

### Описание use-case диаграммы:

Пользователь, посредством взаимодействия с графическим интерфейсом, может совершать ряд действий:

- 1) Создание графа с помощью ввода матрицы смежности.
- 2) Создание графа с помощью его случайной генерации с параметрами : количество ребер, количество вершин, минимальный вес, максимальный вес.
- 3) Добавление вершины в граф при нажатии на кнопку.
- 4) Добавление ребра при вводе инцидентных ребру вершин и веса и нажатии кнопки.
- 5) Добавление ребра от выбранной вершины к другой вершине, при помощи нажатия на нужную вершину правой кнопкой мыши и выборе нужного пункта меню.
- 6) Изменение веса ребра при нажатии на него правой кнопкой мыши и выборе нужного пункта меню.
- 7) Удаление ребра при помощи нажатия на него правой кнопкой мыши и выборе нужного пункта меню.
- 8) Удаление вершины при помощи нажатия на нее правой кнопкой мыши и выборе нужного пункта меню.
- 9) Просмотреть вес ребра, нажав на него левой кнопкой мыши.
- 10) Начать работу алгоритма, нажав кнопку "START" и пошагово просматривать его ход с помощью стрелочек.
- 11) Перейти к результату алгоритма, нажав на кнопку "TO END".
- 12) Завершить алгоритм, нажав на кнопку "STOP".

### Описание uml-диаграммы классов:

- 1) Класс Graph является реализацией графа. Граф хранится в виде набора ребер и набора вершин. В классе Graph реализованы функции для работы с ним.
- 2) Класс Edge является реализацией ребра.
- 3) Класс GraphGenerator генерирует по заданным параметрам связный граф. Для того чтобы граф был связный реализован вспомогательный класс Component, который позволяет хранить и объединять наборы вершин.
- 4) Класс Gui является реализацией графического интерфейса. В данном классе реализованы все необходимые элементы, через которые пользователь может общаться с приложением.
- 5) С помощью классов VertexVisualization и GraphVisualization отрисовываются вершины и ребра.
- 6) Класс GraphVisualization является графическим представлением графа. GraphVisualization располагает вершины равномерно по окружности. Класс реализовывает интерфейс MouseListener и дает возможность прослушивать нажатия мыши пользователем.
- 7) Класс Controller является связующим звеном между графикой и логикой приложения. В классе добавлены ряд классов-слушателей, которые реагируют на события пользователя при нажатии кнопок.
- 8) Классы EdgeHandler и VertexHandler реализуют интерфейс ElementHandler и представляют из себя обработчиков нажатия пользователя по элементам графа (вершинам и ребрам).
- 9) Класс Boruvka является реализацией алгоритма. С помощью класса BoruvkaSteps реализован паттерн «Снимок», на каждой итерации алгоритма создается снимок и

добавляется в BoruvkaSteps. Затем при нажатии пользователем стрелочек производится перемещение по снимкам.

- 10) Класс CheckingCorrect представляет различные функции для проверки корректности ввода.
- 11) Класс Converter представляет функции преобразования строкового ввода в необходимый формат.