МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

|  |  |
| --- | --- |
|  | Утвержден на заседании кафедры  «Вычислительная техника»  "\_\_\_" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20 г.  Заведующий кафедрой                                              М.А. Митрохин |
|  |  |

**ОТЧЕТ ПО УЧЕБНОЙ (ОЗНАКОМИТЕЛЬНОЙ) ПРАКТИКЕ**

(2023/2024 учебный год)

                                          Панькин Роман Александрович

Направление подготовки 09.03.01 «Информатика и вычислительная техника»

Наименование профиля подготовки «Прикладной искусственный интеллект»

Форма обучения – очная Срок обучения в соответствии с ФГОС – 4 года

Год обучения1семестр 2

Период прохождения практики с 25.06.24 по 08.07.24

Кафедра «Вычислительная техника»

Заведующий кафедрой д.т.н., профессор, Митрохин М.А.

*(должность, ученая степень, ученое звание, Ф.И.О.)*

Руководитель практики к.т.н., доцент, Карамышева Н.С.

*(должность, ученая степень, ученое звание)*

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

|  |  |
| --- | --- |
|  | Утвержден на заседании кафедры  «Вычислительная техника»  "\_\_\_" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20 г.  Заведующий кафедрой                                              М.А. Митрохин |
|  |  |

**ИНДИВИДУАЛЬНЫЙ ПЛАН ПРОХОЖДЕНИЯ УЧЕБНОЙ (ОЗНАКОМИТЕЛЬНОЙ) ПРАКТИКИ**

(2023/2024 учебный год)

                                            Панькин Роман Александрович

Направление подготовки 09.03.01 «Информатика и вычислительная техника»

Наименование профиля подготовки «Прикладной искусственный интеллект»

Форма обучения – очная Срок обучения в соответствии с ФГОС – 4 года

Год обучения                  1                 семестр                 2

Период прохождения практики с 25.06.24 по 08.07.24

Кафедра «Вычислительная техника»

Заведующий кафедрой д.т.н., профессор, Митрохин М.А.

*(должность, ученая степень, ученое звание, Ф.И.О.)*

Руководитель практики к.т.н., доцент, Карамышева Н.С.

*(должность, ученая степень, ученое звание)*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Планируемая форма работы во время практики | Количество часов | Календарные сроки проведения работы | Подпись  руководителя  практики от вуза |
| 1 | Выбор темы и разработка индивидуального плана проведения работ | 2 | 25.06.2024 -  25.06.2024 |  |
| 2 | Подбор и изучение материала по теме работы | 15 | 26.06.2024 –  28.06.24 |  |
| 3 | Разработка алгоритма | 43 | 01.07.24 –  03.07.24 |  |
| 4 | Описание алгоритма и программы | 18 | 03.07.24 –  04.07.24 |  |
| 5 | Тестирование | 5 | 04.07.24 –  05.07.24 |  |
| 6 | Получение и анализ результатов | 10 | 05.07.24 –  08.07.24 |  |
| 7 | Оформление отчёта | 15 | 05.07.24 –  08.07.2024 |  |
|  | **Общий объём часов** | 108 |  |  |

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

**ОТЧЁТ**

**О ПРОХОЖДЕНИИ УЧЕБНОЙ (ОЗНАКОМИТЕЛЬНОЙ) ПРАКТИКИ**

(2023/2024 учебный год)

                                           Панькин Роман Александрович

Направление подготовки 09.03.01 «Информатика и вычислительная техника»

Наименование профиля подготовки «Прикладной искусственный интеллект»

Форма обучения – очная Срок обучения в соответствии с ФГОС – 4 года

Год обучения1семестр 2

Период прохождения практики с 25.06.24 по 08.07.24

Кафедра «Вычислительная техника»

Панькин Р.А. выполнял практическое задание «Сортировка методом Шелла». Осуществил работу с файлами. Создал графический интерфейс. Также создал таймер для подсчёта времени на выполнение сортировки.

Бакалавр Панькин Р.А. \_\_\_\_\_\_\_\_\_\_\_\_ "\_\_\_" \_\_\_\_\_\_\_\_\_\_\_ 2024 г.

Руководитель Карамышева Н.С. \_\_\_\_\_\_\_\_\_\_\_\_ "\_\_\_" \_\_\_\_\_\_\_\_\_\_\_ 2024 г.

практики

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

**ОТЗЫВ**

**О ПРОХОЖДЕНИИ УЧЕБНОЙ (ОЗНАКОМИТЕЛЬНОЙ) ПРАКТИКИ**

(2022/2023 учебный год)

                                           Панькин Роман Александрович

Направление подготовки 09.03.01 «Информатика и вычислительная техника»

Наименование профиля подготовки «Прикладной искусственный интеллект»

Форма обучения – очная Срок обучения в соответствии с ФГОС – 4 года

Год обучения1семестр 2

Период прохождения практики с 25.06.24 по 08.07.24

Кафедра «Вычислительная техника»

В процессе выполнения практики Панькин Р.А. решал следующие задачи: создание графического интерфейса, осуществление работы с файлами., создание таймера для подсчёта времени на выполнение сортировки.

За период выполнения практики были освоены основные понятия и реализован метод работы с файлами, создание графического интерфейса, создание таймера для подсчёта времени на выполнение сортировки. Во время выполнения работы Панькин Р. А. показал себя ответственным, добросовестным учеником, знающим свой предмет, имеющим представление о современном состоянии науки, владеющим современными общенаучными знаниями по информатике и вычислительной технике, программированию и сортировке.

За выполнение работы Панькин Р. А. заслуживает оценки «\_\_\_\_\_\_».

Руководитель практики к.т.н., доцент, Карамышева Н.С. « » 2024 г.

Содержание

[Введение 7](#_Toc170996725)

[1. Разработка и описание алгоритма 8](#_Toc170996726)

[1.1. Графический интерфейс 8](#_Toc170996727)

[1.2. Работа с файлами 10](#_Toc170996728)

[1.3. Таймер для отслеживания времени 11](#_Toc170996729)

[3. Заключение 12](#_Toc170996730)

[4. Список использованных источников 13](#_Toc170996731)

[5. Приложение 14](#_Toc170996732)

[Листинг 14](#_Toc170996733)

**Введение**

Microsoft Visual Studio представляет собой мощную программную среду для разработки приложений под операционную систему Windows. Она включает в себя редактор кода с поддержкой IntelliSense для удобства написания кода и базового рефакторинга, отладчик для исправления ошибок, редактор форм для создания пользовательских интерфейсов, веб-редактор, дизайнер классов и инструменты для работы с базами данных.

Visual Studio также предоставляет возможность расширения функциональности через сторонние плагины, которые добавляют поддержку систем контроля версий, новые инструменты для разработки на специализированных языках программирования и другие аспекты разработки ПО.

Основные компоненты включают Visual Basic.NET, Visual C++, Visual C#, Visual F#, обеспечивая разработку на различных языках семейства Microsoft. Visual Studio предоставляет высокоуровневые функциональные возможности, включая встроенный веб-сервер, поддержку множества языков, интуитивный стиль кодирования, ускоренную разработку и эффективные инструменты отладки.

1. **Разработка и описание алгоритма**

Я разработал приложение на C# с использованием Windows Forms для создания графического интерфейса. Основное окно программы (Form1) содержит несколько элементов управления, таких как кнопки.

Для записи неотсортированного массива в файл. При нажатии на соответствующую кнопку, программа генерирует случайный массив из 100 целых чисел, выбранных из диапазона от -10000 до 10000.Этот массив записывается в текстовый файл array.txt, который сохраняется в директории, где находится исполняемый файл программы. После успешной записи пользователь получает сообщение о создании файла с неотсортированным массивом.

1. Графический интерфейс

Кнопка "Создать неотсортированный массив" генерирует случайный массив целых чисел и записывает его в файл array.txt. Кнопка "Отсортировать массив" считывает неотсортированный массив из файла, сортирует его с использованием алгоритма сортировки Шелла и записывает отсортированный массив в файл sorted\_array.txt. При этом, для отслеживания времени, затраченного на сортировку, используется класс Stopwatch из пространства имен System.Diagnostics. Таймер запускается перед началом сортировки и останавливается после её завершения, позволяя пользователю увидеть время, необходимое для выполнения операции.

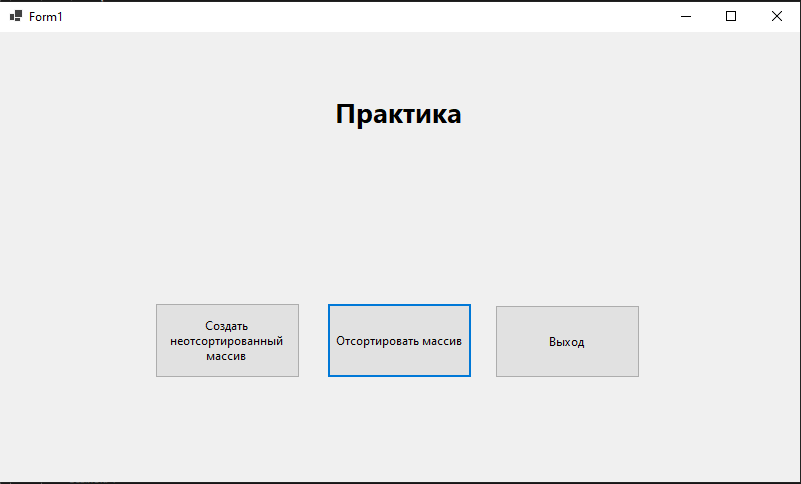


Рисунок 1 – Меню приложения

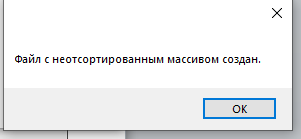


Рисунок 2 ­­– Сообщение о записи неотсортированного массива в файл

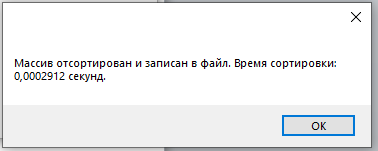


Рисунок 3 – Сообщение о записи отсортированного массива в файл и времени его сортировки.

Графический интерфейс приложения предоставляет пользователю визуальные элементы управления (кнопки) и связывает их с функциональностью через обработчики событий, позволяя пользователю взаимодействовать с программой через нажатия кнопок.

1. Работа с файлами

string filePath = "array.txt";

using (StreamWriter writer = new StreamWriter(filePath))

{

foreach (int number in array)

{

writer.WriteLine(number);

}

}

MessageBox.Show("Массив отсортирован и записан в файл.");

Неотсортированный массив записывается в текстовый файл array.txt. После завершения записи пользователь получает сообщение о том, что неотсортированный массив был сохранен в файле.

string sortedFilePath = "sorted\_array.txt";

using (StreamWriter writer = new StreamWriter(sortedFilePath))

{

foreach (int number in array)

{

writer.WriteLine(number);

}

}

Отсортированный массив записывается в текстовый файл sorted\_array.txt. После завершения записи пользователь получает сообщение о том, что массив был отсортирован и сохранен в файле.

Таймер для отслеживания времени

using System.Diagnostics;//Пространство имен System.Diagnostics для создания таймера

Stopwatch stopwatch = new Stopwatch();//Создание таймера до сортировки

stopwatch.Start();//Запуск таймера

stopwatch.Stop();//Остановка таймера после завершения сортировки

В программе присутствует возможность отслеживать время, затрачиваемое на сортировку массива, используя класс Stopwatch из пространства имен System.Diagnostics. Для этого в методе button1\_Click\_1 создается и запускается таймер перед началом сортировки массива. После завершения сортировки таймер останавливается, и время выполнения сортировки отображается пользователю в сообщении. Это позволяет точно измерить и вывести время, которое потребовалось для выполнения сортировки массива методом Шелла. Таким образом, пользователь может видеть, сколько времени заняла сортировка, что добавляет полезную функциональность для анализа производительности алгоритма сортировки.

1. **Совместная работа**

Во время работы над данной практикой, нашей бригадой осуществлялась совместная работа в GitHub.

Данная программа была написана Прохоровым Д.С и владельцем репозитория Панькиным Р.А. Молев П.О провёл тестирование данной программы

После написания алгоритма сортировки программы Прохоровым Д.С, она была выгружена на удаленный репозиторий Github .Сделал комит, и выгрузил обновленный код программы на удаленный репозиторий GitHub.

После этого, второй участник – Панькин Р.А. загрузил данную программу себе на компьютер, с помощью git clone <https://github.com/Eliksirowich/practice1.git> , и добавил в нее алгоритм, считающий время работы программы в секундах ,графический интерфейс и работу с файлами. Сделал комит, и выгрузил обновленный код программы на удаленный репозиторий GitHub.

Далее, третий участник – Молев П.О загрузил программу себе на компьютер, с помощью git clone <ссылка>, и провёл тестировку на наличие ошибок в программе.

Ссылка на репозиторий: https://github.com/Eliksirowich/practice1.git

**Заключение**

В ходе разработки данного приложения на C# с использованием Windows Forms я создал графический интерфейс, который позволяет пользователю взаимодействовать с программой. Основное окно программы содержит несколько кнопок, каждая из которых выполняет определенную функцию.

Графический интерфейс достаточно понятен и помогает пользователю , посредствам нажатия на кнопки, выполнять операции, заложенные в коде программы.

Добавление таймера для измерения времени позволяет пользователю оценить производительность алгоритма сортировки и понять, сколько времени занимает обработка данных.

Приложение демонстрирует основные возможности разработки на C# с использованием Windows Forms, включая создание и управление графическим интерфейсом, работу с файлами и массивами данных, а также использование таймеров для измерения производительности сортировки.

**Список использованных источников**

1. "Programming Windows Forms in C#"( Чарльз Петцольд)
2. "C# 8.0 and .NET Core 3.0 – Modern Cross-Platform Development"( Марк Дж. Прайс)
3. "Introduction to Algorithms"( Томас Х. Кормен, Чарльз Э. Лейзерсон, Рональд Л. Ривест, Клиффорд Штейн)
4. "The C# Programming Yellow Book"( Роб Майлз)
5. "Pro C# 8 with .NET Core: Foundational Principles and Practices in Programming"(Эндрю Троелсен, Филип Япиксе)
6. "Data Structures and Algorithms in C#"( Роберт Лафоре)
7. "Windows Forms Programming in C#"( Крис Селлс)
8. "C# in Depth"( Джон Скит)
9. "Algorithms in C#, Parts 1-4: Fundamentals, Data Structures, Sorting, Searching"(Роберт Седжвик)
10. "CLR via C#"(Джеффри Рихтер)
11. **Приложение**

## Листинг А:

**Sort.cs:**

sing System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace practice

{

public partial class Form1

{

//метод для обмена элементов

static void Swap(ref int a, ref int b)

{

(a, b) = (b, a);

}

static int[] ShellSort(int[] array)

{

//расстояние между элементами, которые сравниваются

var d = array.Length / 2;

while (d >= 1)

{

for (var i = d; i < array.Length; i++)

{

var j = i;

while ((j >= d) && (array[j - d] > array[j]))

{

Swap(ref array[j], ref array[j - d]);

j = j - d;

}

}

d = d / 2;

}

return array;

}

}

}

## Листинг Б:

**Form1.cs:**

using System;

using System.Diagnostics;

using System.IO;

using System.Windows.Forms;

using System.Diagnostics;

namespace practice

{

public partial class Form1 : Form

{

int a = 100000;

int b = -1001;

int c = 1001;

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

}

private void button2\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void button3\_Click(object sender, EventArgs e)

{

Random random = new Random();

int[] array = new int[a];

for (int i = 0; i < array.Length; i++)

{

array[i] = random.Next(b+1, c+1);

}

string filePath = "array.txt";

using (StreamWriter writer = new StreamWriter(filePath))

{

foreach (int number in array)

{

writer.WriteLine(number);

}

}

MessageBox.Show("Ôàéë ñ íåîòñîðòèðîâàííûì ìàññèâîì ñîçäàí.");

}

private void button1\_Click\_1(object sender, EventArgs e)

{

string filePath = "array.txt";

if (!File.Exists(filePath))

{

MessageBox.Show("Ôàéë ñ ìàññèâîì íå íàéäåí.");

return;

}

string[] lines = File.ReadAllLines(filePath);

int[] array = new int[lines.Length];

for (int i = 0; i < lines.Length; i++)

{

array[i] = int.Parse(lines[i]);

}

Stopwatch stopwatch = new Stopwatch();

stopwatch.Start();

array = ShellSort(array);

stopwatch.Stop();

string sortedFilePath = "sorted\_array.txt";

using (StreamWriter writer = new StreamWriter(sortedFilePath))

{

foreach (int number in array)

{

writer.WriteLine(number);

}

}

MessageBox.Show($"Ìàññèâ îòñîðòèðîâàí è çàïèñàí â ôàéë. Âðåìÿ ñîðòèðîâêè: {stopwatch.Elapsed.TotalSeconds} ñåêóíä.");

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

try

{

a = Convert.ToInt32(textBox1.Text);

}

catch (FormatException)

{

MessageBox.Show("Íåêîððåêòíûé ââîä, ââåäèòå ÷èñëî.");

}

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

try

{

b = Convert.ToInt32(textBox2.Text);

}

catch (FormatException)

{

if (textBox2.Text != "-")

{

MessageBox.Show("Íåêîððåêòíûé ââîä, ââåäèòå ÷èñëî.");

}

}

}

private void textBox3\_TextChanged(object sender, EventArgs e)

{

try

{

c = Convert.ToInt32(textBox3.Text);

}

catch (FormatException)

{

if (textBox3.Text != "-")

{

MessageBox.Show("Íåêîððåêòíûé ââîä, ââåäèòå ÷èñëî.");

}

}

}

}

}