**Министерство образования и науки Российской Федерации**

**федеральное государственное автономное образовательное учреждение высшего образования**

**“САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,**

**МЕХАНИКИ И ОПТИКИ”**

**Факультет** СПО

**Направление подготовки (специальность)** 09.02.03 Программирование в компьютерных системах

**О Т Ч Е Т**

**об учебной практике   
по ПМ.01 «Разработка программных модулей программного обеспечения для компьютерных систем»**

Тема задания: Пирамидальная сортировка

Обучающийся Силакова А.Д. Группа Y2436

Руководитель практики от организации: Сорокин Дмитрий Юрьевич, преподаватель

Филиппова Ольга Юрьевна, преподаватель

Ответственный за практику от университета: Королев В.В., зам. декана ФСПО по НО

Практика пройдена с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_ (Кудрявцев А.С.)

\_\_\_\_\_\_\_\_ (Сорокин Д.Ю.)

\_\_\_\_\_\_\_\_ (Филиппова О.Ю.)

«\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_20\_\_\_г.

Подписи членов комиссии

Санкт-Петербург

2018/2019

**Министерство образования и науки Российской Федерации**

**федеральное государственное автономное образовательное учреждение высшего образования**

**“САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,**

**МЕХАНИКИ И ОПТИКИ”**

**Факультет среднего профессионального образования**

**И Н Д И В И Д У А Л Ь Н О Е З А Д А Н И Е**

**на учебную практику по ПМ.01 «Разработка программных модулей программного обеспечения для компьютерных систем»**

(наименование практики)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Студент** | Штыхина Мария Алексеевна | | **Группа №** | Y2436 |
| (Фамилия И. О.) | | | | |
|  | | | | |
| **Руководитель** | | Сорокин Дмитрий Юрьевич, преподаватель факультета СПО  Филиппова Ольга Юрьевна, преподаватель факультета СПО | | |
|  | | | | |

|  |  |  |
| --- | --- | --- |
| **Тема задания:** | Реализация сортировки выбором | |
| **Сроки прохождения практики:** | | 03.09.2018 – 11.01.2019 |
| **Место прохождения практики:** | | Факультет СПО |
| **1. Виды работ и требования к их выполнению:** | |  |

Учебная практика проводится аудиторно распределено (понедельно в течение семестра). В ходе прохождения практики выполняются следующие виды работ:

1. Вводный инструктаж по технике безопасности и общим целям, и задачам практики.
2. Анализ поставленной задачи: формирование по результатам анализа поставленной задачи исходных материалов для формирования технического задания на разработку. Изучение требований к составлению программной документации.
3. Выполнение индивидуального задания: формирование функциональные требований к разработке, выбор программных средств для реализации задачи, технологий тестирования, моделирование разработки, формирование разделов технического задания, программная реализация и тестирование разработки, формирование разделов технического описания.
4. Формирование отчета по учебной практике, включающего техническое задание и техническое описание разработанного программного продукта.

|  |  |
| --- | --- |
| **2. Виды отчетных материалов и требования к их оформлению:** |  |
|  | |

**3. ПЛАН-ГРАФИК**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| №  этапа | Наименование этапа | Срок завершения этапа | Виды работ | Форма отчетности |
| 1 | 2 | 3 | 4 | 5 |
| 1 | Вводный инструктаж | 03.09.2018 – 09.09.2018 | Ознакомление с инструкцией по технике безопасности. Ознакомление с целями и задачами учебной практики | Журнал по технике безопасности |
| 2 | Анализ поставленной задачи | 11.09.2018 – 30.09.2018 | Анализ индивидуального задания. Обследование предметной области согласно индивидуальной теме учебной практики. | Отчет по практике: техническое задание |
| 3 | Выполнение индивидуального задания | 02.10.2018 – 09.12.2018 | Определение требований к разработке. Выполнение поиска и мотивированного выбора инструментальных средств для решения задачи. Выбор технологий тестирования.  Формирование технического задания. Разработка кода программного продукта на основе готовых спецификаций на уровне модуля. Тестирование разработки. | Отчет по практике: техническое описание |
| 4 | Подготовка отчетных материалов | 11.12.2018 – 23.12.2018 | Формирование отчета о прохождении практики. | Отчет по практике |
| 5 | Защита результатов практики | 25.12.2018 – 11.01.2019 | Защита результатов практики в форме устного собеседования и представления результатов с руководителями практики и представителями работодателя. | Отчет по практике |

**Задание утверждено на заседании Совета факультета СПО**

**(протокол от «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_\_г. №\_\_\_\_\_\_\_).**

|  |  |  |
| --- | --- | --- |
| **Дата выдачи задания:** |  | |
|  | |  |
| **Руководитель** | | Д.Ю. Сорокин  О.Ю.Филиппова |
| **Консультант от работодателя** | | А.С.Кудрявцев, ведущий программист Филиала акционерного общества «Концерн радиостроения «Вега» в г.Санкт-Петербурге |
|  | | (подпись руководителя) |
| **Задание принял к** | |  |
| **исполнению** | |  |
|  | | (подпись студента) |

ОГЛАВЛЕНИЕ

[ОГЛАВЛЕНИЕ 4](#_Toc533089720)

[ТЕХНИЧЕСКОЕ ЗАДАНИЕ 5](#_Toc533089721)

[1. Назначение разработки 6](#_Toc533089722)

[2. Цели и задачи 7](#_Toc533089723)

[3. Требования к разработке 8](#_Toc533089724)

[3.1. Требования к программе или программному изделию 8](#_Toc533089725)

[3.2. Требования к надежности 8](#_Toc533089726)

[3.3. Требования к функциональным характеристикам входных и выходных данных 8](#_Toc533089727)

[3.4. Требования к информационной совместимости 9](#_Toc533089728)

[3.5. Стадии и этапы разработки 9](#_Toc533089729)

[3.6. Тестирование 10](#_Toc533089730)

[ТЕХНИЧЕСКОЕ ОПИСАНИЕ 11](#_Toc533089731)

[1. Общие сведения о разработки 12](#_Toc533089732)

[1.1. Наименование разработки 12](#_Toc533089733)

[1.2. Языки программирования, с помощью которых осуществлена разработка 12](#_Toc533089734)

[1.3. Назначение и функции, выполняемые разработкой 12](#_Toc533089735)

[1.4. Описание основных характеристик особенностей разработки 12](#_Toc533089736)

[2. Сведения о входных и выходных данных 14](#_Toc533089737)

[3. Описание логической структуры 15](#_Toc533089738)

[3.1. Модель работы системы 15](#_Toc533089739)

[3.2. Описание и методы решаемых задач 15](#_Toc533089740)

[3.3. Структура системы. Сведения о составных частях системы. Сведения о связях между составными частями системы. 15](#_Toc533089741)

[3.4. Описание выполняемых функций 16](#_Toc533089742)

[4. Описание выполнения программы 20](#_Toc533089743)

[4.1. Загрузка системы 20](#_Toc533089744)

[4.2. Способ вызова системы 20](#_Toc533089745)

[4.3. Режим работы системы 20](#_Toc533089746)

[4.4. Завершение работы системы 20](#_Toc533089747)

[ПРИЛОЖЕНИЕ 21](#_Toc533089748)

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

1. Назначение разработки

Графический редактор – это программа (или пакет программ), позволяющая создавать, просматривать, обрабатывать и редактировать цифровые изображения (рисунки, картинки, фотографии) на компьютере. Используется не только в учебных целях, но и часто применяется на практике. Разработка предназначена для изучения принципов работы с растровыми объектами. В дальнейшем разрабатываемая программа может использоваться в качестве многофункционального растрового графического редактора.

1. Цели и задачи

Разработка программы графического редактора предназначена для изучения работы с растровыми объектами.

В ходе выполнения работы будет необходимо не только отсортировать элементы массива, но и реализовать графический интерфейс для более удобного использования программы.

1. Требования к разработке
   1. Требования к программе или программному изделию

Разрабатываемая программа должна обеспечивать сортировку числовых элементов типа integer в одномерном массиве.

Программа будет реализована с использованием графического пользовательского интерфейса.

Программа должна поддерживать ввод пользователем массива данных для сортировки и визуализировать сортировку введенных чисел, отрицательных и положительных.

Соответственно, в качестве результата выполнения поставленной задачи, программа будет возвращать отсортированный одномерный массив элементов.

* 1. Требования к надежности

Разрабатываемая программа не должна прекращать свою работу в том случае, если на ввод будут даны некорректные данные. При их получении программа должна отобразить информацию о том, что данные неверны и, что необходимо ввести корректные значения. Программа не должна приводить к аварийному отключению компьютера.

* 1. Требования к функциональным характеристикам входных и выходных данных

Входные данные – это положительное или отрицательное число.

Выходные данные – это отсортированный массив целочисленных значений, визуализированный графически.

* 1. Требования к информационной совместимости

Программное обеспечение разрабатывается под платформу Windows. Программа должна работать под основными версиями этой платформы: Windows7, Windows8, Windows10.

Программа написана на языке с++.

* 1. Стадии и этапы разработки

Разработка программы от начала и до конца проходит следующие этапы:

1. Анализ требований.
2. Проектирование.
3. Кодирование (программирование).
4. Тестирование и отладка.
5. Эксплуатация и сопровождение.

На этапе анализа требований были выявлены требования и критерии, которым должна отвечать разрабатываемая программа. Программа должна осуществлять пирамидальную сортировку и иметь пользовательский интерфейс.

На этапе проектирования разрабатываемая программа реализуется на диаграммах. Во время этого этапа была создана Use case диаграмма, которая отражает действия, совершаемые пользователем во время работы с ПО. Она представлена на рисунке 1 в приложении.

В ходе кодирования происходит написание кода, который реализует выполнение поставленной задачи.

На этапе тестирования и отладки, готовый код проверяется на наличие программных и аппаратных ошибок.

Во время эксплуатации происходит использование программного продукта. Сопровождение, в свою очередь, заключается в поддержке и усовершенствовании работы ПО со стороны разработчика.

* 1. Тестирование

Для тестирования программы одним из наиболее подходящих методов является “White box”, относящийся к unit-тестированию.

Также нужно проверить работу интерфейса посредством интеграционного тестирования. То есть, в ходе интеграционного тестирования будет осуществлена проверка того, правильно ли проектируемая программа считывает данные, которые вводятся пользователем вручную с клавиатуры.

ТЕХНИЧЕСКОЕ ОПИСАНИЕ

1. Общие сведения о разработки
   1. Наименование разработки

Программа визуализации сортировки методом пирамиды.

* 1. Языки программирования, с помощью которых осуществлена разработка

Программное обеспечение разработано с помощью:

* реализация программного кода – C++;
* реализация графического интерфейса и программного кода с помощью среды для разработки Qt.
  1. Назначение и функции, выполняемые разработкой

Программа предназначена для визуализации процесса сортировки массива методом «пирамида».

Обеспечивает:

* возможность сортировки каждого элемента;
* интерфейс взаимодействия с пользователем;
* функцию ввода элементов массива;
* вывод визуализированного отсортированного массива на экран.

Разработанный программный продукт предоставляет пользователю следующие возможности:

1. Ввод элементов массива (отрицательные и положительные числа).
2. Вывод визуализированных элементов массива.
3. При нажатии на кнопку «сортировать» элементы массива меняют свое место согласно методу сортировки «пирамиды».
   1. Описание основных характеристик особенностей разработки

Для корректной работы программы требуется язык программирования C++.

Для визуализации и выполнения программного кода используется Qt.

1. Сведения о входных и выходных данных

Входные и выходные данные для перечисленных функций соответствуют указанным в техническом задании и приведены в таблице 1.

Таблица 1- Описание входных и выходных данных

|  |  |  |
| --- | --- | --- |
| Функция | Входные данные | Выходные данные |
| Введение значения элемента массива | Число | Заполненное поле |
| «Добавить элемент» | Идентификатор нажатия манипулятора «мышь» | Добавление элементов массива в строку через запятую |
| «Запустить алгоритм» | Идентификатор нажатия манипулятора «мышь» | Отсортированный массив элементов, преобразованных в прямоугольные объекты |

1. Описание логической структуры
   1. Модель работы системы

Приложение реализовано в соответствии с USE-case диаграммой на рисунке 1.

* 1. Описание и методы решаемых задач

Разработанная система решает ряд задач. Ниже представлено описание основных задач и методы их решения.

1. Реализация пользовательского интерфейса, визуализирующего сортировку массива.
2. Наличие окна для ввода элементов массива. Элементы могут вводиться с клавиатуры.
3. Визуализация элементов массива.
4. При нажатии кнопки «добавить» отображается новый элемент массива.
5. При нажатии кнопки «Запустить алгоритм» происходит процесс визуализации сортировки.
6. В конце прохождения алгоритма сортировки элементы массива отображаются на соответствующих сортировке местах.
   1. Структура системы. Сведения о составных частях системы. Сведения о связях между составными частями системы.

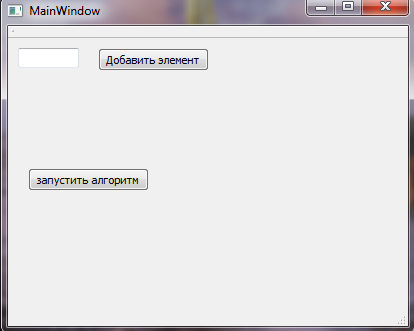
Состав файлов представлен в таблице 2.

Таблица 2 Структура системы

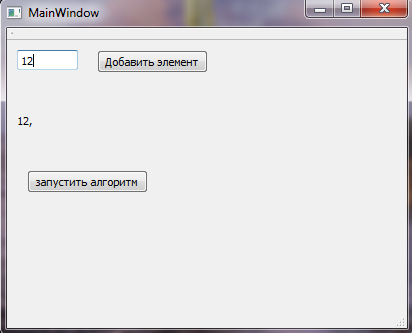
|  |  |
| --- | --- |
| Файл | Назначение |
| main.cpp | Реализация кода |
| main.pro | Файл, в котором декларируются основные параметры проекта, такие как подключаемые модули, шаблон приложения, список исходных кодов программы, а также заголовочных файлов |
| main.h | Описание объявления переменных, функций |

* 1. Описание выполняемых функций

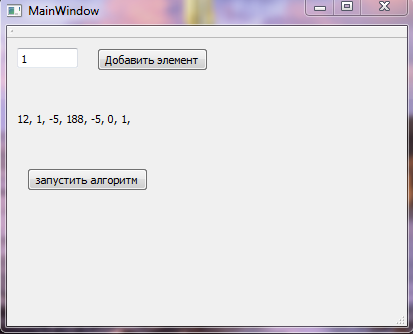
Функции системы реализованы в соответствии с техническим заданием. Изображения графических интерфейсов, соответствующие этим функциям, представлены на рисунках 1 – 6.



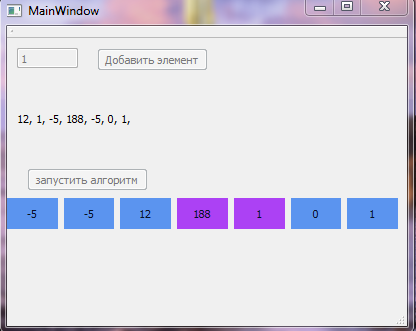
*Рисунок 1 – Внешний вид интерфейса*



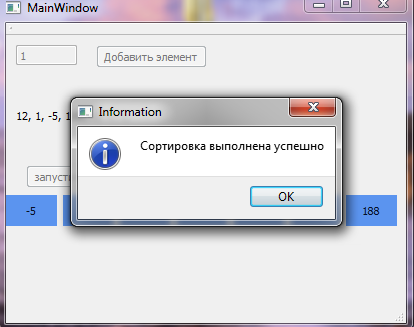
*Рисунок 2 – Ввод и добавление элемента массива*

**

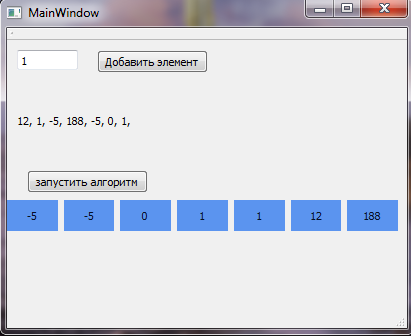
*Рисунок 3 – Введенный массив*



*Рисунок 4 – Сортировка массива*



*Рисунок 5 – Конец сортировки*



*Рисунок 6 – Отсортированный массив*

1. Описание выполнения программы
   1. Загрузка системы

Установка системы не требуется. Для работы программы необходимо запустить exe файл.

* 1. Способ вызова системы

Запуск системы осуществляется путем запуска исполняемого файла.

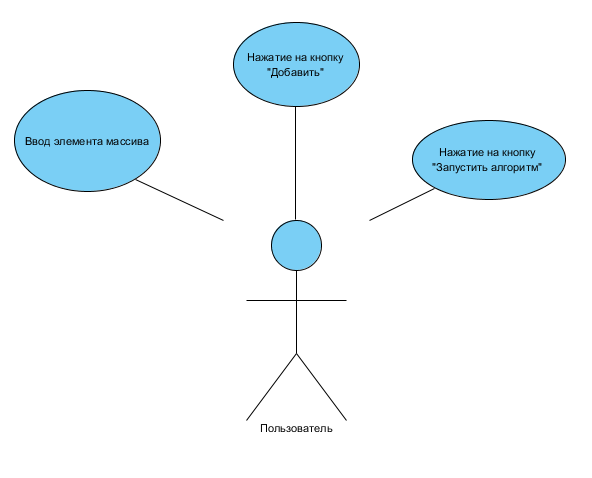
* 1. Режим работы системы

Взаимодействия пользователя с приложением осуществляется с помощью манипуляторами типов «мышь» и «клавиатура».

* 1. Завершение работы системы

Завершение работы производится закрытием окна программы.

ПРИЛОЖЕНИЕ



*Рисунок П1.1 – Диаграмма прецедентов*