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 (ЭКСПЛУАТАЦИОННОЙ) ПРАКТИКЕ**

(2023/2024 учебный год)

                                         Панькин Роман Александрович

Направление подготовки 09.03.01 «Информатика и вычислительная техника»

Наименование профиля подготовки «Вычислительные машины, комплексы, системы и сети»

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

Год обучения1семестр 2

Период прохождения практики с 20.06.2024 по 17.07.2024

Кафедра «Вычислительная техника»

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

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

Руководитель практики д.т.н., профессор, Зинкин С.А.

*(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 (ЭКСПЛУАТАЦИОННОЙ) ПРАКТИКИ**

(2023/2024 учебный год)

                                          Панькин Роман Александрович

Направление подготовки 09.03.01 «Информатика и вычислительная техника»

Наименование профиля подготовки «Вычислительные машины, комплексы, системы и сети»

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

Год обучения                  1                семестр                 2

Период прохождения практики с 20.06.2024 по 17.07.2024

Кафедра «Вычислительная техника»

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

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

Руководитель практики д.т.н., профессор, Зинкин С.А.

*(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 | 26 | 20.06.24 –  24.06.24 |  |
| 2 | Подбор и изучение материала по теме работы | 26 | 24.06.24 –  26.06.24 |  |
| 3 | Установка виртуальной машины | 26 | 26.06.24 –  28.06.24 |  |
| 4 | Установка операционной системы | 30 | 01.07.24 –  04.07.24 |  |
| 5 | Разработка программы на языке Си | 30 | 04.07.24 –  09.07.24 |  |
| 6 | Тестирование и отладка | 38 | 09.07.24 –  12.07.24 |  |
| 7 | Оформление отчёта | 40 | 12.07.24 –  17.07.24 |  |
|  | **Общий объём часов** | 216 |  |  |

**1 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.

**2 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.

**Команды отвечающие за создание массива:**

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.");

Для сортировки массива и записи отсортированного массива в файлПри нажатии на другую кнопку, программа считывает неотсортированный массив из файла ”array.txt”.Считанный массив сортируется с использованием алгоритма сортировки Шелла.

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.Таким образом, пользователь может управлять программой через простой и понятный графический интерфейс, выполняя операции с массивами данных с минимальными усилиями.

**Команды отвечающие за сортировку массива:**

string sortedFilePath = "sorted\_array.txt";

using (StreamWriter writer = new StreamWriter(sortedFilePath))

{

foreach (int number in array)

{

writer.WriteLine(number);

}

}  
  
 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.

**Команды отвечающие за таймер:**

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

Кроме этого я добавили три кнопки :”Создать неотсортированный массив”,”Отсортировать массив” и “Выход”. Каждая кнопка имеет привязанный метод-обработчик событий, который вызывается при нажатии на кнопку. ”Отсортировать массив” предназначена для запуска сортировки массива. В методе button1\_Click\_1 считывается массив из файла, сортируется методом Шелла, а затем записывается в новый файл. “Выход” используется для выхода из приложения. При нажатии на эту кнопку вызывается метод Application.Exit(), который завершает работу приложения. :”Создать неотсортированный массив” генерирует неотсортированный массив случайных чисел и записывает его в файл. При нажатии на кнопку вызывается метод button3\_Click, в котором создается массив случайных чисел, записывается в файл и выводится сообщение об успешном завершении операции.

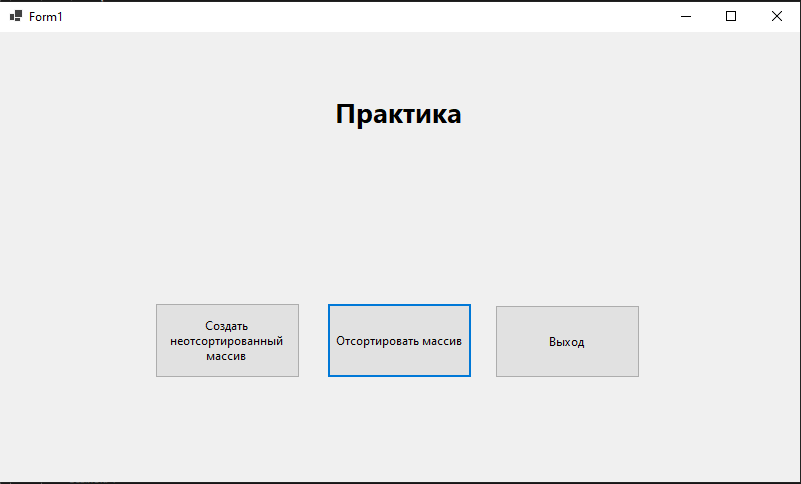


Рисунок 1

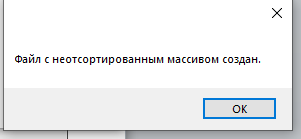


Рисунок 2

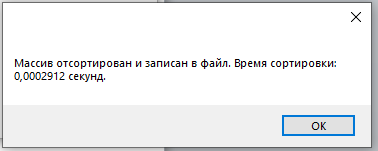


Рисунок 3

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.

**3.Вывод**

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

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.

Графический интерфейс приложения предоставляет пользователю удобные визуальные элементы управления, такие как кнопки, и связывает их с функциональностью через обработчики событий. Это позволяет пользователю выполнять различные операции с массивами данных с минимальными усилиями, улучшая пользовательский опыт и делая работу с программой более продуктивной.

Добавление таймера для измерения времени сортировки является важным улучшением, которое позволяет пользователю оценить производительность алгоритма сортировки и понять, сколько времени занимает обработка данных. Это добавляет полезную функциональность для анализа и оптимизации работы программы.

В целом, 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.