**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МОЭВМ**

отчет

**по учебной практике**

О проделанной работе на 23.06.2017

|  |  |  |
| --- | --- | --- |
| Студент гр. 5382 |  | Дворецкий В.Н. |
| Студент гр. 5382 |  | Лончина А.Е. |
| Студент гр. 5382 |  | Разбитнева А.М. |
| Руководитель |  | Фирсов М.А. |

Санкт-Петербург

2017

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | **1. требования к программе**  **Исходные требования к программе**  Формулировка задания: Написание мини-проекта по визуализации алгоритма пирамидальной сортировки на языке программирования Java. Программа включает: алгоритм сортировки, графический интерфейс, генерация массивов для тестирования программы.  1. Требования к вводу исходных данных   * В массиве все элементы должны быть одного типа. * Сортировка должна работать на элементах любого типа.   2. Требования к визуализации и т.д.   * Визуализация должна наглядно демонстрировать процесс сортировки. * Визуализация должна быть понятной.   **Спецификация программы**  Основой программы является алгоритм пирамидальной сортировки (HeapSort), который строит кучу для последующего выведения большего (или меньшего) элемента массива в корень (или начало массива).  Для проверки корректности алгоритма будет создан генератор массивов разных типов с последующей записи массивов в файл.  Ввод/вывод будет осуществляться посредством файла, содержащего нужный массив.  Так же для визуализации будет создан графический интерфейс, реализующий множество функций, связанных с различными возможностями визуализации. Далее с помощью скриншотов будет объяснена структура интерфейса.    рис. 1. Начальный экран программы  На начальном экране имеются 2 кнопки отвечающие за сортируемый массив.  Левая клавиша: генерация случайного массива. После ее нажатия на экране появляется сгенерированный массив.    рис. 2. Генерация массива  Правая клавиша: ввод массива с клавиатуры. После нажатия которой появляется окно содержащее строку, в которой пользователь вводит массив с клавиатуры. Интерфейс будет выглядеть как на рисунке 3.    рис. 3. Ввод массива с клавиатуры  После ввода или генерации массива предлагается выбрать один из трех вариантов визуализации. Обновленная страница интерфейса:    рис. 4. Варианты визуализации  Слева представлена визуализация на массиве с построением дерева, по центру – визуализация объектов разных размеров, справа – визуализация объектов разных цветов. После выбора одного из вариантов выплывает окно, предлагающее запустить визуализацию.    рис. 5. Запуск визуализации  Далее появляется окно с визуализацией, после окончания которой всплывает окно, позволяющее выбрать другой тип визуализации (кнопка слева сверху), вернуться к начальному экрану (кнопка справа сверху) или покинуть программу (кнопка снизу).    рис. 6. Окно выбора дальнейших действий  **Добавление к спецификации**  Предусмотрено создание визуализации с параметрами скорость (которая вводится через форму в миллисекундах и подтверждается нажатием кнопки) и в самом окне визуализации кнопка «пауза», по нажатию которой будет происходить остановка визуализации для более детального рассмотрения. Примерный шаблон окна визуализации предоставлен далее    **Спецификация визуализации**  Первая визуализация заключается в представлении массива в виде последовательно соединенных ячеек (таблицы размером Nx1, где N – размер массива). В соответствующие ячейки будут вставлены соответствующие элементы массива. Далее снизу будет строиться бинарное дерево, отображающее построение кучи. После чего стрелками будет показываться с какой по какую ячейку будут меняться нужные элементы и вместе с этим меняться структура кучи. После чего сами элементы поменяются индексами. Данные действия будут повторяться до тех пор, покуда массив не будет отсортирован. Далее на рисунках:    На первом этапе будет начинаться поэтапное построение бинарного дерева с выделением цвета какой элемент в какой лист дерева заполняется.    На этом этапе будет отображаться «прочесывание» самого дерева с выходом нужного элемента вверх. Так же цветом (но уже другим) будет показываться соответствие листы-элементы.    Отсортированные (а значит, незадействованные) элементы будут отображаться третьим цветом.    Само дерево будет выглядеть так:    Вторая визуализация будет разбита на два типа: по размеру и по цвету. Первый тип будет отображать в линию элементы (круги) разных радиусов. После чего стрелками будет показываться с какой по какой круг будет проходить swap элементов. После чего сами элементы поменяются местами. Данные действия будут повторяться до тех пор, покуда массив не будет отсортирован.    Разница между двумя визуализациями заключается в том, что в первом этапе отображается куча и на ней показывается сортировка, а во втором случае просто показывается какие элементы перемещаются после каждой итерации. |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |

**2. кто в бригаде за что отвечает**

При разработке проекта, применялись следующие роли:

* *Разработчик:* Дворецкий В.Н.

выполняет текущие задачи итерации, проектирует и реализовывает классы и алгоритмы.

* *Ответственный за итерацию:* Лончина А.Е.

проверяет качество и полноту исполнения задач в трекере и изменение их статуса, ответственна за написание генератора тестов и ввод их в программу.

* *Руководитель проекта:*  Разбитнева А.М.

определяет состояния исполнения задач проекта на каждой итерации, проектирует и разрабатывает интерфейс, оформляет документацию.

**3.план разработки**

26.06

Полностью прописан прототип графического интерфейса, но без возможности переходить по программе.

27.06

Реализация класса сортировки, соединения между переходами внутри программы, ручной ввод в программу, реализована визуализация сортировки на ячейках чисел, но без изображения построения бинарного дерева.

28.06

Дописано построение бинарного дерева, предоставлена другая версия визуализации (по размеру кругов).

30.06

Добавлена функция генерации в программе, окончательная версия программы, готовый отчет.