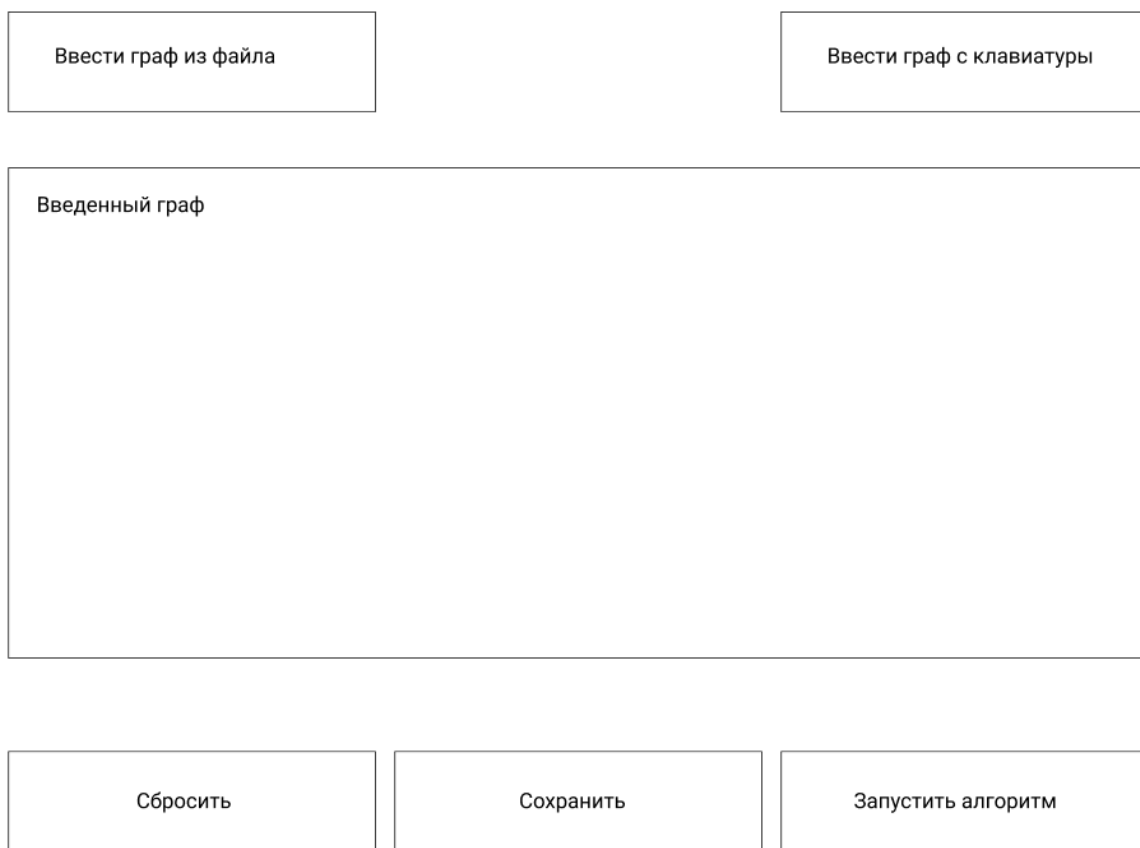


Итерация №1

1. Эскиз GUI

На вход программе подается ориентированный граф, который задается в текстовом виде через диалоговое окно или из файла. После завершения алгоритма полученные компоненты сильной связности отображаются на главном окне, также присутствует возможность записи полученного результата в текстовый файл. В ходе выполнения алгоритма пользователь может видеть предыдущее и текущее состояние программы.

Главное окно программы (рисунок 1) представляет собой окно, содержащее 2 кнопки для ввода графа: ввод из файла (открывается диалоговое окно), ввод с клавиатуры (открывается окно с полем для ввода текста). При неправильном формате ввода программа показывает сообщение с ошибкой, пользователь может ввести граф еще раз. Также окно содержит кнопку сохранения графа в файл и кнопку запуска алгоритма, по нажатию на которую открывается окно, отображающее состояние программы, главное окно скрывается. Также окно поле, на котором визуализируется введенный граф.



The image shows the main menu of a program. It consists of a central rectangular area labeled "Введенный граф" (Entered graph) which is currently empty. Above this area are two buttons: "Ввести граф из файла" (Load graph from file) on the left and "Ввести граф с клавиатуры" (Load graph from keyboard) on the right. Below the central area are three buttons: "Сбросить" (Reset) on the left, "Сохранить" (Save) in the middle, and "Запустить алгоритм" (Run algorithm) on the right.

Рис 1. Главное меню программы

Окно, отображающее состояние программы (рисунок 2), содержит два поля, в которых отображается состояние алгоритма до и после выполнения текущего шага. В нижней части поля вывода предыдущего состояния расположено поле для вывода комментариев лога работы алгоритма. В правой части окна содержатся кнопки “Шаг вперед” и “Шаг назад”, по нажатию на которые программа переходит к следующему шагу алгоритма или возвращается на шаг назад. Также есть кнопка “Прервать алгоритм”, по нажатию на которую закрывается окно с состоянием программы и отображается главное окно, при этом поле с результатом работы остается пустым. В силу специфики алгоритма добавлен лог топологической сортировки вершин по времени выхода.

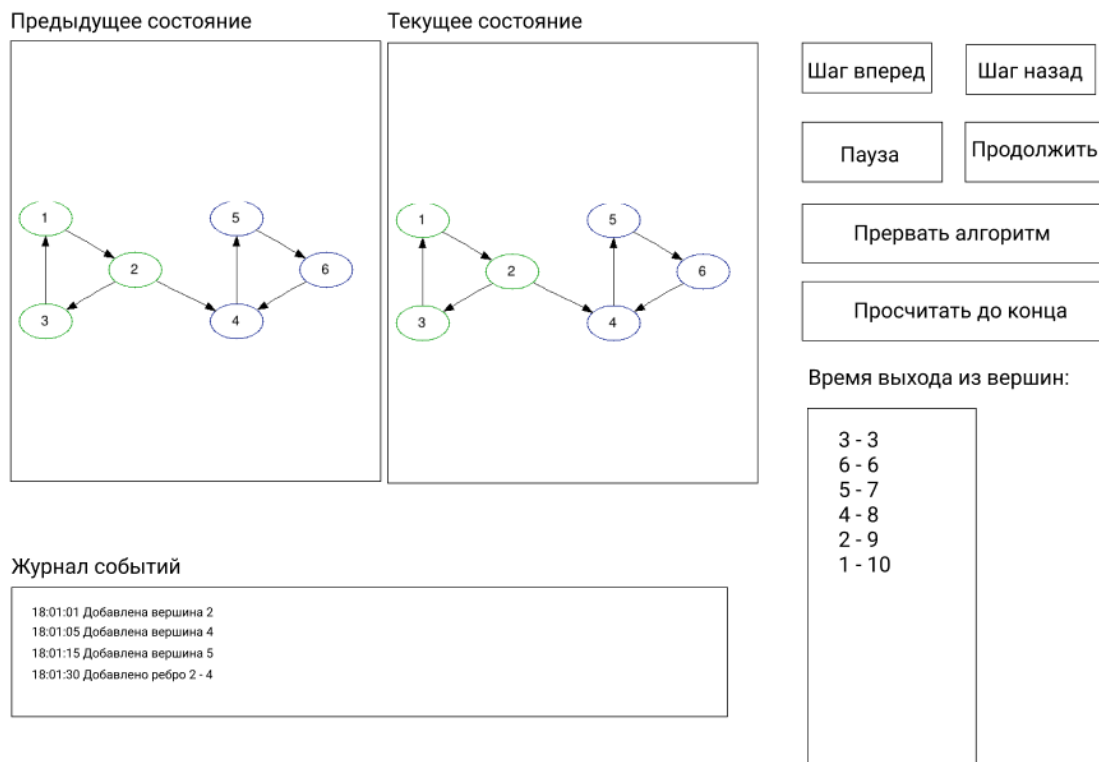


Рис 2. Окно исполнения алгоритма

2. Описание архитектуры

В основе архитектуры приложения взята модель MVC, обеспечивающая высокий уровень абстрагирования, а следовательно надежности, удобочитаемости и простоты кода и разработки. Ниже представлены некоторые uml-диаграммы, характеризующие логику и архитектуру построения приложения.

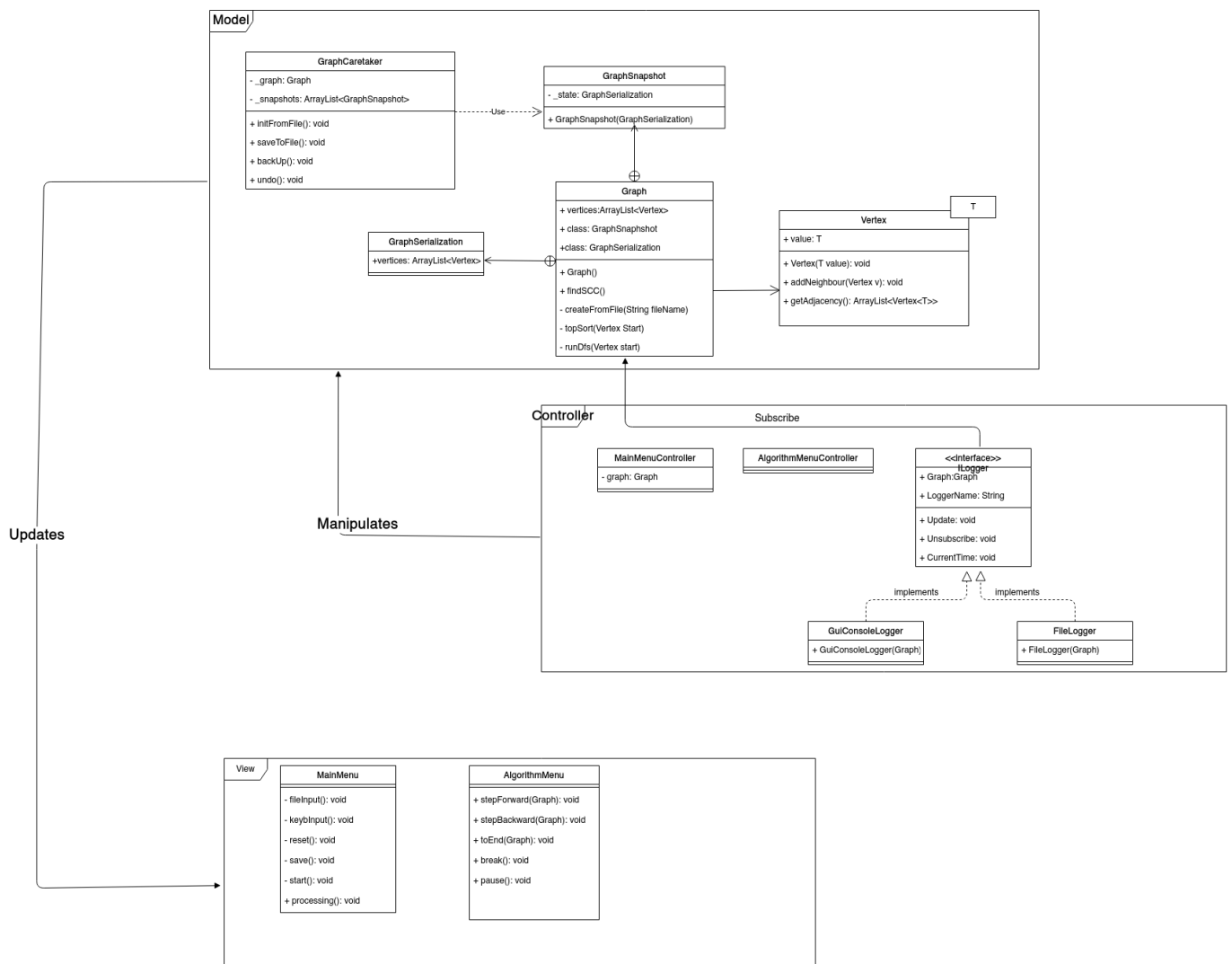


Рис 3. Uml - диаграмма классов

Диаграмма состояний

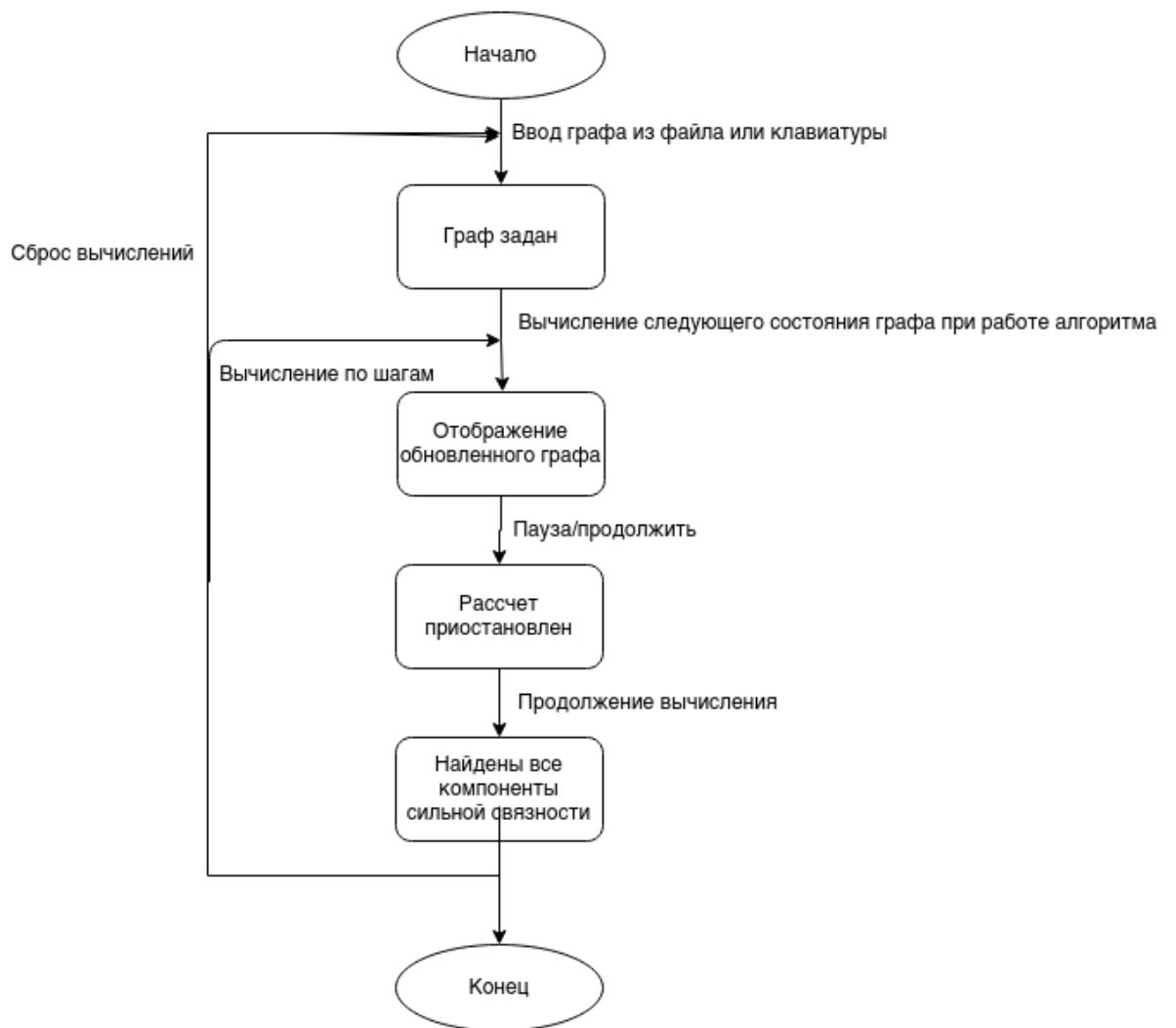


Рис 4. Uml - диаграмма состояний

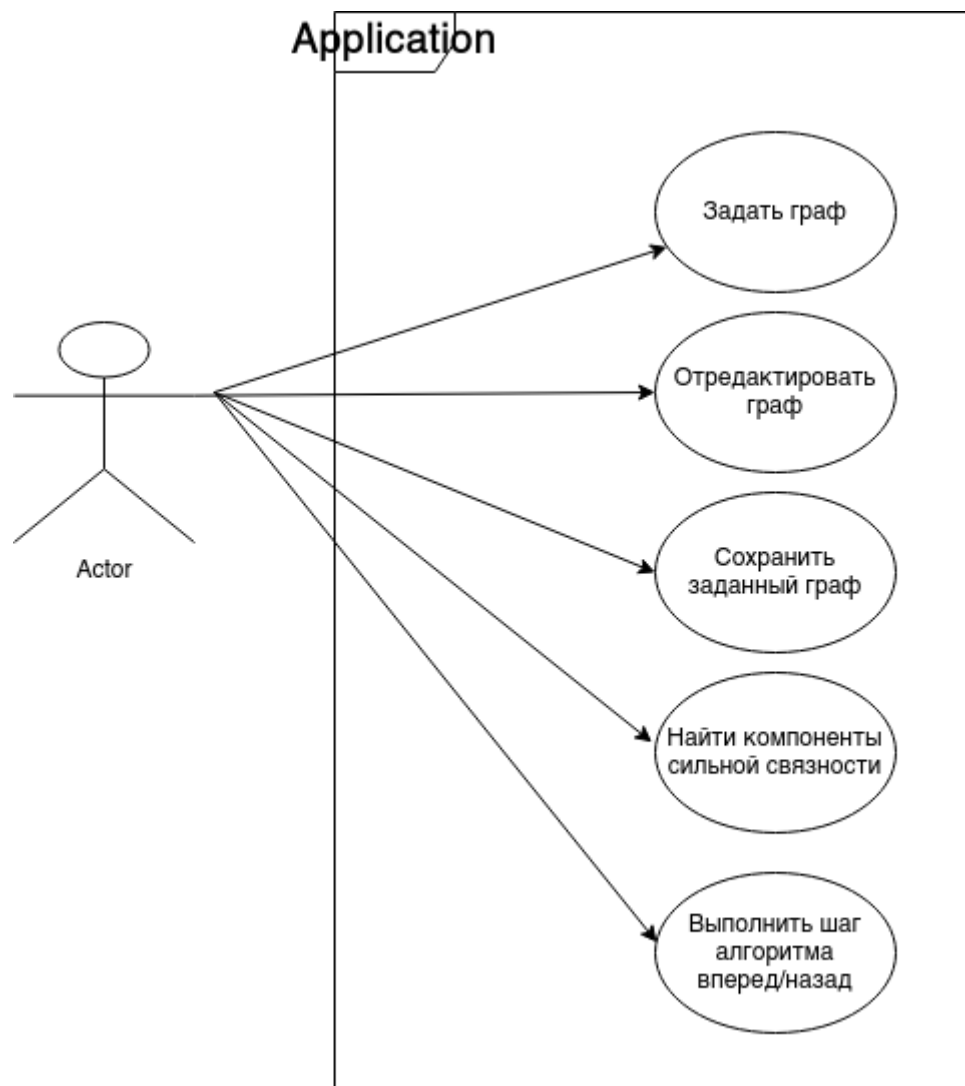


Рис 5. Uml - диаграмма возможностей