы алгоритма на различных графах: граф из одной вершины, граф с несколькими компонентами сильной связности, несвязный граф, полный граф, граф с петлями.

2. ПЛАН РАЗРАБОТКИ И РАСПРЕДЕЛЕНИЕ РОЛЕЙ В БРИГАДЕ

2.1 План разработки

Дата	Этап проекта	Реализованные возможности	Выполнено
29.06.25	Согласование плана и спецификации	Написание требований и плана	
01.06.25	Сдача прототипа	Разработка прототипа интерфейса и написание алгоритма	
02.06.25	Сдача версии 1		
04.06.25	Сдача версии 2		
06.06.25	Сдача версии 3		
06.06.25	Сдача отчёта		

2.2 Распределение ролей

Разработка алгоритма и его внедрение в интерфейс: Кривошеина Дарья

Разработка интерфейса: Матвеев Никита