**Спецификация**

Алгоритм Борувки работает с графом и предназначен для нахождения минимального остовного дерева.

Для работы с данной программой пользователь должен обладать начальными знаниями в теории графов.

*Постановка задачи*

Требуется разработать программу на языке Java, визуализирующую работу алгоритма Борувки для неориентированного неотрицательного графа. При этом должен быть реализован графический интерфейс.

*Ввод-вывод*

Входные данные

1. Количество вершин - натуральное число
2. Координаты расположения вершины (автоматически определяются при создании пользователем вершины в области активного поля)
3. Вес ребер - положительные рациональные числа

Выходные данные

1. Пошаговая работа алгоритма - вывод промежуточных результатов, представленных в виде компонент связности
2. Вывод итогового остовного дерева
3. Подсказки пользователю и сообщения об ошибках

Подсказки пользователю и сообщения об ошибках:

* «Введите количество вершин» (появляется сразу при запуске)
* «Вы ввели некорректные данные, количество вершин должно быть натуральным числом. Повторите ввод»
* «Если Вы хотите создать ребро, выделите 2 вершины» (появляется, когда введено нужное количество вершин)
* «Теперь введите вес ребра»
* «Вы ввели некорректные данные, вес ребра должен быть положительным рациональным числом. Повторите ввод»
* «Алгоритм в процессе выполнения» (всегда высвечивается, пока не дойдем до последнего шага алгоритма)
* «Алгоритм закончил свою работу. Получено минимальное остовное дерево»

*Особенности реализации*

Для проекта потребуется визуализация неориентированного взвешенного связного графа. Для составления графа будут использованы параметры, приведенные в таблице 1.

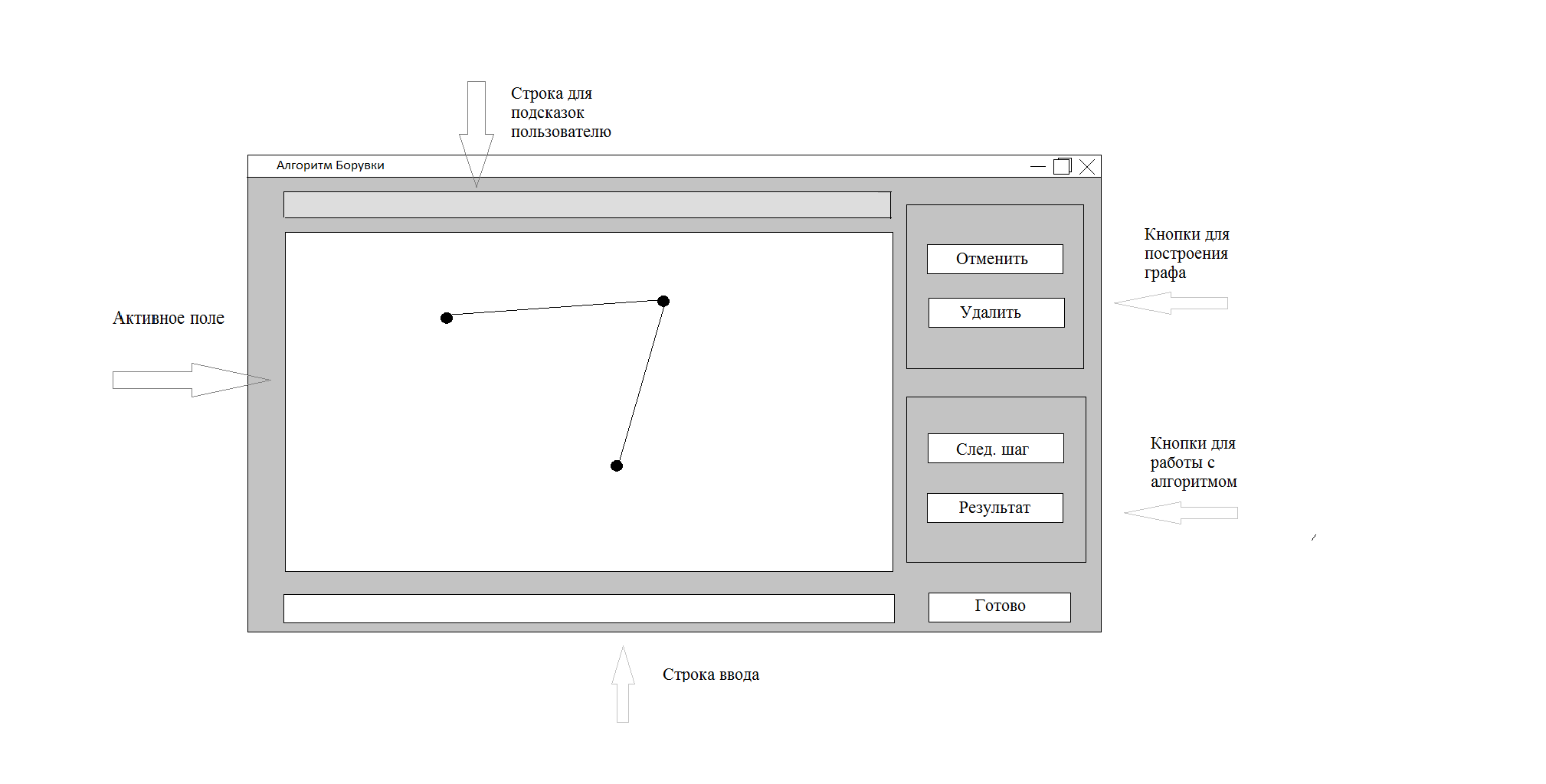
|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип** | **Назначение** |
| vert | Vertex | Вершина |
| edge | Edge | Ребро |
| weight | double | Вес ребра |
| txtWeight | Text | Подпись веса ребра |
| vertIsCheck | boolean | Отмечено ли ребро |

*Таблица 1 "Составляющие для визуализации графа "*

Для конструирования исходного графа пользователь в строке ввода должен ввести количество вершин. Далее кнопкой мыши в области активного поля будут создаваться вершины. У пользователя нет возможности создать дополнительную вершину, выходящую за рамки допустимого количества вершин, т.к. активное поле не будет в активном состоянии. Создав нужное количество вершин, пользователь может создать ребра. Для этого он должен выделить 2 вершины, которые надо соединить, и в строке ввода задать вес нового ребра.

После составления изначального графа запускается сам алгоритм. Пользователь сможет отследить его работу пошагово. В самом начале останутся вершины из исходного графа, ребра пропадут. Дальше будут добавляться минимальные ребра к каждой вершине, постепенно образуя компоненты связности. Компоненты соединят минимальные из оставшихся ребер. В итоге получится минимальное остовное дерево, являющееся результатом работы и показанное пользователю.

*Интерфейс*



*Рисунок 1 "Интерфейс"*

При запуске проекта будет создаваться главное окно. Главное окно будет состоять из активного поля (где будет вводиться/выводится граф), подсказок пользователю, кнопок, с помощью которых пользователь сможет работать с графом, и строки ввода.

Кнопки:

* Готово (при нажатии добавляет информацию: количество вершин, вес ребра, запуск алгоритма)
* Отменить (отменяет последнее действие)
* Удалить (полностью очищает граф)

После создания графа и нажатия кнопки «Готово» начнется работа алгоритма, и пользователь сможет использовать другие кнопки.

Кнопки:

* След. шаг (переход к следующему шагу алгоритма)
* Результат (переход сразу к последнему шагу алгоритма)

Последнее добавленное ребро по алгоритму будет окрашиваться красным цветом, все остальные ребра будут черные.

*Разделение труда*

Забалуев *–* реализация алгоритма Борувки, разработка приемочных тестов и тестирование.

Васильева *–* разработка графического интерфейса, создание прототипа, возможность ввода графа.

Петрова *–* пошаговая визуализация алгоритма, разработка пояснительной записки.

*План разработки*

*23.06.17 –* согласование спецификации;

*26.06.17 –* сдача прототипа (демонстрация пользовательского интерфейса без реализации алгоритма, возможность ввести граф);

*28.06.17 –* сдача первой версии (визуализация работы алгоритма без пошагового просмотра, сразу вывод результата), проект пояснительной записки;

*30.06.17 –* сдача финальной версии и пояснительной записки.