МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное

образовательное учреждение высшего образования

«Крымский федеральный университет имени В.И. Вернадского»

Таврический колледж

(структурное подразделение)

**ОТЧЕТ ПО ПРАКТИКЕ**

**Учебная практика по профессиональному модулю**

**ПМ.03 Участие в интеграции программных модулей**

Специальность **09.02.03 Программирование в компьютерных системах**

Обучающийся 4 курса группы **4ПКС17**

форма обучения⠀⠀⠀⠀очная⠀⠀⠀⠀

(очная, заочная)

⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀ ⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀  
(фамилия, имя, отчество)

Место практики

Таврический колледж (структурное подразделение) ФГАОУ «КФУ им. В.И. Вернадского»

(наименование организации)

Срок практики с **16 марта 2023 г. по 22 марта 2023 г.**

Руководитель практики

от колледжа

преподаватель\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Руденко А.В. /

должность подпись (Ф.И.О.)

Зам директора

по учебно-производственной

практике \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Малюга Г.Г. /

подпись (Ф.И.О.)

Итоговая оценка по практике \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(отлично, хорошо, удовлетворительно)

МП

г. Симферополь, 2023 г.

**ОГЛАВЛЕНИЕ**

ВВЕДЕНИЕ 3

ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ 5

1.1 Техника безопасности 5

1.2 Язык программирования C# + Windows Form 6

1.3 Веб-сервис GitHub 7

ГЛАВА 2. ТЕХНИЧЕСКОЕ ЗАДАНИЕ 9

2.1 Разработка технического задания 9

2.2 Разработка спецификаций 10

2.3 Разработка диаграмм 11

ГЛАВА 3. РАЗРАБОТКА И ИНТЕГРАЦИЯ ПРОГРАММНЫХ МОДУЛЕЙ 14

3.1 Модуль интерфейса 14

3.2 Модуль CoctailSort 15

3.3 Модуль QuickSort 16

3.4 Модуль CombSort 18

3.5 Модуль InsertionSort 20

ГЛАВА 4. ТЕСТИРОВАНИЕ ПРОГРАММЫ 21

4.1 Процедура тестирования 21

4.2 Протокол тестирования 22

ГЛАВА 5. РАЗРАБОТКА СПРАВОЧНОЙ СИСТЕМЫ 24

5.1 Справочная информация 24

5.2 Руководство пользователя 24

ЗАКЛЮЧЕНИЕ 26

СПИСОК ЛИТЕРАТУРЫ 28

**ВВЕДЕНИЕ**

Учебная практика по программному модулю ПМ.03 "Участие в интеграции программных модулей", проводилась согласно положению о практике обучающихся, осваивающих основные профессиональные образовательные программы среднего профессионального образования, утвержденного Приказом Министерства образования Российской Федерации № 093 от 02 марта 2012 г.

Цель практики: формирование и развитие общих и профессиональных компетенций по модулю ПМ.03 Участие в интеграции программных модулей. Для достижения цели практики были поставлены следующие задачи:

* закрепить теоретические знания о технике безопасности при работе с электронно-вычислительными машинами;
* рассмотреть теоретическую информацию по выбранному языку программирования C# + Windows Form;
* изучить существующие алгоритмы сортировки чисел;
* Разработать техническое задание на программный продукт;
* Разработать спецификацию на программный продукт;
* Разработать функциональную диаграмму программного продукта, диаграмму потоков данных программных модулей продукта;
* Разработать функциональную схему программного продукта, составить блок-схемы программных модулей программного продукта;
* Разработать коды программных модулей программного продукта;
* Разработать пользовательский интерфейса программного продукта в визуальной среде;
* Выполнить интеграцию программных модулей в программный продукт;
* Разработать процедуру тестирования программного продукта;
* Выполнить тестирование программного продукта. Результат тестирования оформить протоколом тестирования;
* Разработать справочную систему программного продукта;
* Разработать руководства оператора (пользователя);
* Создать аккаунт в GitHub. Создать папку проекта. В папку загрузить разработанный программный комплекс, всю разработанную документацию к проекту;
* Составить отчет о выполнении;

Структура отчета по практике: содержание, введение, основная часть, состоящая из двух разделов, заключение, список использованных источников, приложения.

Первый раздел основной части, «Теоретические сведения», включает в себя информацию о технике безопасности, языке программирования C# + Windows Form и сервисе GitHub.

Второй раздел – «Реализация программного комплекса» содержит описание и саму документация программного комплекса, а также включает в себя краткое описание реализации программных модулей.

# ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

## 1.1 Техника безопасности

В кабинете вычислительной техники (КВТ) установлена дорогостоящая, сложная и требующая осторожного и аккуратного обращения аппаратура - компьютеры (ЭВМ), принтер, другие технические средства. Исходя из этого, следует:

* бережно обращайтесь с этой техникой;
* спокойно, не торопясь, не толкаясь, не задевая столы, входите в кабинет и занимайте отведенное вам место, ничего не трогая на столах.

На рабочем месте размещены составные части ЭВМ — системный блок, клавиатура, монитор (дисплей) и др. Во время работы лучевая трубка монитора (дисплея) работает под высоким напряжением. Неправильное обращение с аппаратурой, кабелями и мониторами может привести к тяжелым поражениям электрическим током, вызвать загорание аппаратуры.

Запрещается:

* трогать разъемы соединительных кабелей;
* прикасаться к питающим проводам и устройствам заземления;
* прикасаться к экрану и к тыльной стороне монитора;
* включать и отключать аппаратуру без указания преподавателя;
* класть диск, книги, тетради на монитор и клавиатуру;
* работать во влажной одежде и влажными руками.

Перед началом работы:

убедитесь в отсутствии видимых повреждений рабочего места; сядьте так, чтобы линия взора приходилась в центр экрана, чтобы, не наклоняясь пользоваться клавиатурой и воспринимать передаваемую на экран монитора информацию;

разместите на столе тетрадь, учебное пособие так, чтобы они не мешали работе на ЭВМ;

внимательно слушайте объяснения учителя и старайтесь понять цель и последовательность действий; в случае необходимости обращайтесь к преподавателю;

хорошо разберитесь в особенностях применяемых в работе аппаратов;

начинайте работу только по указанию преподавателя «Приступить к работе».

Во время работы:

Строго выполняйте все указанные выше правила, а также текущие указания учителя;

Следите за исправностью аппаратуры и немедленно прекращайте работу при появлении необычного звука или самопроизвольного отключения аппаратуры. Немедленно докладывайте об этом преподавателю;

Плавно нажимайте на клавиши, не допуская резких ударов;

Не пользуйтесь клавиатурой, если не подключено напряжение;

Работайте на клавиатуре чистыми руками;

Никогда не пытайтесь самостоятельно устранять неисправности в работе аппаратуры;

Не вставайте со своих мест, когда в кабинет входят посетители.

По окончании работы выключите компьютер [1].

## 1.2 Язык программирования C# + Windows Form

Данный язык программирования был выбран по причине удобной реализации графического интерфейса программы с помощью Windows Form.

C# (произносится как "си шарп") — современный объектно-ориентированный и типобезопасный язык программирования. C# позволяет разработчикам создавать разные типы безопасных и надежных приложений, выполняющихся в .NET. C# относится к широко известному семейству языков C, и покажется хорошо знакомым любому, кто работал с C, C++, Java или JavaScript.

C# — объектно-ориентированный, ориентированный на компоненты язык программирования. C# предоставляет языковые конструкции для непосредственной поддержки такой концепции работы. Благодаря этому C# подходит для создания и применения программных компонентов. С момента создания язык C# обогатился функциями для поддержки новых рабочих нагрузок и современными рекомендациями по разработке ПО. В основном C# — объектно-ориентированный язык [2].

Windows Forms — это платформа пользовательского интерфейса для создания классических приложений Windows. Она обеспечивает один из самых эффективных способов создания классических приложений с помощью визуального конструктора в Visual Studio. Такие функции, как размещение визуальных элементов управления путем перетаскивания, упрощают создание классических приложений.

В Windows Forms можно разрабатывать графически сложные приложения, которые просто развертывать, обновлять, и с которыми удобно работать как в автономном режиме, так и в сети. Приложения Windows Forms могут получать доступ к локальному оборудованию и файловой системе компьютера, на котором работает приложение [3].

## 1.3 Веб-сервис GitHub

GitHub — веб-сервис на основе Git, который помогает совместно разрабатывать IT-проекты. Создатели сайта называют GitHub «социальной сетью для разработчиков».

Кроме размещения кода, участники могут общаться, комментировать правки друг друга, а также следить за новостями знакомых.

С помощью широких возможностей Git программисты могут объединять свои репозитории — GitHub предлагает удобный интерфейс для этого и может отображать вклад каждого участника в виде дерева.

Для проектов есть личные страницы, небольшие Вики и система отслеживания ошибок.

Прямо на сайте можно просмотреть файлы проектов с подсветкой синтаксиса для большинства языков программирования.

Также:

* можно создавать приватные репозитории, которые будут видны только вам и выбранным вами людям. Раньше такая возможность была платной;
* есть возможность прямого добавления новых файлов в свой репозиторий через веб-интерфейс сервиса;
* код проектов можно не только скопировать через Git, но и скачать в виде обычных архивов с сайта;
* на сайте есть pastebin-сервис gist.github.com для быстрой публикации фрагментов кода;
* файлы из репозитория могут автоматически публиковаться в виде статического сайта с помощью GitHub Pages.

# ГЛАВА 2. ТЕХНИЧЕСКОЕ ЗАДАНИЕ

## 2.1 Разработка технического задания

Наименование программы: "УП\_Соколов\_Артём\_4ПКС17".

Программа предназначена для сортировки массивов различных размеров 4-мя разными методами сортировки.

Программа предоставляет свой пользовательский интерфейс

На стадии разработки технического задания должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии рабочего проектирования должны быть выполнены перечисленные ниже этапы работ:

Этапы разработки:  
1. Разработка программы

2. Разработка программной документации.

3. Испытание программы.

На стадии внедрения должен быть выполнен этап разработки подготовка и передача программы.

На этапе разработки технического задания должны быть выполнены перечисленные ниже работы:   
1. постановка задачи;   
2. определение и уточнение требований к техническим средствам;   
3. определение требований к программе;  
4. определение стадий, этапов и сроков разработки программы и документации на неё;   
5. согласование и утверждение технического задания.

На этапе разработки программы должна быть выполнена работа по программированию (кодированию) и отладке программы.

На этапе разработки программной документации должна быть выполнена разработка программных документов в соответствии с требованиями к составу документации.

На этапе испытаний программы должны быть выполнены перечисленные ниже виды работ:   
1. разработка, согласование и утверждение и методики испытаний;   
2. проведение приемо-сдаточных испытаний;   
3. корректировка программы и программной документации по результатам испытаний.

На этапе подготовки и передачи программы должна быть выполнена работа по подготовке и передаче программы и программной документации в эксплуатацию на объектах Заказчика.

2.2 Разработка спецификаций

Программа “ УП\_Соколов\_Артём\_4ПКС17” реализована на языке программирования C# + Windows Form, предназначена для выполнения задач, связанных с сортировкой различного размера массивов разными методами, среди них: метод пузырьком, метод выборки, метод перестановки, и метод вставки.

В программе представлены следующие модули:

* модуль интерфейса;
* модуль QuickSort;
* модуль CombSort;
* модуль InsertionSort;
* модуль CoctailSort. ;

Программа предназначена для людей, у которых стоит потребность в сортировке массивов различных размеров.

2.3 Разработка диаграмм



Рисунок 1 – Функциональная диаграмма

Данная диаграмма отражает взаимосвязи функций разрабатываемого программного обеспечения.

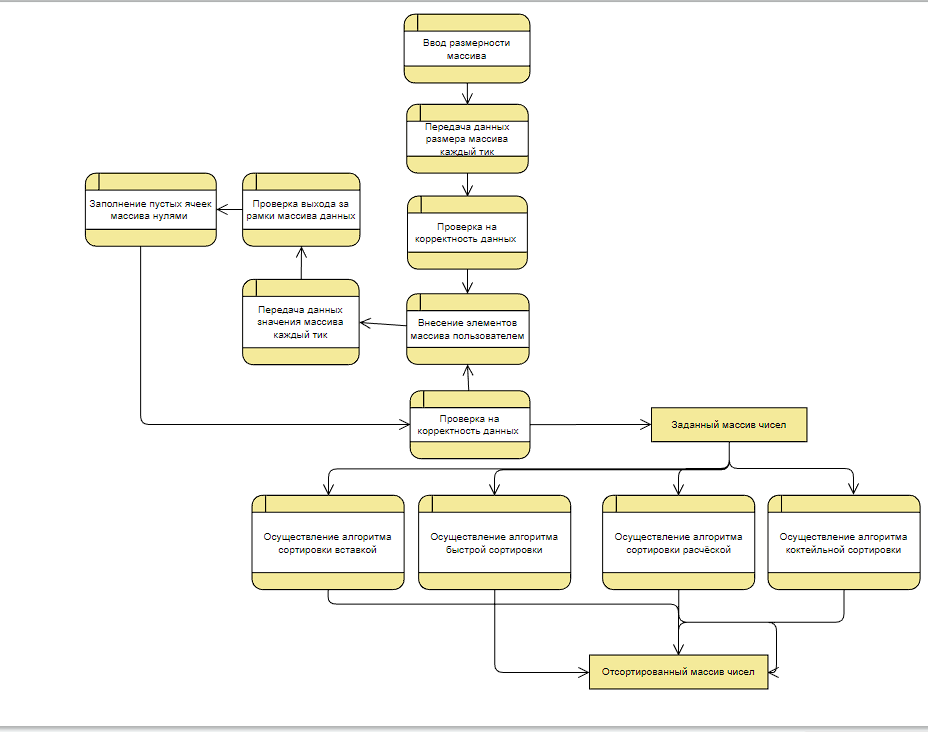


Рисунок 2 – Диаграмма потока данных

На данном рисунке представлена диаграмма потока данных в которой показывается какой именно путь данные проходят от начала программы до конечного результата.

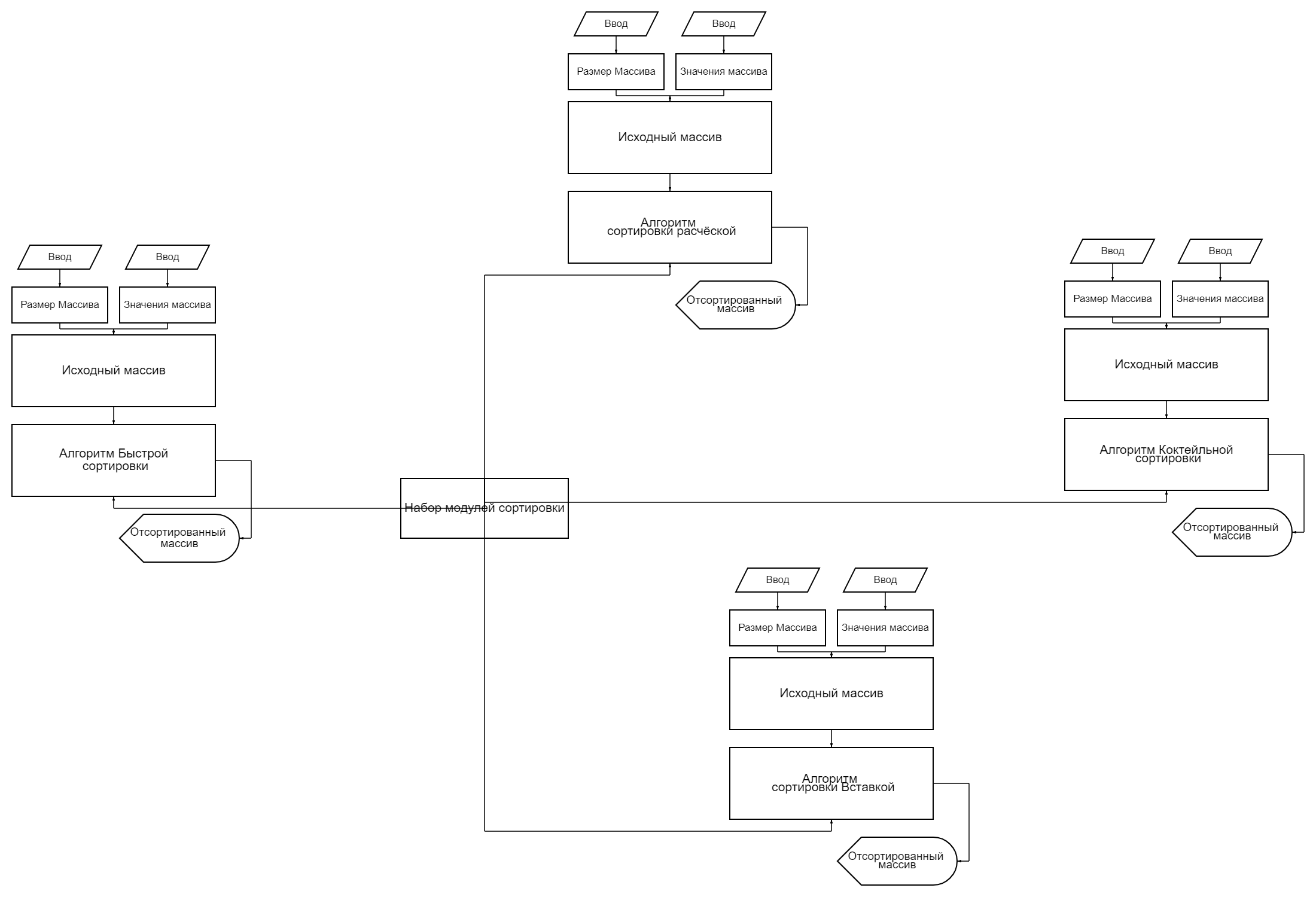


Рисунок 3 – Функциональная схема продукта

На данном рисунке представлена функциональная схема продукта где наглядно описана функциональная составляющая программного продукта

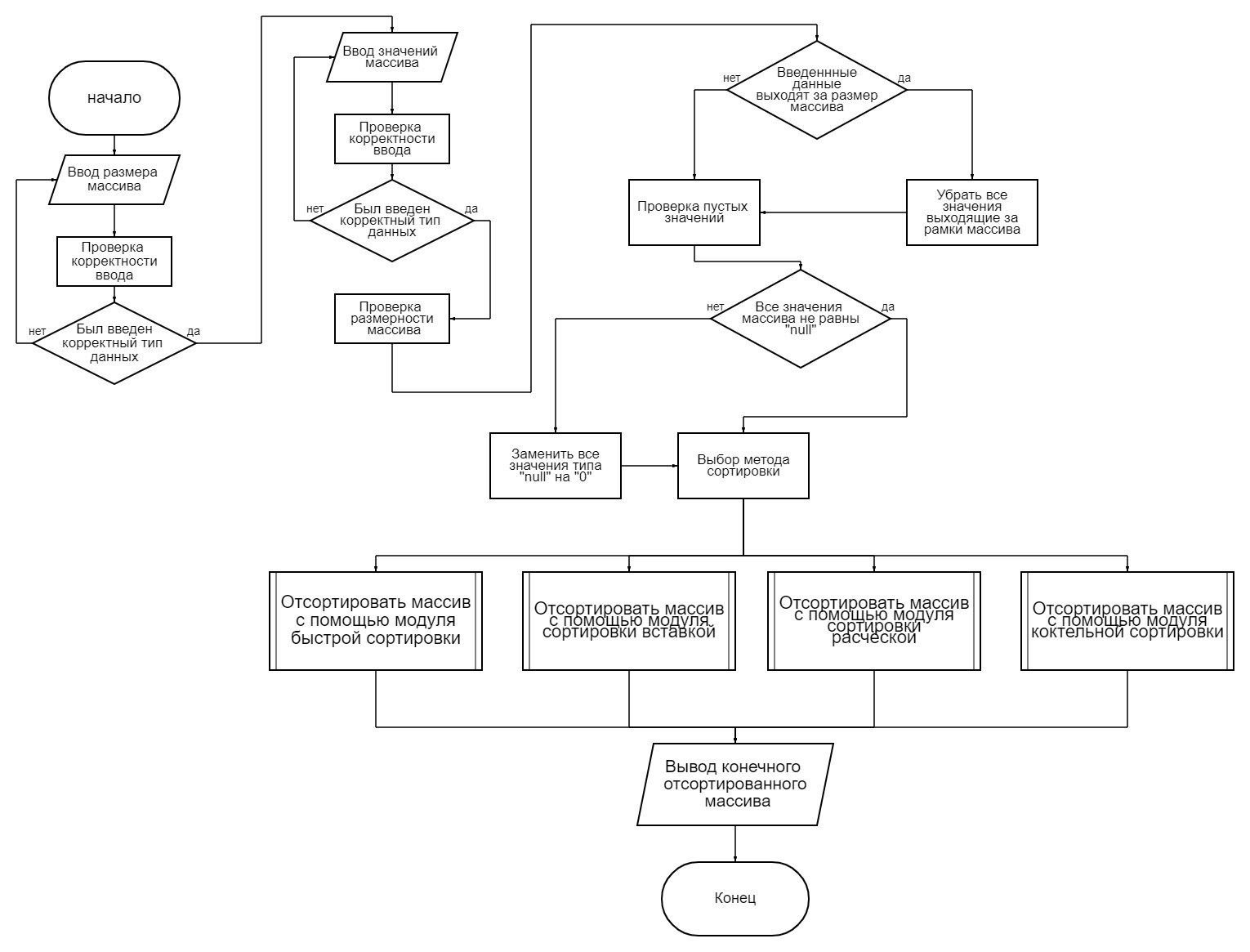


Рисунок 4 – Блок схема программы

На данном рисунке представлена блок схема реализованной программы где наглядно описана логика программного продукта со всеми необходимыми этапами.

**ГЛАВА 3. РАЗРАБОТКА И ИНТЕГРАЦИЯ ПРОГРАММНЫХ МОДУЛЕЙ**

3.1 Модуль интерфейса

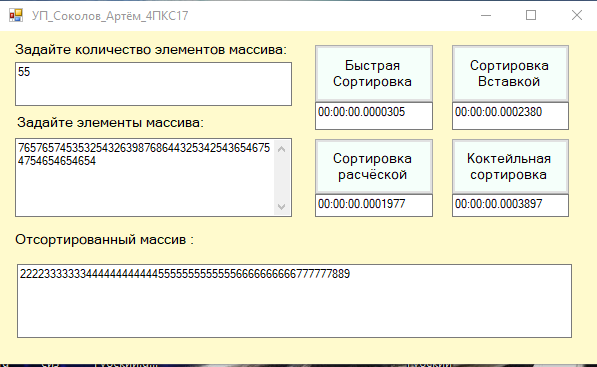


Рисунок 5 – Модуль интерфейса

На данном рисунке представлен общий интерфейс программы, в нем реализованы такие кнопки как:” Быстрая Сортировка”,” Сортировка Вставкой”,” Сортировка расчёской», «Коктейльная сортировка”, а также окна: “Задайте количество элементов Массива:”, “Задайте элементы массива:”, “Отсортированный массив:”.

3.2 Модуль CoctailSort

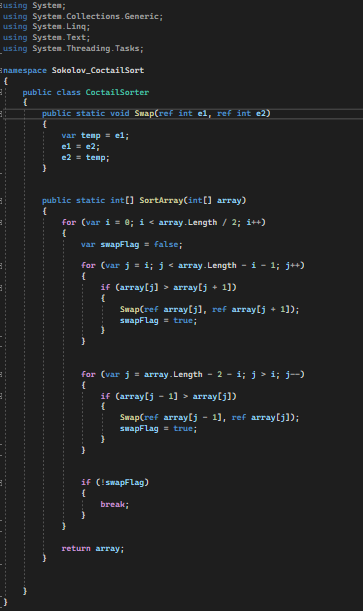


Рисунок 6 – Модуль CoctailSort

На данном рисунке представлена реализация метода коктейльной сортировки, а именно, весь необходимый для реализации внутри программы код.

Шейкерная (коктейльная) сортировка (Cocktail Sort). Модифицированный и немного улучшенный алгоритм пузырьковой сортировки, при котором обмен выполняется в двух направлениях – наибольшие элементы перемещаются в правую сторону, а во время обратного движения наименьшие движутся в левую сторону.

Алгоритм сортировки：

1. Сначала отсортируйте массив слева направо в пузырьковом порядке (в порядке возрастания), затем самый большой элемент перейдет в крайнее правое положение.

2. Затем отсортируйте массив справа налево в пузырьковом порядке (в порядке убывания), затем наименьший элемент перейдет в крайнее левое положение.

3. Цикл 1, 2 шага, последовательное изменение направления пузырьков и продолжение сужения диапазона несортированных элементов до конца последнего элемента [4]

3.3 Модуль QuickSort

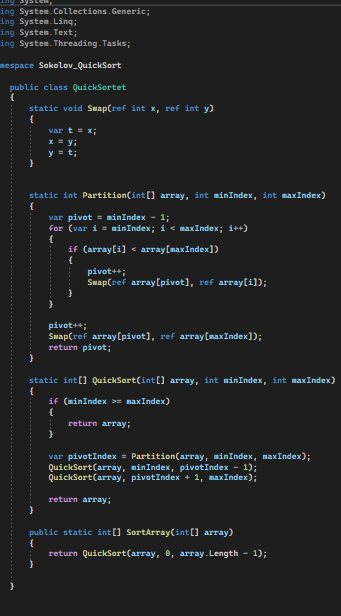


Рисунок 7 – Модуль QuickSort

На данном рисунке представлена реализация метода быстрой сортировки, а именно, весь необходимый для реализации внутри программы код.

Алгоритм быстрой сортировки является рекурсивным, поэтому для простоты процедура на вход будет принимать границы участка массива от l включительно и до r не включительно. Понятно, что для того, чтобы отсортировать весь массив, в качестве параметра l надо передать 0, а в качестве r — n, где по традиции n обозначает длину массива.

В основе алгоритма быстрой сортировке лежит процедура partition. Partition выбирает некоторый элемент массива и переставляет элементы участка массива таким образом, чтобы массив разбился на 2 части: левая часть содержит элементы, которые меньше этого элемента, а правая часть содержит элементы, которые больше или равны этого элемента. Такой разделяющий элемент называется пивотом.

Пивот в нашем случае выбирается случайным образом. Такой алгоритм называется рандомизированным. На самом деле пивот можно выбирать самым разным образом: либо брать случайный элемент, либо брать первый / последний элемент учаcтка, либо выбирать его каким-то «умным» образом. Выбор пивота является очень важным для итоговой сложности алгоритма сортировки, но об этом несколько позже. Сложность же процедуры partition — O(n), где n = r — l — длина участка.

Крайний случай — массив из одного элемента обладает свойством упорядоченности. Если массив длинный, то применяем partition и вызываем процедуру рекурсивно для двух половин массива.

Если прогнать написанную сортировку на примере массива 1 2 2, то можно заметить, что она никогда не закончится. Почему так получилось?

При написании partition мы сделали допущение — все элементы массива должны быть уникальны. В противном случае возвращаемое значение m будет равно l и рекурсия никогда не закончится, потому как sort(l, m) будет вызывать sort(l, l) и sort(l, m). Для решения данной проблемы надо массив разделять не на 2 части (< pivot и >= pivot), а на 3 части (< pivot, = pivot, > pivot) и вызывать рекурсивно сортировку для 1-ой и 3-ей частей. [5]

3.4 Модуль CombSort

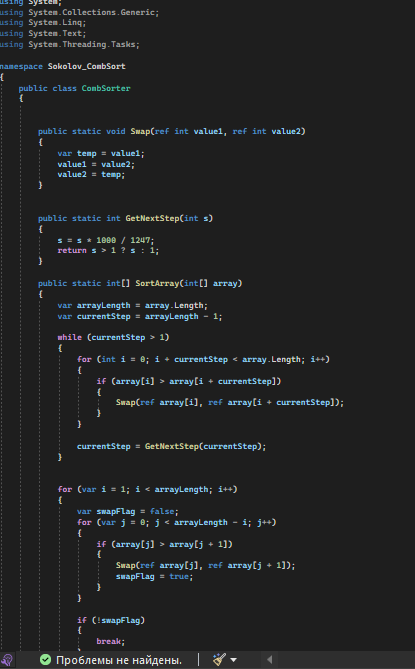


Рисунок 8 – Модуль CombSort

Раз у нас большие элементы могут тормозить весь процесс, то можно их перекидывать не на соседнее место, а подальше. Так мы уменьшим количество перестановок, а с ними сэкономим и процессорное время, нужное на их обработку.

Оптимизация кода:

Но отправлять каждый большой элемент сразу в конец массива будет недальновидно — мы же не знаем, насколько этот элемент большой по сравнению с остальными элементами. Поэтому в сортировке расчёской сравниваются элементы, которые отстоят друг от друга на каком-то расстоянии. Оно не слишком большое, чтобы сильно не откидывать элементы и возвращать потом большинство назад, но и не слишком маленькое, чтобы можно было отправлять не на соседнее место, а чуть дальше.

Опытным путём программистов установили оптимальное расстояние между элементами — это длина массива, поделённая на 1,247 (понятное дело, расстояние нужно округлить до целого числа). С этим числом алгоритм работает быстрее всего.

Как работает алгоритм сортировки расчёской

На первом шаге мы находим длину массива (например, 10 элементов) и делим её на 1,247. Допустим, после округления у нас получилось число 8. Теперь мы проходим первый цикл пузырьковой сортировки, только сравнивая не 1 и 2, 2 и 3, а сразу 1 и 8, 2 и 9, 3 и 10. Это отправит самые большие числа, если они есть в начале, в самый конец. Всего на первом шаге будет три сравнения.

На втором шаге мы берём число 8 из предыдущего этапа и снова делим его на 1,247, получая число 6. Теперь мы снова проходим весь массив и сравниваем так:

* 1 и 6
* 2 и 7
* 3 и 8
* 4 и 9
* 5 и 10

Уже получилось 5 перестановок и снова крупные числа улетели поближе к концу массива.

Так мы уменьшаем размер шага до тех пор, пока он не станет меньше единицы — к этому моменту массив будет полностью отсортирован.

Сортировка расчёской называется так из-за того, что мы как бы расчёсываем массив сначала широким гребнем (большой шаг), потом гребнем поменьше (шаг поменьше). В конце шаг равен единице, как в пузырьковой сортировке. [6]

3.5 Модуль InsertionSort

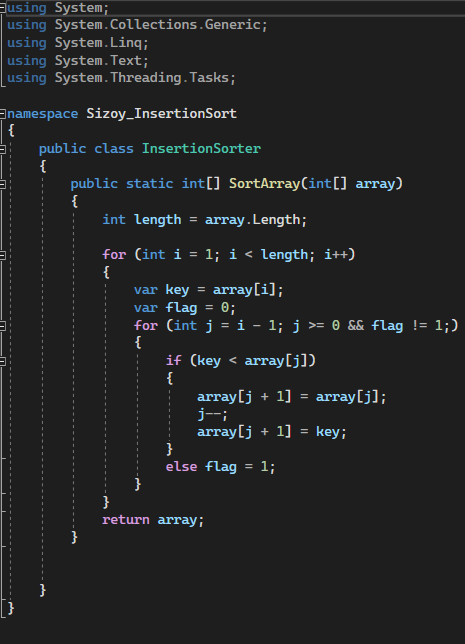


Рисунок 9 – Модуль InsertionSort

На данном рисунке представлена реализация метода вставки, весь необходимый для реализации внутри программы код.

Сортировка вставками - алгоритм, при котором каждый последующий элемент массива сравнивается с предыдущими элементами (отсортированными) и вставляется в нужную позицию.

Общая идея алгоритма: Сравниваем второй элемент с первым элементом массива и при необходимости меняем их местами. Условно эти элементы (первый и второй) будут являться отсортированным массивом, остальные элементы - неотсортированным. [7]

# ГЛАВА 4. ТЕСТИРОВАНИЕ ПРОГРАММЫ

## 4.1 Процедура тестирования

Программа "УП\_Соколов\_Артём\_4ПКС17" тестируется согласно следующим положениям:

Системные требования для компьютера, на котором происходит тестирование:

* процессор Pentium-2.0Hz, не менее;
* оперативную память объемом, 1Гигабайт, не менее;
* HDD, 40 Гигабайт, не менее;
* операционную систему Windows 2003 и выше.

Описание процедуры тестирования:

* Тестирование программы происходит без участия сторонних программных средств.
* Пользователю тестировщику предоставляется набор входных и предполагаемых выходных данных для программы (Табл. 1).
* Пользователь тестировщик проверяет соответствие входных и выходных данных во время реальных запусков с табличными значениями.
* По окончании процедуры тестирования составляется протокол тестирования.

Содержание протокола тестирования:

* Таблица реальных входных и выходных данных с комментариями тестировщика.
* Анализ результатов.

Входные данные представляют собой наборы цифр, введённые сплошным массивом.

Таблица 1 Набор входных и предполагаемых выходных данных

| **№** | **Метод сортировки** | **Входные данные: количество чисел в массиве** | **Входные данные: массив чисел** | **Ожидаемые результаты (отсортированный массив чисел)** | **Ожидаемая ошибка** | **Комментарий к сценарию проверки** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | CoctailSort | 5 | 32457 | 23457 | - | Проверка на нормальном вводе массива из 5 чисел. |
| 2 | QuickSort | 5 | 32457 | 23457 | - | Проверка на нормальном вводе массива из 5 чисел. |
| 3 | CombSort | 5 | 32457 | 23457 | - | Проверка на нормальном вводе массива из 5 чисел. |
| 4 | InsertionSort | 5 | 32457 | 23457 | - | Проверка на нормальном вводе массива из 5 чисел. |
| 5 | CoctailSort | 10 | 0.585 13 -0.53 14 55 0 0 -14 1.002 1.0002 | - | Неверный тип данных! | Проверка на вводе массива из 10 чисел с повторяющимися и дробными числами. |
| 6 | QuickSort | 0 | - | - | Введите данные! | Проверка на пустом вводе. |
| 7 | CombSort | 0 | Текст | - | Неверный тип данных! | Проверка на ввод текста. |
| 8 |  | 5 | 32457 | - | Выберите тип сортировки! | Проверка на возможность не выбирать тип сортировки. |

## 4.2 Протокол тестирования

По результатам запусков, описанных в процедуре тестирования, была составлена таблица результатов тестирования (Табл. 2).

Исходя из полученных данных, программа работает стабильно, критических ошибок не обнаружено.

Вердикт: программа полностью готова для потребления конечным пользователем.

Таблица 2 Результаты тестирования.

| **№** | **Метод сортировки** | **Входные данные: количество чисел в массиве** | **Входные данные: массив чисел** | **Результаты (отсортированный массив чисел)** | **Ошибка** | **Результат тестирования** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | CoctailSort | 5 | 32457 | 23457 | - | Полученные результаты соответствуют ожидаемым. |
| 2 | QuickSort | 5 | 32457 | 23457 | - | Полученные результаты соответствуют ожидаемым. |
| 3 | CombSort | 5 | 32457 | 23457 | - | Полученные результаты соответствуют ожидаемым. |
| 4 | InsertionSort | 5 | 32457 | 23457 | - | Полученные результаты соответствуют ожидаемым. |
| 5 | CoctailSort | 10 | 0.585 13 -0.53 14 55 0 0 -14 1.002 1.0002 | - | Неверный тип данных! | Полученные результаты соответствуют ожидаемым. |
| 6 | QuickSort | 0 | - | - | Введите данные! | Полученные результаты соответствуют ожидаемым. |
| 7 | CombSort | 0 | Текст | - | Неверный тип данных! | Полученные результаты соответствуют ожидаемым. |
| 8 |  | 5 | 32457 | - | Выберите тип сортировки! | Полученные результаты соответствуют ожидаемым. |

# ГЛАВА 5. РАЗРАБОТКА СПРАВОЧНОЙ СИСТЕМЫ

## 5.1 Справочная информация

Справочная информация – исчерпывающая информация о программе.

Пример такой информации из учебной практики:

Эта программа представляет собой приложение Windows Forms, которое позволяет пользователю сортировать список целых чисел, используя один из четырех различных алгоритмов сортировки. Интерфейс программы состоит из текстового поля для ввода списка целых чисел, группы кнопок для выбора алгоритма сортировки и текстового поля для отображения отсортированного списка. Так же присутствует отображение времени работы алгоритма.

Программа поддерживает четыре алгоритма сортировки:

* Шейкерная сортировка
* Сортировка расческой
* Сортировка вставками
* Быстрая сортировка

Чтобы воспользоваться программой, введите цельный массив цифр, которые вы хотите отсортировать, в первое текстовое поле. Выберите один из четырех алгоритмов сортировки с помощью кнопок. Отсортированный список будет отображен во втором текстовом поле.

Обратите внимание, что программа предполагает, что список ввода содержит только цифры, без разделения пробелами и т.д, и любые другие символы или форматирование приведут к сбою программы.

## 5.2 Руководство пользователя

Руководство пользователя — документ, назначение которого — предоставить людям помощь в использовании некоторой системы.

Пример такого руководства из учебной практики:

Введение:

Область применения: введённые данные и рамки приложения.

Описание возможностей: сортировка массива 4-мя методами, а также возможность задания размерности массива и его значений.

Уровень подготовки пользователя: умение работать с операционными системами типа Windows, умение печатать на клавиатуре и пользоваться компьютерной мышью.

Перечень документации для пользователя:

* Справочная информация;
* Руководство пользователя.

Назначение и условия применения:

-Данный программный продукт предназначен для автоматизации процесса сортировки массива.

-Условиями применения являются задание конечного размера массива и ввод целых действительных чисел.

Подготовка к работе:

- Загрузочный пакет программы

- Порядок загрузки программы: распаковка загрузочного пакета, взаимодействие с корневым exe файлом по средству его открытия, загрузка приложения.

- Порядок загрузки данных:

* Ввод с клавиатуры значения размерности массива;
* Ручной ввод значений массива;
* Клик по одной из 4-х кнопок для выбора метода сортировки.

Аварийные ситуации:

В случае какого-либо отклонения от правильной работы следует закрыть программу через диспетчера задач в главном его окне по средству по средству вызова диспетчера задач с помощью сочетания клавиш ctrl + alt + del и последующего открытия программы.

# ЗАКЛЮЧЕНИЕ

Завершив разработку программного комплекса, выполняющего функцию сортировки массива чисел разными методами, можно сказать, что в результате работы цель учебной практики была достигнута. А именно, сформированы и развиты общие и профессиональные компетенций по модулю ПМ.03 Участие в интеграции программных модулей.

При достижении цели данного проекта были выполнены следующие задачи:

* закреплены теоретические знания о технике безопасности при работе с электронно-вычислительными машинами;
* рассмотрена теоретическая информация по выбранному языку программирования C# + Windows Form;
* изучены существующие алгоритмы сортировки чисел;
* Разработано техническое задание на программный продукт;
* Разработана спецификация на программный продукт;
* Разработаны функциональная диаграмма программного продукта, диаграмма потоков данных программных модулей продукта;
* Разработаны функциональная схема программного продукта, составлены блок-схемы программных модулей программного продукта;
* Разработаны коды программных модулей программного продукта;
* Разработан пользовательский интерфейса программного продукта в визуальной среде;
* Выполнена интеграция программных модулей в программный продукт;
* Разработана процедура тестирования программного продукта;
* Выполнено тестирование программного продукта. Результат тестирования оформлен протоколом тестирования;
* Разработана справочная система программного продукта;
* Разработано руководство оператора (пользователя);
* Создан аккаунт в GitHub с папкой проекта
* Составлен отчет о выполнении;

Дальнейшую перспективу разработки составляет возможность улучшения пользовательского интерфейса, добавления новых, более оптимизированных алгоритмов сортировки и добавления новых параметров сортировки (порядок сортировки, сортировка слов и т.д.).

Все материалы выполненной работы находятся в репозитории GitHub по ссылке: https://github.com/SokolovUcheba.

**СПИСОК ЛИТЕРАТУРЫ**

1. ПРАВИЛА ПОВЕДЕНИЯ И ТЕХНИКА БЕЗОПАСНОСТИ В КАБИНЕТЕ ИНФОРМАТИКИ // belledahlia.jimdo.com URL: https://belledahlia.jimdo.com/общие-темы/тб-и-сангигиена/ (дата обращения: 21.03.2023).
2. Краткий обзор языка C# // learn.microsoft.com URL: https://learn.microsoft.com/ru-ru/dotnet/csharp/tour-of-csharp/ (дата обращения: 21.03.2023).
3. Руководство по классическим приложениям (Windows Forms .NET) // learn.microsoft.com URL: https://learn.microsoft.com/ru-ru/dotnet/desktop/winforms/overview/?view=netdesktop-6.0 (дата обращения: 21.03.2023).
4. Богданов О.Н. Коктейльная сортировка // russianblogs.com – URL: https://russianblogs.com/article/8827516452/. – (дата обращения 20.03.2023)
5. Репейников Г.Н. Сортировка массивов методом быстрой сортировки // techiedelight.com – URL: https://www.techiedelight.com/ru/quicksort/. – (дата обращения 21.03.2023)
6. Дедулев В.В. Алгоритм сортировки Расчёской // thecode.media – URL https://thecode.media/comb-sort/. – (дата обращения 21.03.2023)
7. Пильщикой Б.Р. Сортировка Вставкой // techiedelight.com – URL: <https://www.techiedelight.com/ru/insertion-sort-iterative-recursive/>. – (дата обращения 21.03.2023)