**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

Спецификация

**по дисциплине «Объектно-ориентированные языки и среды»**

Тема: Визуализация алгоритмов сортировки

|  |  |  |
| --- | --- | --- |
| Студенты гр.5304 |  | Калганов Р.Г. |
|  |  | Круглик А.Д. |
|  |  | Русу Я. |
| Преподаватель |  |  |

Санкт-Петербург

2017

Оглавление

[**Требования к программе** 3](#_Toc485921124)

[1. Описание основной задачи программы 3](#_Toc485921125)

[2. Формат входных данных и результат работы программы 3](#_Toc485921126)

[3. Эскизы интерфейса 4](#_Toc485921127)

[4. Общий алгоритм работы программы в соответствии с эскизом интерфейса 6](#_Toc485921128)

[**План разработки** 7](#_Toc485921129)

[**Разделение обязанностей** 8](#_Toc485921130)

# **Требования к программе**

# Описание основной задачи программы

Программа предназначена для визуального представления сортировки перемешиванием. Необходимо реализовать алгоритм данной сортировки, которая представляет собой разновидность [пузырьковой](http://dic.academic.ru/dic.nsf/ruwiki/45663). Отличается она тем, что просмотры элементов выполняются один за другим в противоположных направлениях, при этом большие элементы стремятся к концу массива, а маленькие - к началу.

Программа должна иметь удобный интерфейс и понятный способ визуализации. Для наглядного изображения отличия работы сортировки перемешиванием от других обменных сортировок программа должна предоставлять возможность визуализации других алгоритмов, таких как:

* Пузырьковая сортировка
* Глупая сортировка
* Быстрая сортировка

# Формат входных данных и результат работы программы

На вход программа получает количество и массив элементов. Элементы могут быть заданы пользователем, либо образованы с помощью генерации случайных чисел.

Ограничение значении элементов: от 0 до 1000.

Ограничение количества элементов: от 1 до 65.

В результате работы, программы должна, основываясь на введённой последовательности элементов, визуально изобразить работу сортировки и предоставить отсортированный массив чисел.

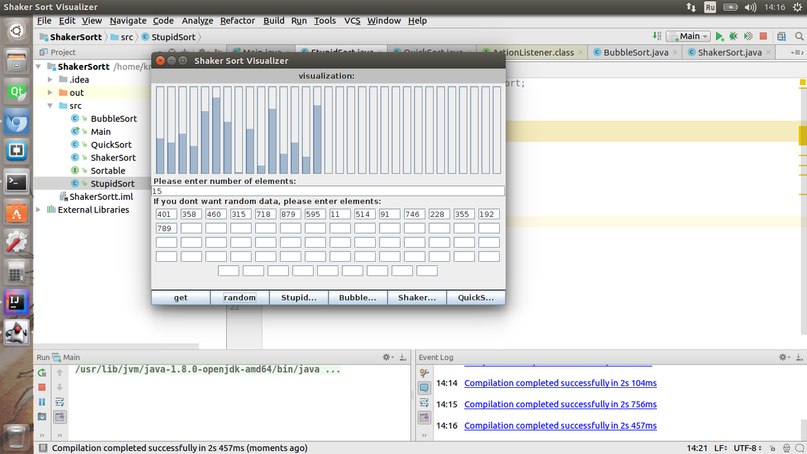
# Эскизы интерфейса

Интерфейс программы должен быть удобен в использовании, поэтому для достижения данной цели принимаются следующие меры:

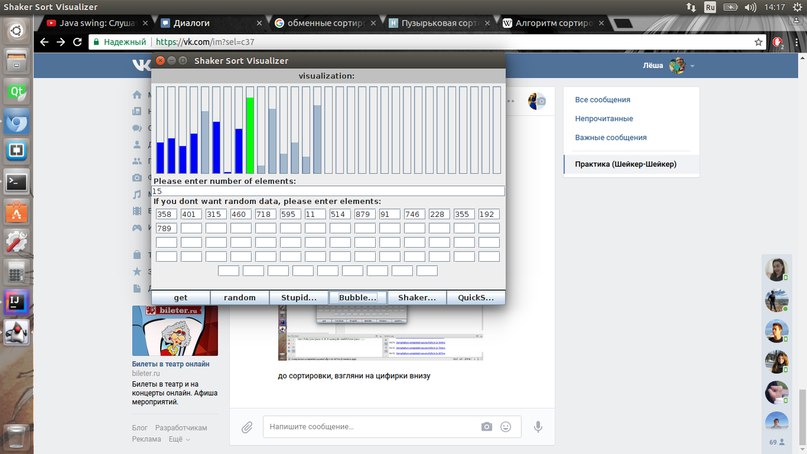
* Каждое окно для ввода данных подписывается по назначению
* Организовывается панель внизу, содержащая кнопки get, которая берёт введённые пользователем элементы, и random, которая отвечает за подбор случайных элементов последовательности
* На панели внизу также располагаются кнопки, запускающие алгоритмы сортировки. Данные кнопки подписаны в соответствии с названием необходимого алгоритма
* Область визуализации организовывается в верхней части основного окна программы и представлена в виде столбиков, количество которых соответствует количеству элементов, а уровень окраски -величине значения элемента на данной позиции

Эскизы интерфейса представлены на рисунках 1 – 4:

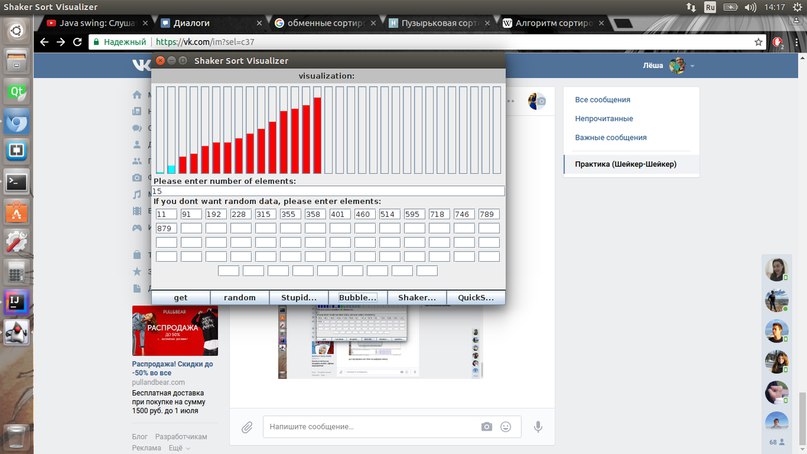
*Рисунок №1: Основное окно программы после ввода количества элементов и генерации последовательности случайных чисел*



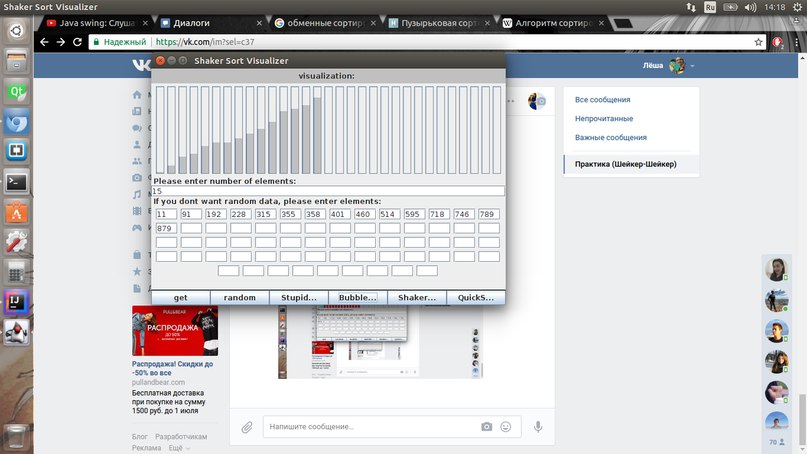
*Рисунок №2: Основное окно программы после выбора алгоритма (в процессе визуализации сортировки)*



*Рисунок №3: Основное окно программы при достижении правильной (отсортированной) последовательности*

**

*Рисунок №4: Основное окно программы после окончания работы данного алгоритма сортировки*

**

# Общий алгоритм работы программы в соответствии с эскизом интерфейса

* Получаем количество элементов для сортировки
* Получаем начальные значения с помощью генерации случайных чисел, либо при помощи ввода необходимых значений пользователем
* Исходя из начальных значений генерируем массив элементов и ожидаем команды от пользователя
* При нажатии пользователем кнопки:
  + - «Stupid sort» запускаем глупую сортировку
    - «Bubble sort» запускаем сортировку пузырьком
    - «Shaker sort» запускаем сортировку перемешиванием
    - «Quick sort» запускаем быструю сортировку
* После запуска одной из сортировок, производим визуализацию её алгоритма:
  + - Первым (синим) цветом подсвечиваются элементы, которые были отсортированы
    - Вторым (зелёным) цветом подсвечиваются текущие элементы, которые сортируются
    - Третьим (серым) цветом подсвечиваются элементы, которые ещё не сортировались
* После окончания работы сортировки все элементы окрашиваем в новый цвет (красный), проходимся по ним другим цветом (бирюзовым) для визуального подтверждения того, что сортировка окончена, и возвращаемся к серому цвету элементов.

# **План разработки**

План разработки и реализации программы представлен в таблице 1:

*Таблица №1: План разработки*

|  |  |  |  |
| --- | --- | --- | --- |
| Номер этапа работы | Дата начала этапа | Дата окончание этапа | Назначение этапа (описание работа, проводимых в данный промежуток времени) |
| 1 | 21.06.17 | 23.06.17 | * Определение темы проекта * Написание спецификации * Составление плана разработки * Распределение ответственности |
| 2 | 23.06.17 | 26.06.17 | * Создание прототипа программы, демонстрирующего пользовательский интерфейс |
| 3 | 26.06.17 | 28.06.17 | * Создание рабочей программы для визуализации сортировок:   + Программа должна работает, на оттестированных данных   + Должны быть реализованы все классы и методы, прописан интерфейс * Поиск проблем и ошибок в работе программы, в том числе проблемы отсутствия обработки некорректных данных |
| 4 | 28.06.17 | 30.06.17 | * Исправление всех ошибок * Создание финальной программы |

# **Разделение обязанностей**

Разделение обязанностей по членам команды представлено в таблице 2:

*Таблица №2: Разделение обязанностей*

|  |  |
| --- | --- |
| ФИО | Обязанности |
| Калганов Р.Г. | Создание интерфейса и визуализации |
| Круглик А.Д. | Разработка архитектуры проекта, координация действии, проверка деятельности других членов команды |
| Русу Я. | Реализация алгоритмов сортировки |