МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

СПЕЦИФИКАЦИЯ

по учебной практике

Тема: Минимальное остовное дерево: алгоритм Борувки

Студент гр. 0303 _____ Середенков А.А.

Студент гр. 0303 _____ Сологуб Н.А.

Студент гр. 0303 _____ Пичугин М.В.

Руководитель _____ Фирсов М.А.

Санкт-Петербург

2022

1. ТРЕБОВАНИЯ К ПРОГРАММЕ

1.1. Исходные Требования к программе

1.1.1. Требования к вводу исходных данных

Исходные данные могут вводится 2 способами:

1) С помощью взаимодействие пользователя с интерфейсом по нажатию мыши на кнопки и графическое полотно. Пользователь нажимает на клавишу для создание вершин графа, чтобы указать программе, что на графическом полотне по нажатию мыши создается вершина графа. Затем пользователь нажимает на клавишу для создания рёбер, и нажимает на две произвольные вершины на полотне для отрисовки и создания ребра между ними.
2) С помощью чтения данных из файла. Приложения сначала проверяет исходный файл на правильность заполнения. Если все верно, то рисуется введенный граф, иначе выведется окно с ошибкой(“WRONG FILE!!!!”).

1.1.2. Требования к визуализации

В левой части программы будет большая область на которой будет отображаться граф, в правом верхнем углу будут находится все необходимые кнопки для работы с приложением, в правом нижнем углу будет находится консоль для вывода логов.

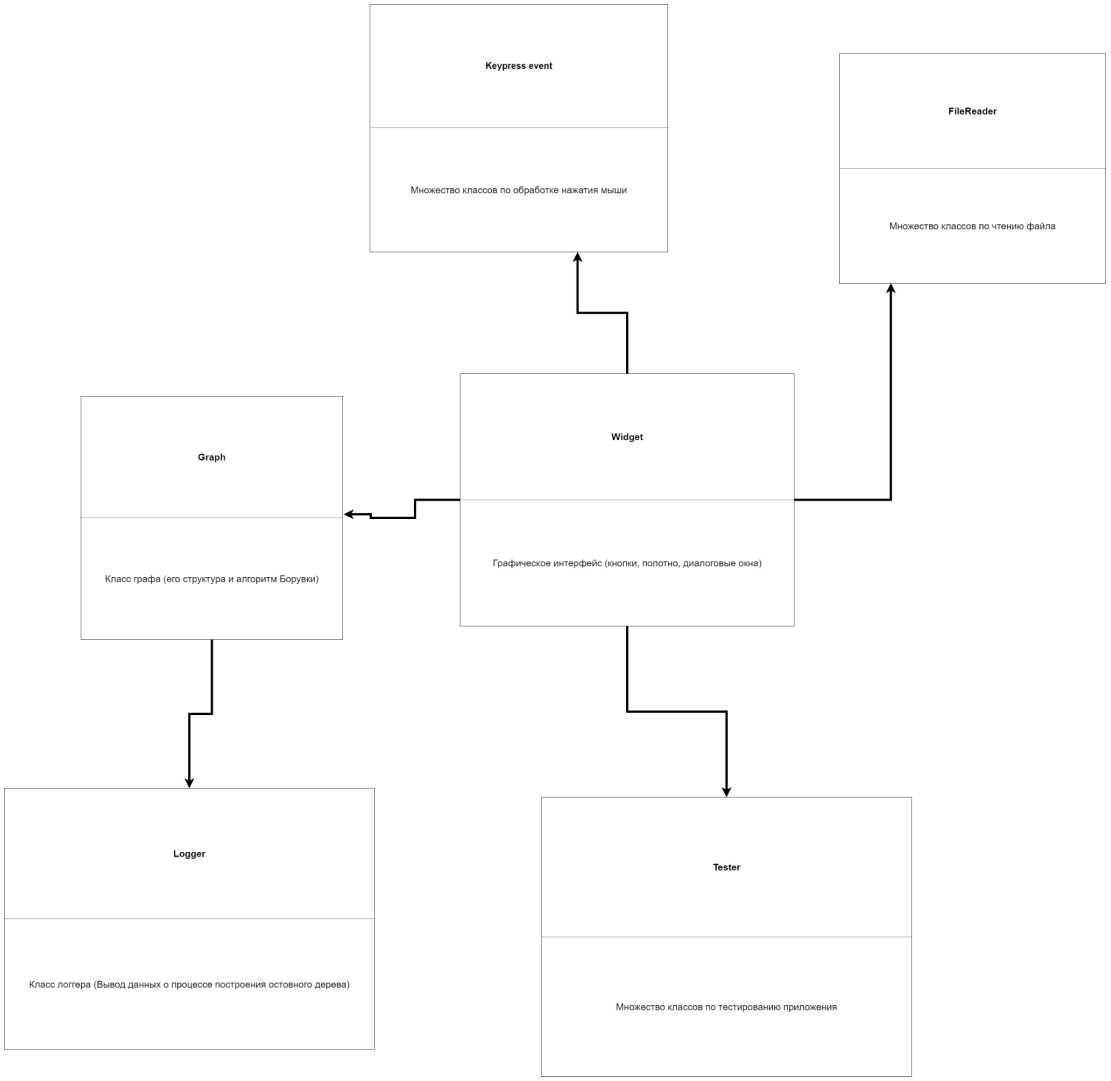
В исходном окне приложения пользователю будет предоставляться выбор для ввода данных: через нажатие кнопки мыши или через файл. Если пользователь выберет ввод данных через файл, то ему необходимо будет нажать соответствующую кнопку после чего откроется новое окно, содержащее поле, в которое необходимо вписать название файла для считывания данных. Также после этого у пользователя будет возможность добавить новые вершины и ребра, с помощью других кнопок. Если же пользователь выберет ввод данных через кнопку мыши, то с помощью других кнопок пользователю будет необходимо добавить все необходимые вершины и ребра с весами.

Элементы графа такие как вершины и ребра будут выглядеть следующим образом:

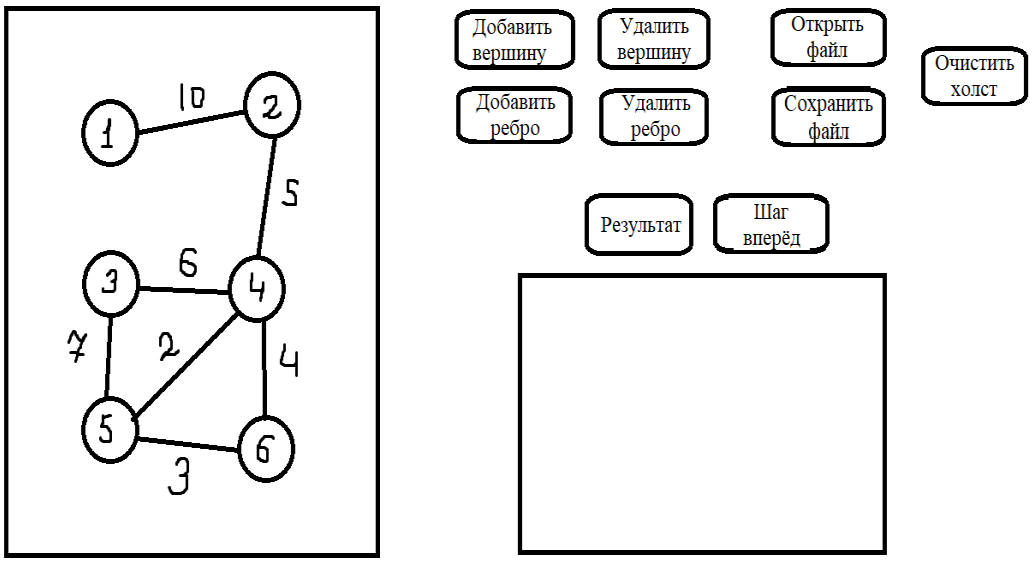
- Вершины выглядят как окружности, внутри которых будут написаны их номера.
- Рёбра представляют собой линии соединяющие вершины, над которыми будет написан их вес.

Процесс алгоритма Борувки будет отображаться через постепенное перекрашивание графа в разные цвета рёбер и вершин, а затем последующее перекрашивание деревьев, полученных на промежуточных результатах.

1.1.3. UML диаграмма



1.1.4. Эскиз приложения



1.1.5. Псевдокод алгоритма

```
// G — исходный граф
// w — весовая функция
function boruvkaMST():
   while T.size<n−1
      for k∈ Component        // Component — множество компонент связности в T. Для
         w(minEdge[k])=∞ //каждой компоненты связности вес минимального ребра = ∞
      findComp(T)        // Разбиваем граф T на компоненты связности обычным dfs-ом.
      for (u,v)∈E
         if u.comp≠v.comp
            if w(minEdge[u.comp])>w(u,v)
               minEdge[u.comp]=(u,v)
            if w(minEdge[v.comp])>w(u,v)
               minEdge[v.comp]=(u,v)
      for k∈ Component
         T.addEdge(minEdge[k])        // Добавляем ребро, если его не было в T
   return T
```