

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 **4ПКС18**

форма обучения очная
(очная, заочная)

Бабенко Василий Владимирович

(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.

ЗАДАНИЕ
на учебную практику по профессиональному модулю
ПМ.03 Участие в интеграции программных модулей

Выдано студенту Таврического колледжа (структурное подразделение)
ФГАОУ ВО «КФУ им. В. И. Вернадского».
Специальность 09.02.03 Программирование в компьютерных системах
4 курса группы 4ПКС18

Бабенко Василий Владимирович
(ФИО обучающегося)

Для прохождения учебной практики на базе:

Таврический колледж (структурное подразделение) ФГАОУ «КФУ им. В.И. Вернадского»
(полное наименование предприятия прохождения практики)

Дата начала практики 16 марта 2023 г.

Дата окончания практики 22 марта 2023 г.

Дата сдачи отчёта по практике 22 марта 2023 г.

Цель практики: формирование и развитие общих и профессиональных компетенций по модулю ПМ.03 Участие в интеграции программных модулей.

Задачи учебной практики: закрепление навыков 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.

Задание для выполнения.

1. Необходимо разработать программный комплекс по демонстрации работы алгоритмов сортировки массивов данных (реализовать не менее 4 алгоритмов сортировки, которые выбрать самостоятельно):

1.1 Разработать техническое задание на программный продукт.

1.2 Разработать спецификацию на программный продукт.

1.3 Разработать функциональную диаграмму программного продукта, диаграмму потоков данных программных модулей продукта.

1.4 Разработать функциональную схему программного продукта, составить блок-схемы программных модулей программного продукта.

1.5 Разработать коды программных модулей программного продукта.

1.6 Разработать пользовательский интерфейс программного продукта в визуальной среде.

1.7 Выполнить интеграцию программных модулей в программный продукт.

1.8 Разработать процедуру тестирования программного продукта. Выполнить тестирование программного продукта. Результат тестирования оформить протоколом тестирования.

1.9 Разработать справочную систему программного продукта.

1.10 Разработать руководства оператора (пользователя).

2. Создать аккаунт в GitHub. Создать папку проекта. В папку загрузить разработанный программный комплекс, всю разработанную документацию к проекту (п.п.1.1 – 1.10).

3. Составить отчет о выполнении.

Требования к отчету по практике.

Согласно ГОСТ 7.32-2001 текст отчета печатается на одной стороне листа белой бумаги формата А4. Общий объем отчета – 30 листов. Первым листом является титульный лист, оформляемый по утвержденному образцу. Второй лист – лист данного задания, напечатанный с двух сторон, третий лист – оглавление.

При компьютерном наборе текста необходимо установить следующие параметры форматирования документа: шрифт Times New Roman, кегль 14, стиль - обычный (normal), шрифт заголовков – полужирный, цвет шрифта - черный; поля: левое - 30 мм, верхнее и нижнее - 20 мм, правое - 10 мм; междустрочный интервал - 1,5; выравнивание текста производится по ширине; отступ первой строки каждого абзаца – 1,25 см.

Отчет должен содержать следующие структурные элементы: титульный лист; оглавление; введение; основная часть, состоящая из двух разделов; заключение; список использованных источников; приложения (при необходимости).

Во введении на 2 страницах необходимо:

- указать цели и задачи практики, общие сведения о разработке программного обеспечения;

В основной части отчета (25-30 листов) необходимо изложить подробный отчет о выполнении пунктов задания, с описанием разработанных технического задания, спецификаций, диаграмм, функциональных схем, блок-схем алгоритмов модулей, процедур тестирования, результатов тестирования, справочной системы и руководства пользователя. Привести ссылку на проект на GitHub.

В заключении на 2 страницах необходимо сделать выводы о проделанной работе.

Список использованных источников должен содержать 4-5 изученных источников.

В приложениях (в случае необходимости) указываются прочие материалы, не вошедшие в основную часть отчета (схема, инструкции, таблицы и т.п.).

Задание выдал «16» марта 2023 г. _____ / Руденко А.В. /
(подпись) (Ф.И.О.)

Дата	Краткое содержание выполненных работ
16.03.23	Вводный инструктаж по прохождению учебной практики. Изучение, наличия компьютерных систем и программного обеспечения.
16.03.23	Разработка технического задания на программный продукт.
16.03.23	Разработка спецификации на программный продукт.
17.03.23	Разработка функциональной диаграммы программ.
17.03.23	Разработка диаграмм потоков данных программ.
17.03.23	Разработка функциональной схемы программ.
20.03.23	Создание блок-схем программ в специализированных программах.
20.03.23	Разработка кода программы в соответствии со стандартами кодирования.
20.03.23	Разработка пользовательского интерфейса на программный продукт в визуальной среде разработки программных продуктов.
20.03.23	Сборка подпрограмм в единую программу.
21.03.23	Интеграция программных модулей.
21.03.23	Написание кода программы с использованием определенной нотации по оформлению кода.
21.03.23	Функциональное тестирование и отладка программ.
22.03.23	Разработка эксплуатационной документации на программный продукт.
22.03.23	Применение различных сред для разработки программ.
22.03.23	Разработка справочной системы программного продукта.
22.03.23	Разработка руководства оператора.
22.03.23	Защита отчета.

[illegible]

ХАРАКТЕРИСТИКА

(отзыв) от руководителя учебной практики

на обучающегося Таврического колледжа (структурное подразделение) ФГАОУ
ВО «КФУ им. В. И. Вернадского»

Бабенко Василий Владимирович

(фамилия, имя, отчество)

группы **4ПКС18** специальности **09.02.03 Программирование в компьютерных
системах**

проходившего с **16 марта 2023 г. по 22 марта 2023 г.**

учебную практику по профессиональному модулю **ПМ.03 Участие в интеграции
программных модулей** на базе:

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

(полное наименование предприятия прохождения практики)

Показатели выполнения учебных заданий:

Уровень теоретической подготовки отличный

Освоенные виды работ, качество, самостоятельность, интерес, инициатива

Во время прохождения практики Бабенко Василий Владимирович закрепил навыки
разработки программного обеспечения, разработки документации на программный
продукт, использования методов для получения кода с заданной
функциональностью и степенью качества

Трудовая дисциплина и соблюдение техники безопасности соблюдались
неукоснительно

Особые замечания и предложения руководителя практики на основании
вышеизложенного можно заключить, что Бабенко Василий заслуживает по итогам
учебной практики оценку «отлично»

Оценка практики отлично

(отлично, хорошо, удовлетворительно)

Дата «22» марта 2023 г.

Руководитель практики

Преподаватель

(должность)

М.П.

(подпись)

/Руденко А.В./

(Ф.И.О.)

АТТЕСТАЦИОННЫЙ ЛИСТ ПО УЧЕБНОЙ ПРАКТИКЕ
по модулю ПМ.03 Участие в интеграции программных модулей

Бабенко Василий Владимирович

(фамилия, имя, отчество)

обучающийся на 4 курсе группа 4ПКС18 специальности 09.02.03 Программирование в компьютерных системах в Таврическом колледже (структурное подразделение) ФГАОУ ВО «КФУ им. В. И. Вернадского» успешно прошел производственную практику по модулю ПМ.03 Участие в интеграции программных модулей

в период с **16 марта 2023 г. по 22 марта 2023 г.**

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

(полное наименование предприятия прохождения практики)

Виды и качество выполнения работ по модулю ПМ.03 Участие в интеграции программных модулей

Виды работ, выполненных обучающимся во время учебной практики	Компетенции (ОК и ПК)	Качество работ (от 1 до 10 баллов)
Изложение требований к программному обеспечению. Разработка технического задания на программный продукт.	ПК 3.1 ОК 1, ОК 2	
Выполнение проектирования программного обеспечения с использованием специализированных программных пакетов. Составление функциональной диаграммы, диаграмм потоков данных, функциональной схемы программы.	ПК 3.2 ОК 2, ОК 3, ОК 4	
Изложение основных методов и средств эффективной разработки программного продукта	ПК 3.3, ОК 5, ОК 6,	
Разработка тестовых наборов и тестовых сценариев;	ПК 3.4, ОК 4, ОК 6	
Изложение стандартов качества программного обеспечения	ПК 3.5, ОК 4, ОК 5, ОК 8	
Изложение методов и средства разработки программной документации	ПК 3.6, ОК3 – ОК8	
Разработка технической документации, эксплуатационной документации на программный продукт, руководства оператора (пользователя)	ПК 3.6, ОК 4, ОК 8, ОК 9	
Оформление отчета по практике	ОК 2	

Подпись руководителя практики
от колледжа

преподаватель
_____/ Руденко А.В.
(Ф.И.О.)

МП

Дата 22 марта 2023 г.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
ГЛАВА 1. ТЕХНИЧЕСКОЕ ЗАДАНИЕ	5
1.1 Разработка технического задания.....	5
1.2 Разработка спецификаций	6
1.3 Разработка диаграмм.....	7
ГЛАВА 2. МОДУЛИ ПРОГРАММЫ.....	11
2.1 Модуль интерфейса.....	11
2.2 Модуль BubbleSort	12
2.3 Модуль GnomeSorter	13
2.4 Модуль QuickSortet	15
2.5 Модуль InsertionSorter	16
ЗАКЛЮЧЕНИЕ	18
СПИСОК ЛИТЕРАТУРЫ.....	20
ПРИЛОЖЕНИЕ А	21

ВВЕДЕНИЕ

Я, Бабенко Василий Владимирович, проходил учебную практику на базе: Таврического колледжа (структурное подразделение) ФГАОУ «КФУ им. В.И. Вернадского».

Дата начала практики: 16 марта 2023 г.

Дата окончания практики: 22 марта 2023 г.

Дата сдачи отчёта по практике: 22 марта 2023 г.

Цель практики:

Формирование и развитие общих и профессиональных компетенций по модулю ПМ.03 Участие в интеграции программных модулей.

Задачи учебной практики:

Закрепление навыков разработки программного обеспечения;

Использование методов для получения кода с заданной функциональностью и степенью качества;

Разработка документации на программный продукт;

Знание моделей процесса разработки программного обеспечения, основных принципов процесса разработки программного обеспечения;

Знание основных подходов к интегрированию программных модулей, основных методов и средств эффективной разработки ПО;

Задание для выполнения:

1. Необходимо разработать программный комплекс по демонстрации работы алгоритмов сортировки массивов данных (реализовать не менее 4 алгоритмов сортировки, которые выбрать самостоятельно):

1.1 Разработать техническое задание на программный продукт.

1.2 Разработать спецификацию на программный продукт.

1.3 Разработать функциональную диаграмму программного продукта, диаграмму потоков данных программных модулей продукта.

1.4 Разработать функциональную схему программного продукта, составить блок-схемы программных модулей программного продукта.

1.5 Разработать коды программных модулей программного продукта.

- 1.6 Разработать пользовательский интерфейс программного продукта в визуальной среде.
 - 1.7 Выполнить интеграцию программных модулей в программный продукт.
 - 1.8 Разработать процедуру тестирования программного продукта. Выполнить тестирование программного продукта. Результат тестирования оформить протоколом тестирования.
 - 1.9 Разработать справочную систему программного продукта.
 - 1.10 Разработать руководства оператора (пользователя).
2. Создать аккаунт в GitHub. Создать папку проекта. В папку загрузить разработанный программный комплекс, всю разработанную документацию к проекту (п.п.1.1 – 1.10).
 3. Составить отчет о выполнении.

ГЛАВА 1. ТЕХНИЧЕСКОЕ ЗАДАНИЕ

1.1 Разработка технического задания

Наименование программы: "SortArray".

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

Программный комплекс предоставляет пользовательский интерфейс, модуль сортировки пузырьков, модуль быстрой сортировки, модуль сортировки вставкой, модуль гномьей сортировки.

Программный комплекс реализован на языке программирования C#, в визуальной среде разработки Visual Studio Code, для операционной системы Windows 10. В программе используются массивы размерностью до 150, с целыми данными, где скорость получения результата в среднем меньше секунды.

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:

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 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.

Этапы разработки:

1. Разработка программы
2. Разработка программной документации.
3. Испытание программы.

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

1.2 Разработка спецификаций

Программа “SortArray” реализована на языке программирования C#, предназначена для выполнения задач, связанных с сортировкой различного размера массивов разными методами, среди них: метод пузырьком, сортировка вставками, гномья сортировка, и быстрая сортировка, в программе представлены следующие модули: Модуль интерфейса, модуль BubbleSort, модуль GnomeSorter, модуль QuickSortet, модуль InsertionSorter.

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

1.3 Разработка диаграмм

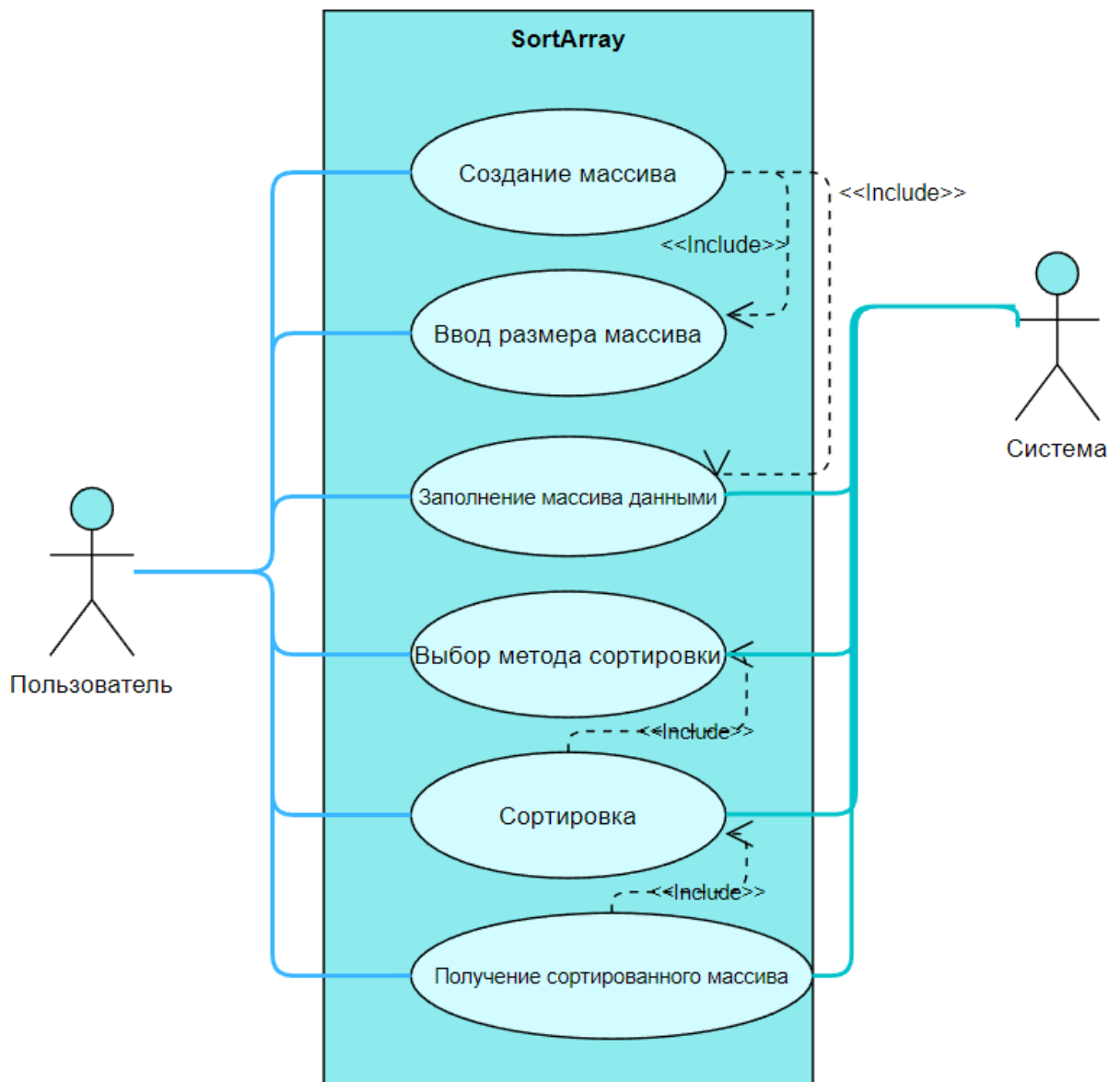


Рисунок 1 – UML диаграмма

На данном рисунке представлена UML диаграмма программного комплекса. Субъекты диаграммы – пользователь и система. Пользователь задает размер массива и его наполнение, а также выбирает метод сортировки. Система считывает массив и сортирует его с помощью выбранного метода, после чего выводит пользователю.

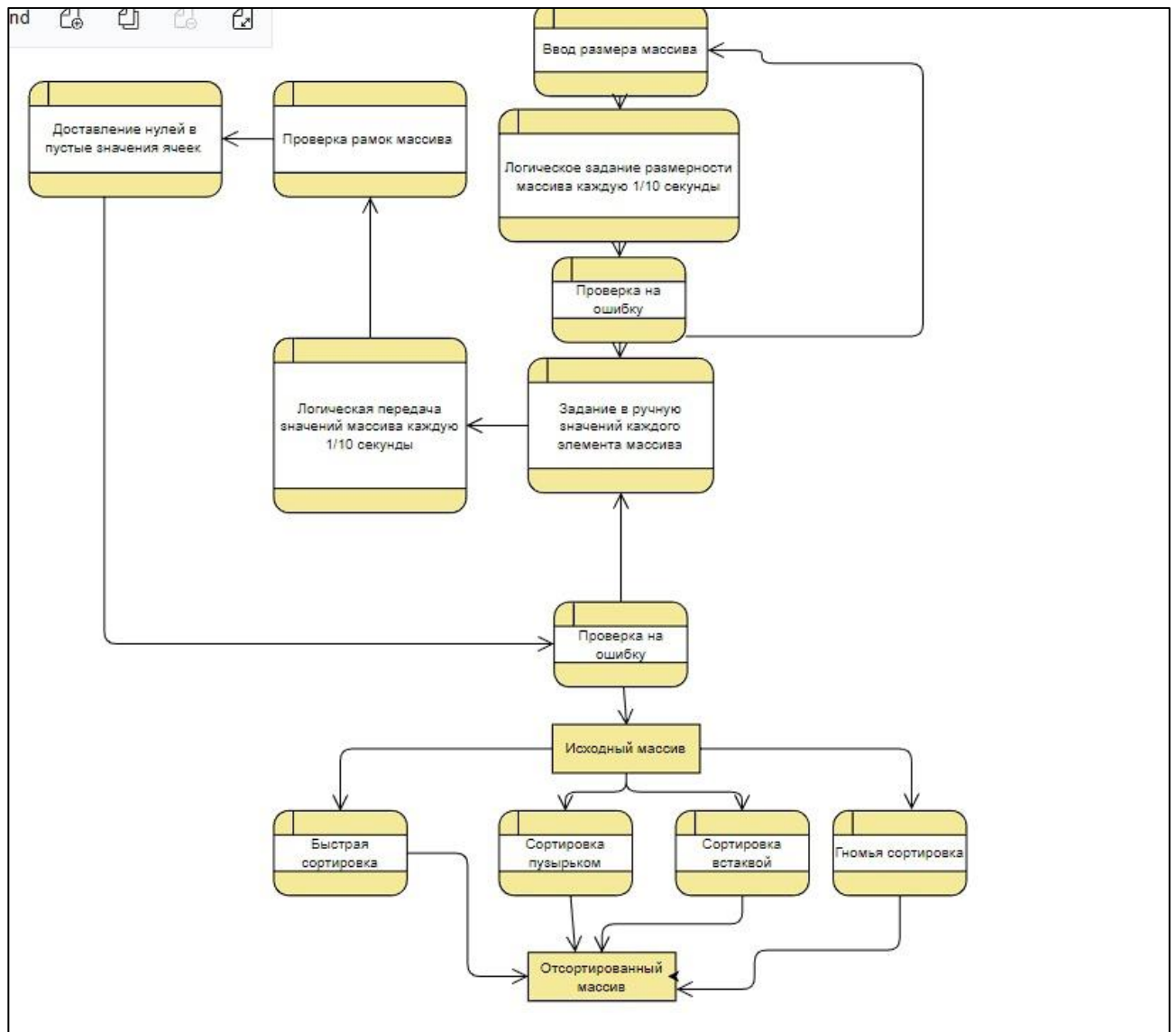


Рисунок 2 – Диаграмма потока данных

На данном рисунке представлена диаграмма потока данных программного комплекса SortArray.

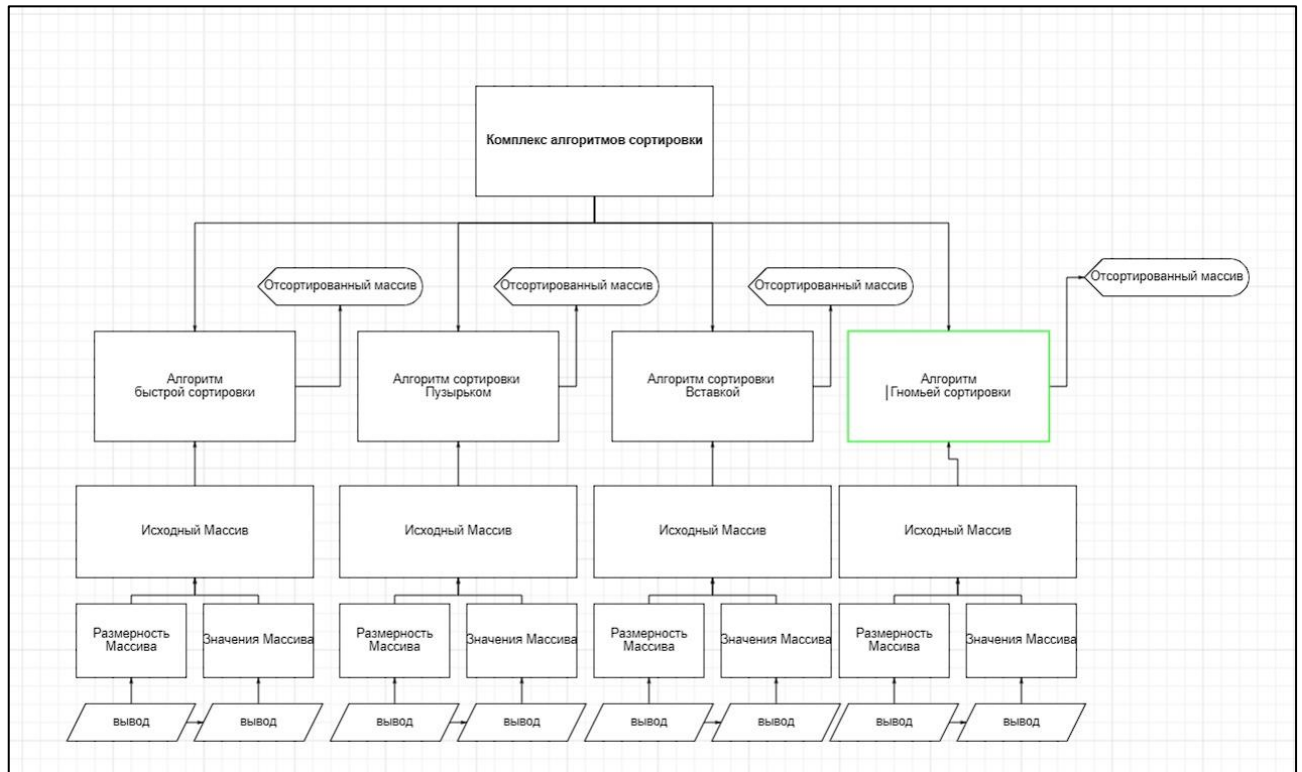


Рисунок 3 – Функциональная схема продукта

На данном рисунке представлена функциональная схема продукта.

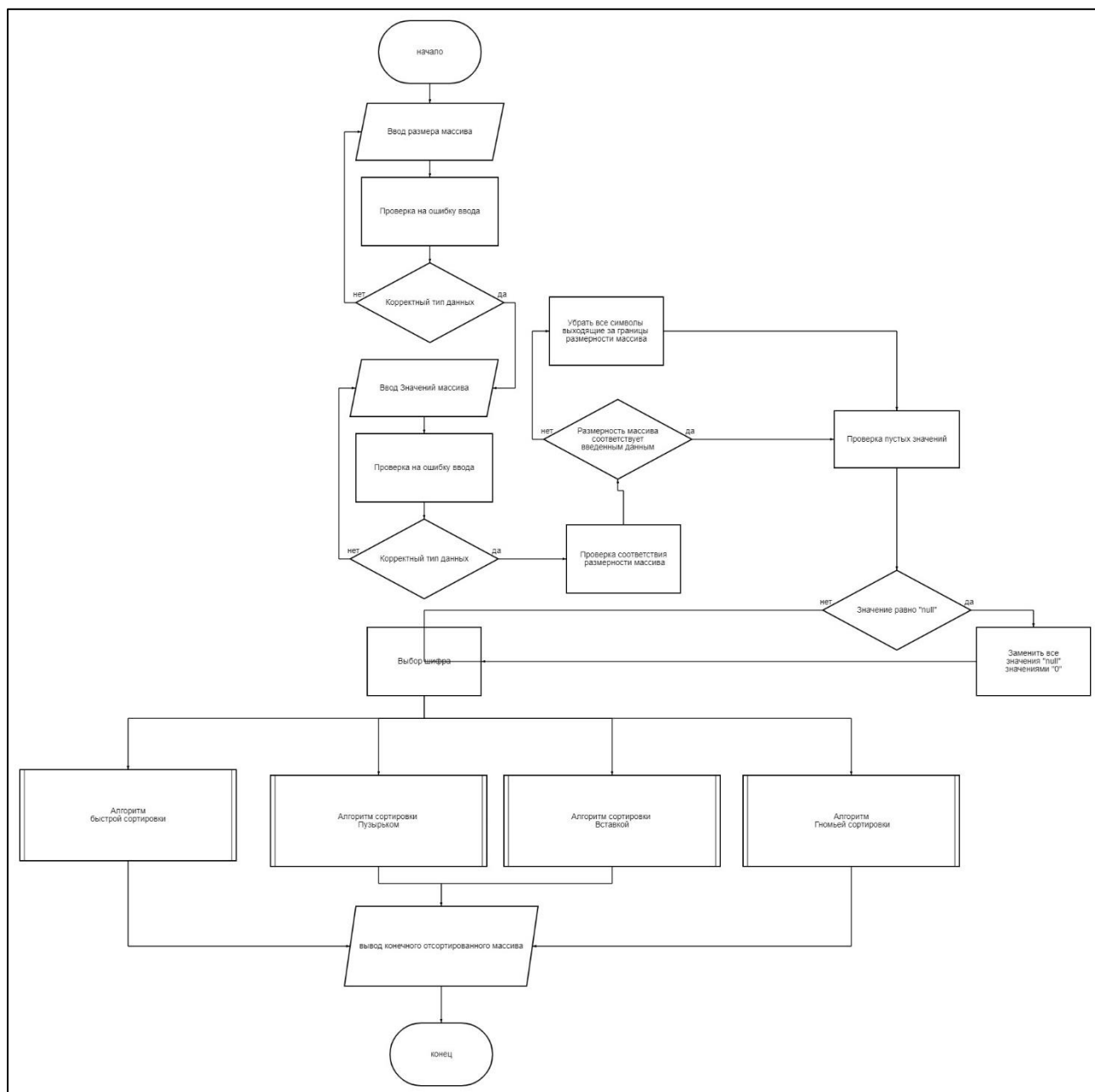


Рисунок 4 – Блок-схема программы
 На данном рисунке представлена блок-схема программы.

ГЛАВА 2. МОДУЛИ ПРОГРАММЫ

2.1 Модуль интерфейса

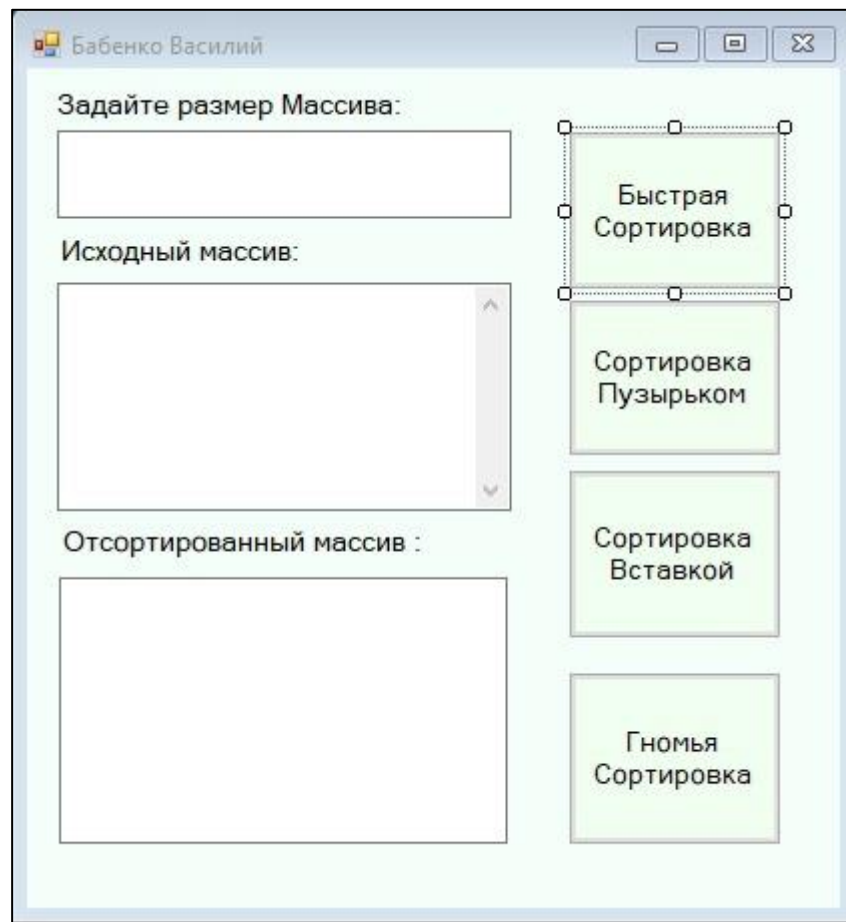


Рисунок 5 – Модуль интерфейса

На данном рисунке представлен общий интерфейс программы, в нем реализованы такие кнопки как: “Быстрая сортировка”, “Сортировка Пузырьком”, “Сортировка вставкой”, “Гномья сортировка”, а также окна: “Задайте размер Массива:”, “Исходный массив:”, “Отсортированный массив:”.

2.2 Модуль BubbleSort

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Babenko_Bubble
{
    public class BubbleSort
    {
        public static int[] SortArray(int[] array)
        {
            int length = array.Length;

            int temp = array[0];

            for (int i = 0; i < length; i++)
            {
                for (int j = i + 1; j < length; j++)
                {
                    if (array[i] > array[j])
                    {
                        temp = array[i];
                        array[i] = array[j];
                        array[j] = temp;
                    }
                }

                return array;
            }
        }
    }
}
```

Рисунок 6 – Модуль BubbleSort

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

Сортировка пузырьком — один из самых известных алгоритмов сортировки. Здесь нужно последовательно сравнивать значения соседних элементов и менять числа местами, если предыдущее оказывается больше последующего. Таким образом элементы с большими значениями оказываются в конце списка, а с меньшими остаются в начале. [1]

2.3 Модуль GnomeSorter

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Babenko_GnomeSort
{
    public class GnomeSorter
    {
        static void Swap(ref int item1, ref int item2)
        {
            var temp = item1;
            item1 = item2;
            item2 = temp;
        }

        public static int[] SortArray(int[] unsortedArray)
        {
            var index = 1;
            var nextIndex = index + 1;

            while (index < unsortedArray.Length)
            {
                if (unsortedArray[index - 1] < unsortedArray[index])
                {
                    index = nextIndex;
                    nextIndex++;
                }
                else
                {
                    Swap(ref unsortedArray[index - 1], ref unsortedArray[index]);
                    index--;
                    if (index == 0)
                    {
                        index = nextIndex;
                        nextIndex++;
                    }
                }
            }

            return unsortedArray;
        }
    }
}
```

Рисунок 7 - Модуль GnomeSorter

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

Гномья сортировка основана на технике, используемой обычным голландским садовым гномом (нидерл. *tuinkabouter*). Это метод, которым садовый гном сортирует линию цветочных горшков. По существу, он смотрит на текущий и предыдущий садовые горшки: если они в правильном порядке, он шагает на один горшок вперёд, иначе он меняет их местами и шагает на один горшок назад. [2]

2.4 Модуль QuickSortet

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Babenko_QuickSort
{
    public class QuickSortet
    {
        static void Swap(ref int x, ref int y)
        {
            var t = x;
            x = y;
            y = t;
        }

        static int Partition(int[] array, int minIndex, int maxIndex)
        {
            var pivot = minIndex - 1;
            for (var i = minIndex; i < maxIndex; i++)
            {
                if (array[i] < array[maxIndex])
                {
                    pivot++;
                    Swap(ref array[pivot], ref array[i]);
                }
            }

            pivot++;
            Swap(ref array[pivot], ref array[maxIndex]);
            return pivot;
        }

        static int[] QuickSort(int[] array, int minIndex, int maxIndex)
        {
            if (minIndex >= maxIndex)
            {
                return array;
            }

            var pivotIndex = Partition(array, minIndex, maxIndex);
            QuickSort(array, minIndex, pivotIndex - 1);
            QuickSort(array, pivotIndex + 1, maxIndex);

            return array;
        }

        public static int[] SortArray(int[] array)
        {
            return QuickSort(array, 0, array.Length - 1);
        }
    }
}
```

Рисунок 8 – Модуль QuickSortet

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

Быстрая сортировка - это сортировка сравнения, означающая, что она может сортировать элементы любого типа, для которых определено отношение "меньше" (формально, общий порядок). Большинство реализаций быстрой сортировки нестабильны, что означает, что относительный порядок одинаковых элементов сортировки не сохраняется. [3]

2.5 Модуль InsertionSorter

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Babenko_InsertionSort
{
    public class InsertionSorter
    {
        public static int[] SortArray(int[] array)
        {
            int length = array.Length;

            for (int i = 1; i < length; i++)
            {
                var key = array[i];
                var flag = 0;
                for (int j = i - 1; j >= 0 && flag != 1;)
                {
                    if (key < array[j])
                    {
                        array[j + 1] = array[j];
                        j--;
                        array[j + 1] = key;
                    }
                    else flag = 1;
                }
            }
            return array;
        }
    }
}
```

Рисунок 9 – Модуль InsertionSorter

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

Сортировка вставками – простой алгоритм сортировки, преимущественно использующийся в учебном программировании. К положительной стороне метода относится простота реализации, а также его эффективность на частично упорядоченных последовательностях, и/или состоящих из небольшого числа элементов. Тем не менее, высокая вычислительная сложность не позволяет рекомендовать алгоритм в повсеместном использовании. [4]

На каждом шаге алгоритма мы выбираем один из элементов входных данных и вставляем его на нужную позицию в уже отсортированном списке до тех пор, пока набор входных данных не будет исчерпан. Метод выбора очередного элемента из исходного массива произволен; может использоваться практически любой алгоритм выбора. Обычно (и с целью получения устойчивого алгоритма сортировки), элементы вставляются по порядку их появления во входном массиве. [5]

ЗАКЛЮЧЕНИЕ

Я, Бабенко Василий Владимирович, проходил учебную практику на базе: Таврического колледжа (структурное подразделение) ФГАОУ «КФУ им. В.И. Вернадского».

Дата начала практики: 16 марта 2023 г.

Дата окончания практики: 22 марта 2023 г.

Дата сдачи отчёта по практике: 22 марта 2023 г.

Была выполнена следующая цель практики:

Формирование и развитие общих и профессиональных компетенций по модулю ПМ.03 Участие в интеграции программных модулей.

Были выполнены следующие задачи учебной практики:

Закрепление навыков разработки программного обеспечения;

Использование методов для получения кода с заданной функциональностью и степенью качества;

Разработка документации на программный продукт;

Знание моделей процесса разработки программного обеспечения, основных принципов процесса разработки программного обеспечения;

Знание основных подходов к интегрированию программных модулей, основных методов и средств эффективной разработки ПО;

Был выполнен весь план задания для выполнения:

Был разработан программный комплекс по демонстрации работы алгоритмов сортировки массивов данных (реализованы 4 алгоритма сортировки):

Было разработано техническое задание на программный продукт.

Была разработана спецификация на программный продукт.

Была разработана функциональная диаграмма программного продукта, диаграмма потоков данных программных модулей продукта.

Была разработана функциональная схема программного продукта, составлена блок-схема программных модулей программного продукта.

Был разработан код программных модулей программного продукта.

Был разработан пользовательский интерфейс программного продукта в визуальной среде.

Была выполнена интеграция программных модулей в программный продукт.

Была разработана процедура тестирования программного продукта.

Было выполнено тестирование программного продукта.
Результат тестирования оформлен протоколом тестирования.

Была разработана справочная система программного продукта.

Было разработано руководство оператора (пользователя).

Ссылка на репозиторий с выполненной работой: <https://github.com/LLIenaT/UP>

СПИСОК ЛИТЕРАТУРЫ

1. Котлин А.П. Алгоритмы сортировки [электронный ресурс] [bimlibik.github.io](https://bimlibik.github.io/posts/sorting-algorithm/) – URL: <https://bimlibik.github.io/posts/sorting-algorithm/>. – (дата обращения 20.03.2023)
2. Липачёв Е.К. Технология программирования. Методы сортировки данных [электронный ресурс] [kpfu.ru](https://kpfu.ru/staff_files/F1066117885/Metody_sortirovki.pdf) – URL: https://kpfu.ru/staff_files/F1066117885/Metody_sortirovki.pdf. – (дата обращения 20.03.2023)
3. Сенегалович А.В. Основные виды сортировок и примеры их реализации [электронный ресурс] [academy.yandex.com](https://academy.yandex.com/journal/osnovnye-vidy-sortirovok-i-primery-ikh-realizatsii) – URL: <https://academy.yandex.com/journal/osnovnye-vidy-sortirovok-i-primery-ikh-realizatsii>. – (дата обращения 20.03.2023)
4. Баргузин В.В. Алгоритмы сортировки массивов. [электронный ресурс] [study.muctr.ru](https://study.muctr.ru/pluginfile.php/263589/mod_resource/content/1/сортировка.pdf) – URL: https://study.muctr.ru/pluginfile.php/263589/mod_resource/content/1/сортировка.pdf. – (дата обращения 20.03.2023)
5. Граф А.Е. Алгоритмы сортировки и поиска [электронный ресурс] [prog-cpp.ru](https://prog-cpp.ru/algorithm-sort/) – URL: <https://prog-cpp.ru/algorithm-sort/>. – (дата обращения 20.03.2023)

ПРИЛОЖЕНИЕ А

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using Babenko_BubbleSort;
using Babenko_GnomeSort;
using Babenko_InsertionSort;
using Babenko_QuickSort;
namespace ArraySorter
{
    public partial class Form1 : Form
    {
        int n = 0;
        public Form1()
        {
            InitializeComponent();
        }

        private void textBox3_TextChanged(object sender, EventArgs e)
        {
        }

        private void button1_Click(object sender, EventArgs e)
```

```

{

    string myArr = textBox4.Text;
    int[] result = myArr.ToString().Select(o => Convert.ToInt32(o) -
48).ToArray();
    string texter = string.Join("",BubbleSort.SortArray(result));
    textBox3.Text = texter;
}

private void textBox1_TextChanged(object sender, EventArgs e)
{
    textBox4.Text = "";
    try
    {
        n = Convert.ToInt32(textBox1.Text);
    }
    catch
    {
        textBox1.Text = "";
        MessageBox.Show(
"Неверный тип данных",
"Ошибка",
MessageBoxButtons.OK,
MessageBoxIcon.Information,
MessageBoxDefaultButton.Button1);

    }

    int[] x = new int[n];
    for (int i = 0; i < x.Length; i++)

```

```

{
    textBox4.Text = textBox4.Text + x[i];
}
}

```

```

private void button2_Click(object sender, EventArgs e)
{
    string myArr = textBox4.Text;
    int[] result = myArr.ToString().Select(o => Convert.ToInt32(o) -
48).ToArray();
    string texter = string.Join("", GnomeSorter.SortArray(result));
    textBox3.Text = texter;
}

```

```

private void textBox4_TextChanged(object sender, EventArgs e)
{
    int[] x = new int[n];
    if (x.Length > textBox4.TextLength)
    {
        int a = x.Length - textBox4.TextLength;
        while (a > 0)
        {
            a--;
            textBox4.Text = textBox4.Text + 0;
        }
    }
}

```

```

else
if (x.Length < textBox4.TextLength)
{
    int a = textBox4.TextLength - x.Length;

    textBox4.Text = textBox4.Text.Remove(textBox4.Text.Length - a);
    a = 0;

}
}

```

```

private void button3_Click(object sender, EventArgs e)
{
    string myArr = textBox4.Text;
    int[] result = myArr.ToString().Select(o => Convert.ToInt32(o) -
48).ToArray();
    int resl = result.Length;
    string texter = string.Join("", InsertionSorter.SortArray(result));
    textBox3.Text = texter;
}

```

```

private void button4_Click(object sender, EventArgs e)
{
    string myArr = textBox4.Text;
    int[] result = myArr.ToString().Select(o => Convert.ToInt32(o) -
48).ToArray();
    int resl = result.Length;
    string texter = string.Join("", QuickSortet.SortArray(result));
    textBox3.Text = texter;
}

```

```
}
```

```
private void label2_Click(object sender, EventArgs e)
```

```
{
```

```
}
```

```
}
```

```
}
```