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.

**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 14

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. 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 интерфейса программного продукта в визуальной среде.
   7. Выполнить интеграцию программных модулей в программный продукт.
   8. Разработать процедуру тестирования программного продукта. Выполнить тестирование программного продукта. Результат тестирования оформить протоколом тестирования.
   9. Разработать справочную систему программного продукта.
   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.

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 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.

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

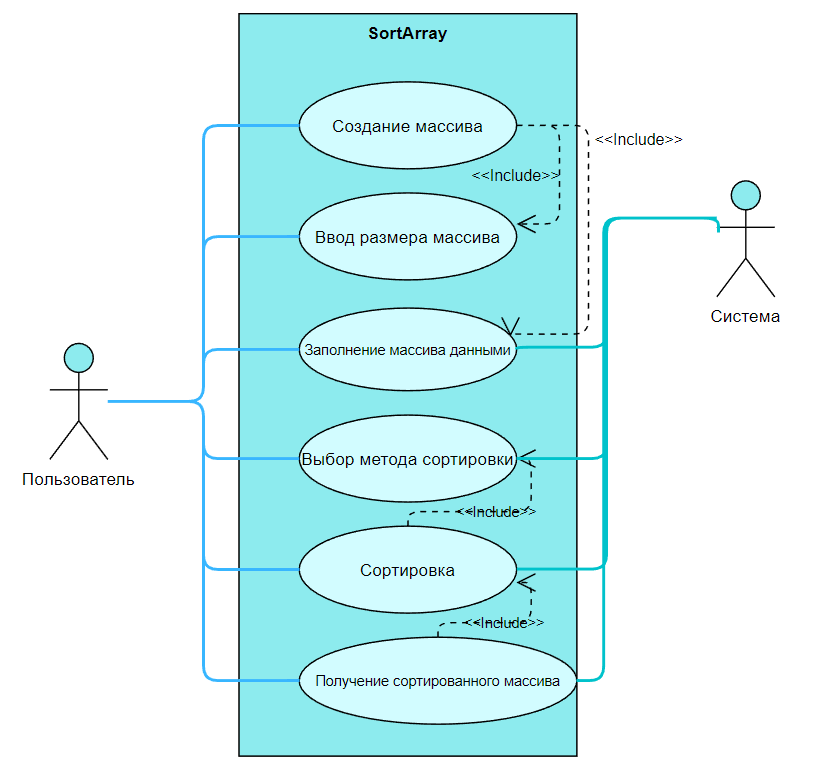
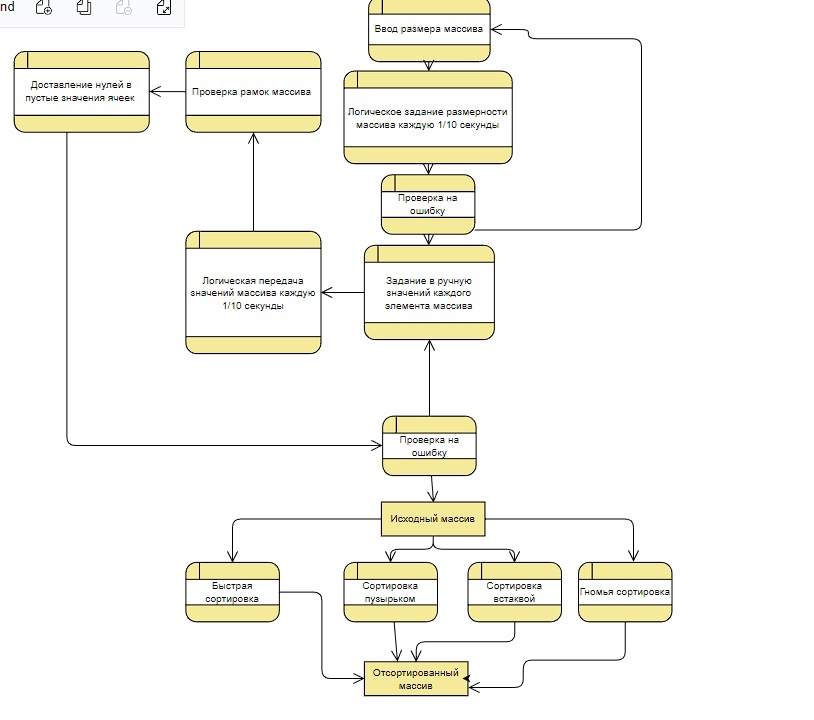
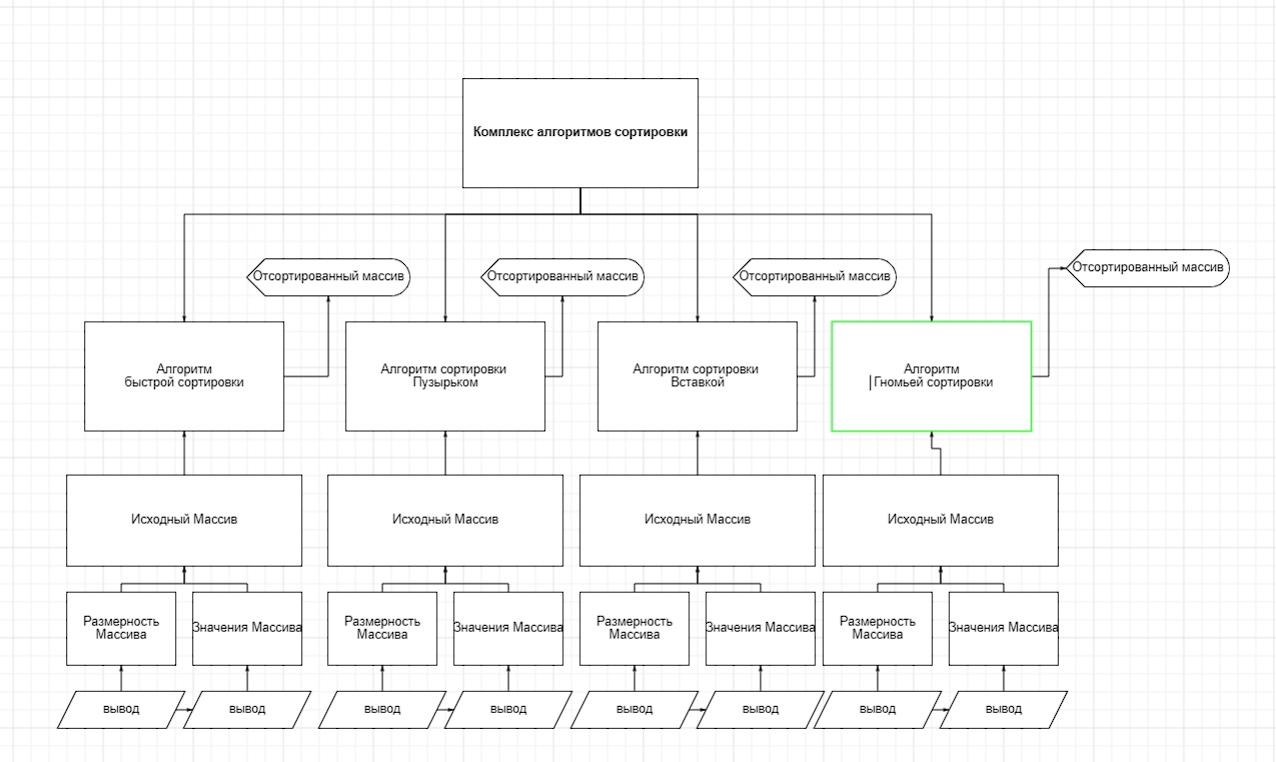


Рисунок 1 – UML диаграмма

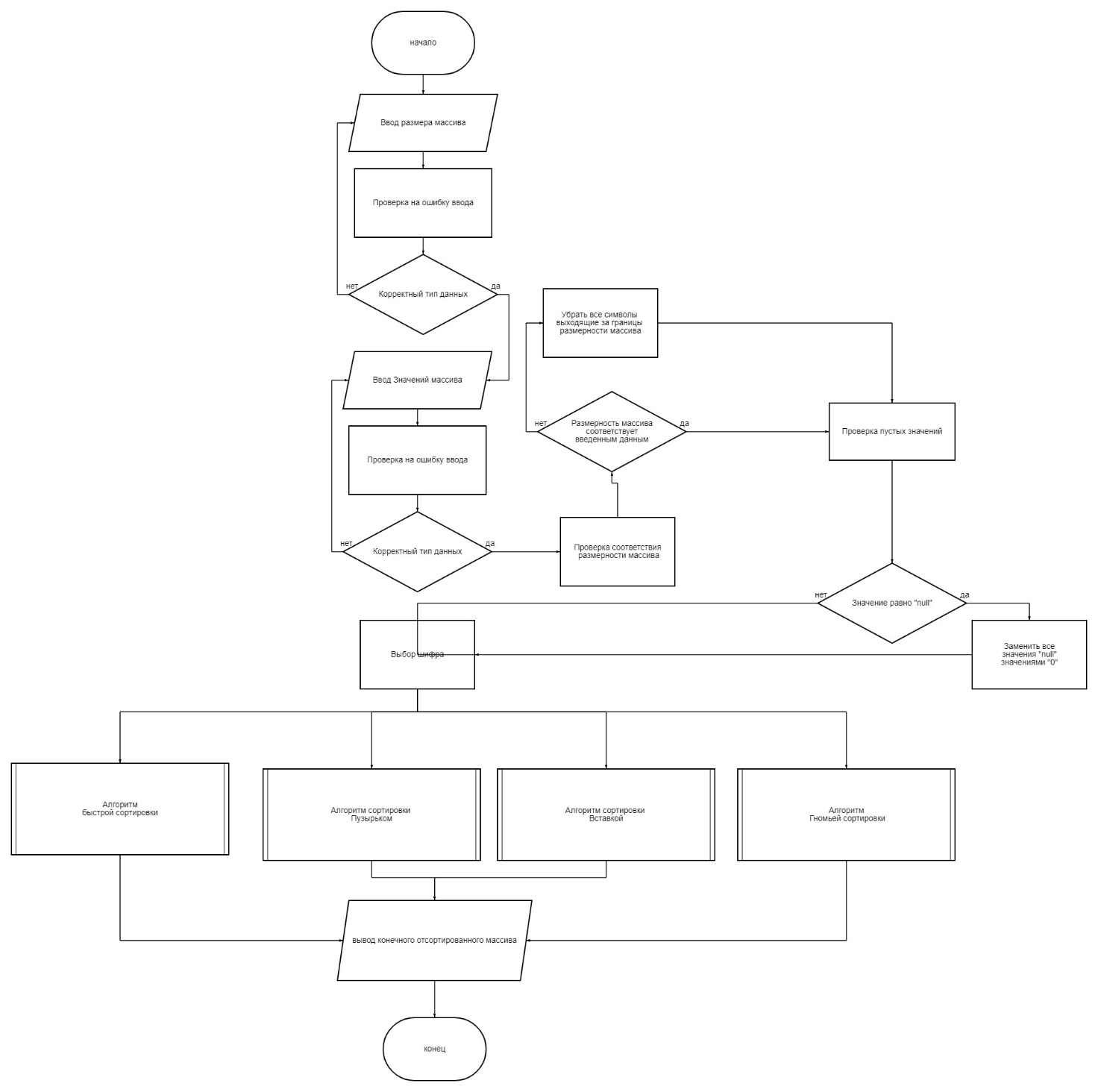
На данном рисунке представлена UML диаграмма программного комплекса. Субъекты диаграммы – пользователь и система. Пользователь задает размер массива и его наполнение, а также выбирает метод сортировки. Система считывает массив и сортирует его с помощью выбранного метода, после чего выводит пользователю.

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

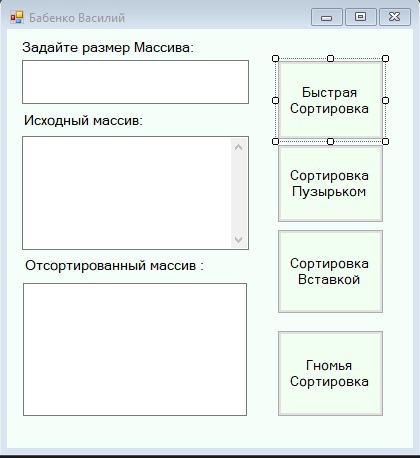


Рисунок 5 – Модуль интерфейса

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

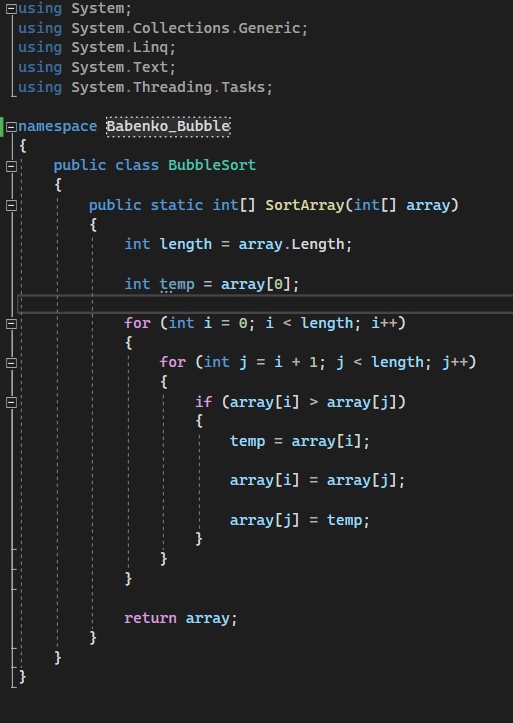


Рисунок 6 – Модуль BubbleSort

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

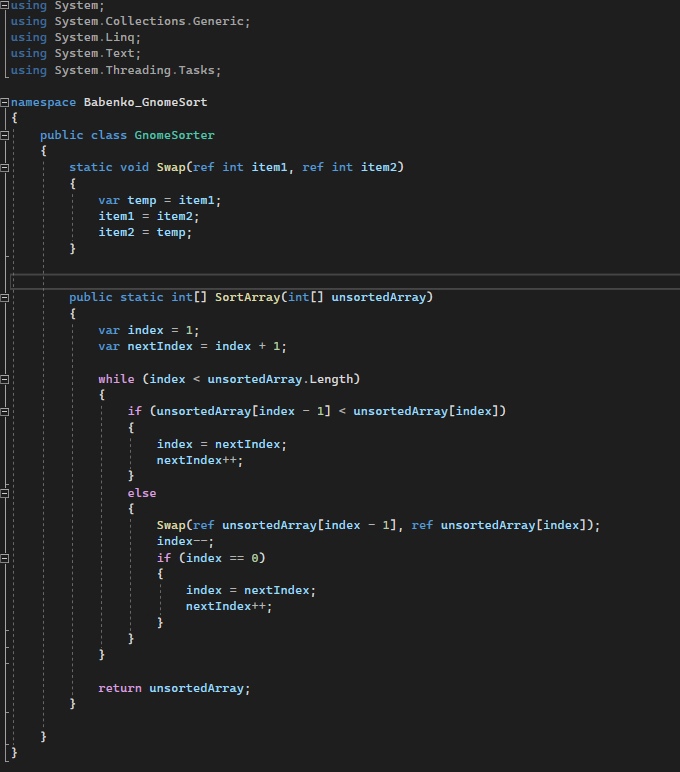
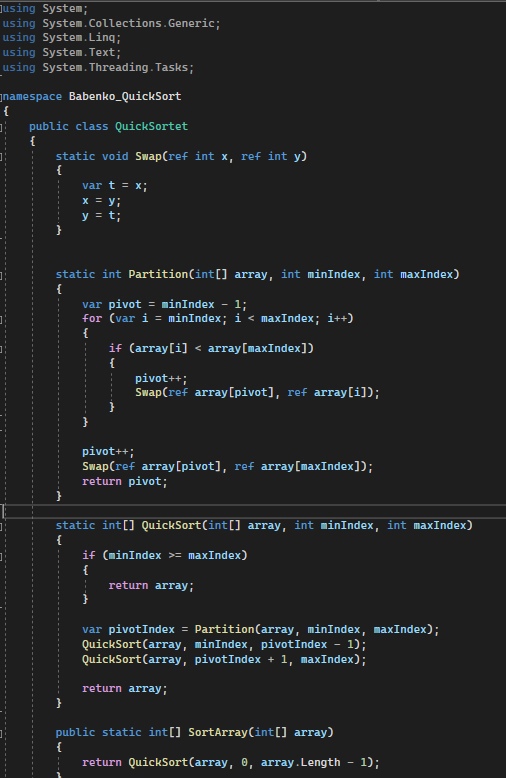


Рисунок 7 – Модуль GnomeSorter

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]

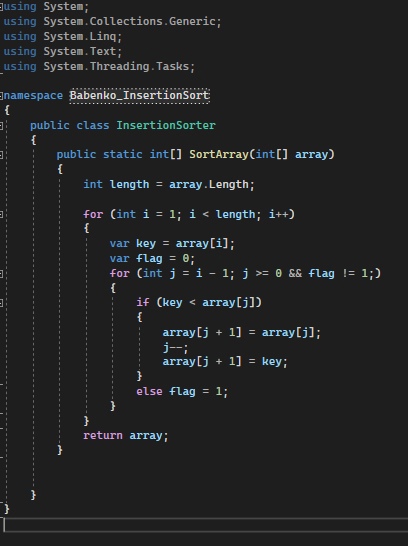
2.4 Модуль QuickSortet

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

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:

1. Необходимо разработать программный комплекс по демонстрации работы алгоритмов сортировки массивов данных (реализовать не менее 4 алгоритмов сортировки, которые выбрать самостоятельно):
   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 интерфейса программного продукта в визуальной среде.
   7. Выполнить интеграцию программных модулей в программный продукт.
   8. Разработать процедуру тестирования программного продукта. Выполнить тестирование программного продукта. Результат тестирования оформить протоколом тестирования.
   9. Разработать справочную систему программного продукта.
   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.

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

**СПИСОК ЛИТЕРАТУРЫ**

1. Котлин А.П. Алгоритмы сортировки [электронный ресурс] bimbilik.github.io – 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 – 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 – 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 – 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 – 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)

{

}

}

}