ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 3](#_Toc130351465)

[РАЗДЕЛ 1 5](#_Toc130351466)

[1.1 Разработка технического задания для программного продукта 5](#_Toc130351467)

[1.2Разработка спецификации на программный продукт 6](#_Toc130351468)

[1.3Разработка функциональной диаграммы и диаграммы потоков данных 6](#_Toc130351469)

[1.4Разработка функциональной схемы программного продукта 8](#_Toc130351470)

[1.5 Разработка кода программных модулей программного продукта 11](#_Toc130351471)

[1.6 Разработка пользовательского интерфейса программного продукта в 13](#_Toc130351472)

[визуальной среде. 13](#_Toc130351473)

[1.7 Интеграция программных модулей в программный продукт. 14](#_Toc130351474)

[1.8 Тестирование программного продукта. 15](#_Toc130351476)

[1.9 Разработка справочной системы программного продукта. 18](#_Toc130351478)

[1.10 Руководство для пользователя. 19](#_Toc130351479)

[Загрузка программного комплекса на GitHab. 19](#_Toc130351480)

[ЗАКЛЮЧЕНИЕ 21](#_Toc130351481)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 22](#_Toc130351482)

## ВВЕДЕНИЕ

Программа профессионального модуля разработана на основе Федерального государственного образовательного стандарта – 3 по специальности «09.02.03 Программирование в компьютерных системах» в части освоения профессионального модуля ПМ.03 Участие в интеграции программных модулей.

Цель практики: закрепить навыки разработки программного обеспечения, использования методов для получения кода с заданной функциональностью и степенью качества, разработки документации на программный продукт; знаний моделей процесса разработки программного обеспечения, основных принципов процесса разработки программного обеспечения, основных подходов к интегрированию программных модулей, основных методов и средств эффективной разработки ПО.

Задачи практики:

1. Разработать техническое задание на программный продукт;
2. Разработать спецификацию на программный продукт;
3. Разработать функциональную диаграмму программного продукта, диаграмму потоков данных программных модулей продукта;
4. Разработать функциональную схему программного продукта, составить блок-схемы программных модулей программного продукта;
5. Разработать коды программных модулей программного продукта;
6. Разработать пользовательский интерфейс программного продукта в визуальной среде;
7. Выполнить интеграцию программных модулей в программный продукт;
8. Разработать процедуру тестирования программного продукта;
9. Составить инструкции по использованию функций программного продукта;
10. разработать руководства оператора (пользователя).

Разработка программного обеспечения-это процесс разработки, уточнения, проектирования, программирования, документирования, тестирования и исправления ошибок, связанных с созданием и поддержкой приложений, фреймворков или других программных компонентов.

Создание программного обеспечения осуществляется последовательно в соответствии со следующими основными этапами:

• постановка задачи. На данном этапе определяются назначение программного обеспечения и основные требования к нему. Определяются структуры входных и выходных данных. Производится предварительный выбор методов решения задач и языков программирования. Определяются стадии, этапы и сроки разработки программы и документации.

• анализ требований и разработка спецификаций. На этом этапе осуществляется описание общего алгоритма решения задачи. Здесь также необходимо подготовить тесты с указанием ожидаемых результатов для поиска программных и иных ошибок.

• проектирование. Основным на данном этапе является создание подробных спецификаций разрабатываемого приложения. Результатом этапа является подробная модель программного обеспечения со спецификациями его компонентов.

• реализация. Данный этап является последовательным процессом создания исходных кодов программы на выбранном языке программирования (кодирование), тестирования и отладки программного обеспечения.

• внедрение и эксплуатация. Данный этап - это процесс работы с программой на тестовых примерах с целью поиска и устранения ошибок в ней.

## РАЗДЕЛ 1

## 1.1 Разработка технического задания для программного продукта

Техническое задание (ТЗ) – основной документ проекта, которым Заявитель устанавливает основные цели и задачи проекта, номенклатуру и назначение продуктов проекта, технические и иные значимые характеристики модернизируемого производства и/или продукта проекта, порядок и последовательность необходимых стадий реализации проекта, создания продукта проекта и контроля его качественных параметров.

Тема: Необходимо разработать программный комплекс по демонстрации работы алгоритмов сортировки массивов данных.

Условие задачи: Разработать проект, который позволит сортировать массивы данных четырьмя способами: сортировка пузырьком, сортировка вставками, сортировка быстрой вставки, сортировка Шелла. Программа будет разработана на языке С++.

Для нормальной работы программы необходимы следующие технические и программные средства:

* компьютер на базе процессора Pentium 100 (или выше)
* жесткий диск объемом 500 Мб (и выше)
* объем оперативной памяти не менее 8 Мб
* операционная система Windows 95 и выше, 32 бита.

Пользовательский интерфейс должен быть реализован в визуальной среде на языке C++. Так же интерфейс должен содержать элементы для ввода массивов данных, выбора метода сортировки и вывода результатов сортировки.

Вся программа должна быть хорошо документирована, включая описание классов, функций и переменных. В документации должны быть описаны все особенности программы, ее возможности и ограничения. Документация должна быть доступна для пользователя в виде руководства пользователя.

## Разработка спецификации на программный продукт

Спецификация - это перечень материалов (работ, услуг) и их количество, которые необходимо переработать для изготовления определенного количества готовой продукции (работ, услуг). В ней указывают: изделие, его сборочные единицы, детали, материалы. Спецификация создается для упрощения процесса ввода данных при списании материалов в производство. Она применяется, когда могут быть заданы нормы расхода материалов на определенное количество единиц готовой продукции.

В окне приложения присутствуют: поле для выбора размера массива и два поля, одно для ввода – в него вставляем исходный массив, а второе- для вывода, где отображается отсортированный массив. Также в окне расположены 4 кнопки для выбора метода сортировки: сортировка Пузырьком, сортировка Шелла, Быстрая сортировка и сортировка Вставки. (см. рис.1)

Язык интерфейса: русский.

Время выполнения сортировки не должно превышать минуты для небольшого потока данных.

## Разработка функциональной диаграммы и диаграммы потоков данных

Функциональная диаграмма – диаграмма, отражающая взаимные связи функций, разрабатываемого программного обеспечения. Они создаются на ранних стадиях проектирования, для того чтобы помочь проектировщику выявить основные функции и по возможности обнаружить и устранить существующие ошибки.

Диаграмма потоков данных — это визуальное представление перемещения данных в пределах процесса или системы. Такие диаграммы помогают совершенствовать внутренние процессы и системы и определять правильный путь для ключевых разделов вашего бизнеса. Диаграмма потоков данных отображает последовательность данных, акторов и этапов в пределах процесса или системы.

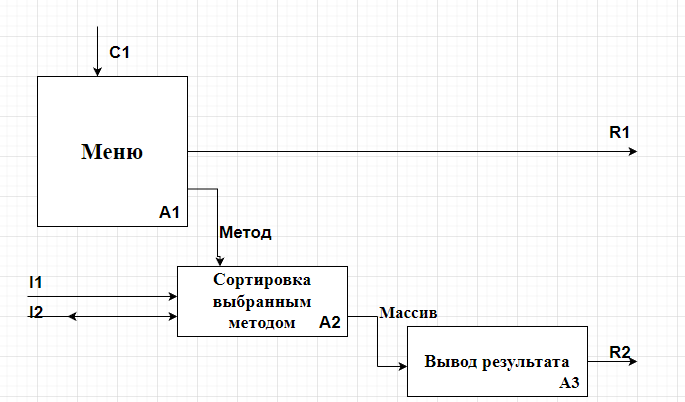


Рисунок 1 – Диаграмма потоков данных.

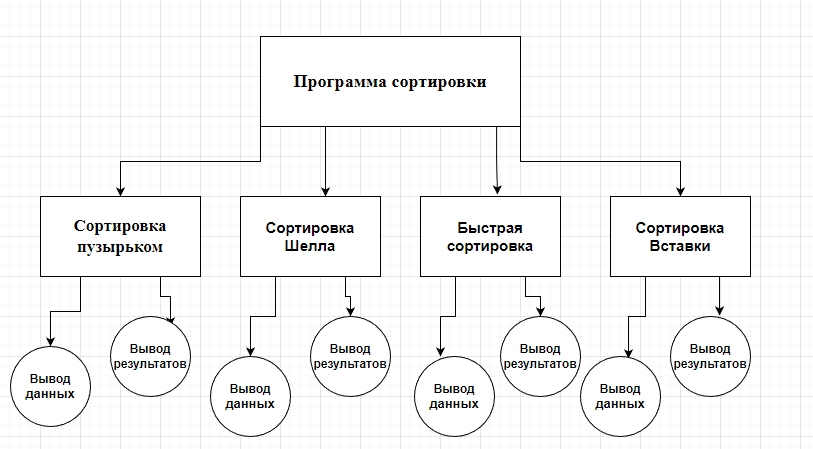


Рисунок 2 – функциональная диаграмма

## Разработка функциональной схемы программного продукта

Блок-схема – это последовательность блоков, предписывающих выполнение определенных операций, и связей между этими блоками. Внутри блоков указывается информация об операциях, подлежащих выполнению.

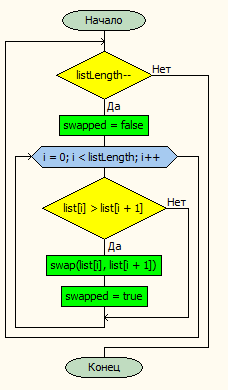


Рисунок 3 – блок-схема сортировки массива способом «Пузырек».

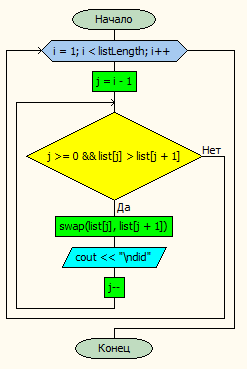


Рисунок 4 – блок-схема сортировки массива способом «Вставки».

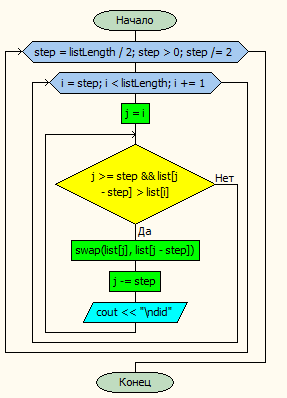


Рисунок 5 – блок-схема сортировки массива способом «Шелла».

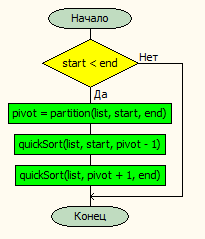


Рисунок 6 – блок-схема сортировки массива способом «Быстрой сортировки».

## 1.5 Разработка кода программных модулей программного продукта

Разработка кода для программного продукта, который сортирует массивы данных четырьмя способами (пузырьковая сортировка, сортировка вставками сортировка Шелла, и быстрая сортировка), согласно техническому заданию, требует языка программирования, такого как C++. Ниже предоставлены коды программных модулей программного продукта.

Сортировка пузырьком

void pyzirok(int list[], int listLength)

{

while(listLength--)

{

bool swapped = false;

for(int i = 0; i < listLength; i++)

{

if(list[i] > list[i + 1])

{

swap(list[i], list[i + 1]);

swapped = true;

}

}

}

При пузырьковой сортировке сравниваются соседние элементы и меняются местами, если следующий элемент меньше предыдущего. Требуется несколько проходов по данным. Во время первого прохода сраваются первые два элемента в массиве. Если они не в порядке, они меняются местами и затем сравнивается элементы в следующей паре. При том же условии они так же меняются местами. Таким образом сортировка происходит в каждом цикле пока не будет достигнут конец массива.

Сортировка методом Вставки

void vstavka(int list[], int listLength)

{

for(int i = 1; i < listLength; i++)

{

int j = i - 1;

while(j >= 0 && list[j] > list[j + 1])

{

swap(list[j], list[j + 1]);

cout<<"\ndid";

j--;

}

}

}

При сортировке вставками массив разбивается на две области: упорядоченную и и неупорядоченную. Изначально весь массив является неупорядоченной областью. При первом проходе первый элемент из неупорядоченной области изымается и помещается в правильном положении в упорядоченной области. На каждом проходе размер упорядоченной области возрастает на 1, а размер неупорядоченной области сокращается на 1.

Сортировка методом Шелла

void shella(int list[], int listLength)

{

for(int step = listLength/2; step > 0; step /= 2)

{

for (int i = step; i < listLength; i += 1)

{

int j = i;

while(j >= step && list[j - step] > list[i])

{

swap(list[j], list[j - step]);

j-=step;

cout<<"\ndid";

}

}

}

Алгоритм включает в себя сортировку вставками. Исходный массив размером N разбивается на подмассивы с шагомN/2. Подмассивы сортируются вставками. Затем вновь разбиваются, но уже с шагом равным N/4. Цикл повторяется. Производим целочисленное деление шага на два каждую итерацию. Когда шаг становится равен 1, массив просто сортируется вставками.

Быстрая сортировка

void quickSort(int list[], int start, int end)

{

if(start < end)

{

int pivot = partition(list, start, end);

quickSort(list, start, pivot - 1);

quickSort(list, pivot + 1, end);

}

}

В основе быстрой сортировки лежит стратегия «разделяй и властвуй». Задача разделяется на более мелкие подзадачи. Подзадачи решаются отдельно, а потом решения объединяют. Точно так же, массив разделяется на подмассивы, которые сортируются и затем сливаются в один.

## 1.6 Разработка пользовательского интерфейса программного продукта в

## визуальной среде.

Пользовательский интерфейс приведен на рис.7. Интерфейс был изобретен в Paint. На основании всех требований заказчика, был создан простой и понятный интерфейс.

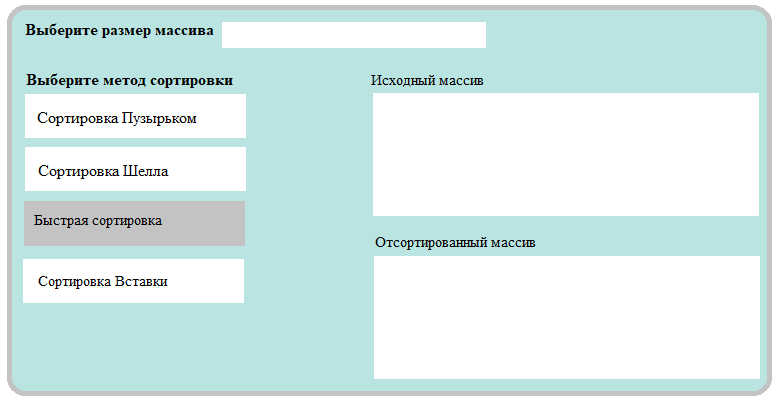


Рисунок 7 – Интерфейс.

1.7 Интеграция программных модулей в программный продукт.

Интеграция – процесс разработки и внедрения программного обеспечения, с помощью которого отдельные компоненты могут быть связаны в единую систему. Такое объединение позволяет поддерживать бизнес-процессы и оперативно обмениваться информацией.

Главная задача процесса – обеспечение безопасного и бесперебойного обмена информацией между программными продуктами, которые изначально не предназначены для совместной работы. Например, программное обеспечение для электронного документооборота между предприятием и его клиентами, организация цепей поставок, ERP-системы, облачные технологии, аналитические модули, системы самообслуживания и т. д.

С помощью интеграции программных модулей обеспечивается не только оперативность и автоматизация работы с данными. Решается ряд более широких задач:

* оперативное внедрение новых программных продуктов в работу предприятия;
* улучшения качества работы с клиентами;
* прозрачность процессов;
* сокращения количества ошибок при обработке данных и т. д.

Интеграция программных модулей – сложный процесс, требующий определенных знаний на каждом этапе его разработки, внедрения, запуска, дальнейшего использования и обслуживания. Поэтому важно уделить внимание не только задачам, которые стоят перед внедрением. Любая работа по внедрению программного обеспечения включает в себя обязательное тестирование и дальнейшее обучение пользователей.

1.8 Тестирование программного продукта.

Модульное тестирование и пользовательское тестирование — это два типа тестирования, которые используются для обеспечения качества программного продукта.

Модульное тестирование — это процесс тестирования отдельных модулей или компонентов программного продукта изолированно от остальной системы. Этот тип тестирования обычно выполняется разработчиками как часть процесса разработки. Целью модульного тестирования является выявление и исправление дефектов в коде как можно раньше, до того, как код будет интегрирован с другими компонентами системы.

Тестирование проводилось в два этапа: авторское тестирование, проведенное самим разработчиком, и тестирование третьим лицом. Результаты тестирования представлены в таблице 1. Оба тестирования производились на ПК DESKTOP-QLMO4RF.

Таблица 1 – результаты авторского тестирования

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Номер | Описание | Шаги | Входные данные | Ожидаемые результаты | Фактические результаты | Статус |
| 1 | Проверка получения верных результатов | Зайти в программу. Выбрать вид сортировки, ввести размер массива, ввести данные | 145635 | 139065 | 139065 | Успешно пройдено |
| 2 | Проверка на выполнение сортировки с большими данными | Зайти в программу. Выбрать вид сортировки, ввести размер массива, ввести данные | 936326768752 | 32788555554557 | 32788555554557 | Успешно пройден |
| 3 | Проверка на выполнения без указания величины массива | Зайти в программу, выбрать сортировку, ввести данные | 2634 | 2346 | Ошибка | Не пройден |
| 4 | Проверка на очистку поля вывода при новых данных | Два раза обработать массив | 24377247 | При выполнении сортировки во второй раз, данные появятся в чистой строке | Данные не стираются | Не пройден |
| 5 | Проверка работоспособности сортировки пузырьком | Зайти в программу, ввести размер массива, ввести данные, выбрать сортировку пузырьком | 72561 | 12567 | 12567 | Успешно пройдено |
| 6 | Проверка работоспособности сортировки Шелла | Зайти в программу, ввести размер массива, ввести данные, выбрать сортировку вставками | 78903 | 093078 | 093078 | Успешно пройдено |
| 8 | Проверка работоспособности быстрой сортировки | Зайти в программу, ввести размер массива, ввести данные, выбрать шейкерную сортировку | 78903 | 093078 | 093078 | Успешно пройдено |
| 9 | Проверка работоспособности сортировки вставки | Зайти в программу, ввести размер массива, ввести данные, выбрать сортировку расческой | 78903 | 093078 | 093078 | Успешно пройдено |
| 10 | Проверка при вводе буквы, вместо чисел | Зайти в программу, ввести размер массива, в котором будут буквы, ввести данные, выбрать сортировку | 0643ы | Окно с ошибкой | Просто ошибка, не работает | Не пройден |

## 1.9 Разработка справочной системы программного продукта.

Описание программы

Программа предназначена для сортировки массивов данных четырьмя способами: сортировка пузырьком, сортировка вставками, сортировка быстрой вставки, сортировка Шелла.

После запуска программы пользователю открывается окно формы, где есть поле для ввода данных, четыре кнопки с выбором метода сортировки и поле для вывода данных. Поле для ввода данных расположено вверху окна. Каждая кнопка подписана соответствующим ей методом сортировки, а именно:

* сортировка Пузырьком;
* сортировка Шелла;
* быстрая сортировка;
* сортировка вставки.

Пример использования программы для сортировки массива данных методом быстрой сортировки:

1. Введите размер массива.
2. Введите массив данных или сгенерируйте случайный массив заданного размера.
3. Выберите метод сортировки " Сортировка Шелла ".
4. Нажмите кнопку "Сортировать".
5. После завершения сортировки, время выполнения и отсортированный массив будет отображен в графическом интерфейсе.
6. Сохраните отсортированный массив в файл.

## 1.10 Руководство для пользователя.

Документ «Руководство пользователя» относится к эксплуатационной документации. Основная функция руководства пользователя заключается в обеспечении пользователя необходимой информацией для самостоятельной работы с программным обеспечением.

Таким образом, «Руководство пользователя» должно отвечать на следующие вопросы: что это за программа, что она может, что необходимо для обеспечения ее корректного функционирования и что делать в случае отказа системы. Программа предназначена для сортировки данных, которые ввел пользователь.

Данные должны вводится строго через пробел, не должно быть никаких знаков или букв, только числа.

В поле, для размера массива вводится количество чисел, которые будут отсортированы. В исходный массив вводятся числа, после выбирается вид сортировки. В поле «отсортированный массив» будут выведен отсортированный массив.

## Загрузка программного комплекса на GitHab.

GitHub — это сервис для совместной разработки и хостинга проектов. C помощью GitHub над кодом проекта может работать неограниченное количество программистов из любых точек мира.

В GitHub есть система контроля (управления) версий Git: сервис позволяет просматривать и контролировать любые изменения кода любым разработчиком и возвращаться к состоянию до изменений.

Ознакомиться со всеми материалами данного продукта можно по ссылке: https://github.com/ailinsuleimanova/sortirovka/upload/main.

## ЗАКЛЮЧЕНИЕ

Учебная практика по программному модулю ПМ.03 «Участие в интеграции программных модулей», проводилась согласно, положения о практике обучающихся, осваивающих основные профессиональные образовательные программы среднего профессионального образования, утвержденного Приказом Министерства образования Российской Федерации № 093 от 02 марта 2012 г. Целью данной практики является улучшение и закрепление навыков создания баз данных. Для этого было предложено к выполнению создание базы данных «ФССП».

В ходе выполнения цели учебной практики были исполнены следующие задачи, а именно:

1. Разработано техническое задание на программный продукт;
2. Разработана спецификация на программный продукт;
3. Разработаны функциональная диаграммы программного продукта, диаграмма потоков данных программных модулей продукта;
4. Разработана функциональная схема программного продукта, составлены блок-схемы программных модулей программного продукта;
5. Разработаны коды программных модулей программного продукта;
6. Разработан пользовательский интерфейс программного продукта в визуальной среде;
7. Выполнена интеграция программных модулей в программный продукт;
8. Разработана процедура тестирования программного продукта;
9. Составлена инструкция по использованию функций программного продукта;
10. Разработаны руководства оператора (пользователя).

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Разработка программного обеспечения – Краткое руководство // https://coderlessons.com URL:https://coderlessons.com/tutorials/akademicheskii/programmnaia-inzheneriia/razrabotka-programmnogo-obespecheniia-kratkoe-rukovodstvo (дата обращения: 16.03.2023).
2. Как составить техническое задание и получить то, что нужно // https://kontur.ru URL: https://kontur.ru/articles/5945 (дата обращения: 16.03.2023).
3. Спецификации и их роль в разработке программ // https://studfile.net URL: <https://studfile.net/preview/3004033/page:2/> (дата обращения: 17.03.2023).
4. Функциональные диаграммы // https://studref.com URL: https://studref.com/311808/informatika/funktsionalnye\_diagrammy (дата обращения: 17.03.2023).
5. Блок-схема и ее элементы // https://present5.com URL: https://present5.com/blok-sxema-i-ee-elementy-blok-sxema-eto-posledovatelnost/ (дата обращения: 20.03.2023).
6. Интеграция программных модулей // https://dynamicsun.ru URL: https://dynamicsun.ru/blog/integraciya-moduley.html (дата обращения: 21.03.2023).