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**

(2022/2023 учебный год)

                                          Николаев Александр Алексеевич

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 29.06.2023 по 12.07.2023

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**

(2022/2023 учебный год)

                                           Николаев Александр Алексеевич

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 29.06.2023 по 12.07.2023

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 | 29.06.2023 -  29.06.2023 |  |
| 2 | Подбор и изучение материала по теме работы | 15 | 30.06.2023 –  02.07.23 |  |
| 3 | Разработка алгоритма | 43 | 02.07.23 –  06.07.23 |  |
| 4 | Описание алгоритма и программы | 18 | 6.07.23 –  08.07.23 |  |
| 5 | Тестирование | 5 | 08.07.23–  08.07.23 |  |
| 6 | Получение и анализ результатов | 10 | 08.07.23 –  10.07.23 |  |
| 7 | Оформление отчёта | 15 | 10.07.23 –  12.07.2023 |  |
|  | **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**

(2022/2023 учебный год)

                                        Николаев Александр Алексеевич

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 29.06.2023 по 12.07.2023

Кафедра «Вычислительная техника»

Николаев А.А. выполнял практическое задание «Сортировка пузырьком». На первоначальном этапе был изучен и проанализирован алгоритм пузырьковой сортировки, был выбран метод решения и язык программирования С, на котором была написана программа сортировки массива методом пузырька. Также осуществил работу с файлами, сделал пользовательский интерфейс. Оформил отчёт.

Бакалавр Николаев А.А. \_\_\_\_\_\_\_\_\_\_\_\_ "\_\_\_" \_\_\_\_\_\_\_\_\_\_\_ 2023 г.

Руководитель Зинкин С.А. \_\_\_\_\_\_\_\_\_\_\_\_ "\_\_\_" \_\_\_\_\_\_\_\_\_\_\_ 2023 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)

                                             Николаев Александр Алексеевич

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 29.06.2023 по 12.07.2023

Кафедра «Вычислительная техника»

В процессе выполнения практики Николаев А.А. решал следующие задачи: анализ алгоритма пузырьковой сортировки и его работы, осуществление работы с файлом, разработка пользовательского интерфейса, сравнение существующих методов сортировки.

За период выполнения практики были освоены основные понятия и технологии пузырьковой сортировки, реализован метод работы с файлами. Во время выполнения работы Николаев А.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 Николаев А.А. заслуживает оценки «\_\_\_\_\_\_\_\_\_\_».

Руководитель практики д.т.н., профессор, Зинкин С.А. « » 2023г.

**Введение**

В современном мире информационные технологии стали неотъемлемой частью жизни практически любого человека.

Снижение трудоёмкости любых производственных, ускорение социальных процессов и, как следствие, повышение их эффективности сейчас связано с существенно изменившимися за последние десятилетия возможностями сбора, обработки, хранения, передачи и представления информации.

Реализация полученной в результате этих процессов информации совершенно нового качества позволяет обеспечить процесс создания и использования материальных и духовных ценностей на более высоком и принципиально новом уровне.

Таким образом изучение современных информационных технологий обеспечивает возможность человеку без проблем ориентироваться в современных тенденциях и путях развития общества, процессах производства и повышать качество социальной жизни.

Изучение всегда начинается с освоения базовых элементов, таких как методы сортировки, хранения и передачи информации. За период выполнения практики мною были освоены основные понятия и технологии пузырьковой сортировки, реализован метод работы с файлами.

В качестве инструмента для реализации был выбран Microsoft Visual Studio и язык программирования С.

Microsoft Visual Studio — это программная среда по разработке приложений для ОС Windows, как консольных, так и с графическим интерфейсом. Она является стартовой площадкой для написания, отладки и сборки кода, а также последующей публикации. Помимо стандартных редактора и отладчика, Visual Studio включает в себя компиляторы, средства авто завершения кода, графические конструкторы и многие другие элементы и функции, позволяющие упростить процесс разработки, что делает среду Visual Studio одной из самых простых и понятных в освоении и использовании.

Язык С является универсальным языком программирования, который сам по себе не связан с какой-либо операционной системой и с успехом использовался и используется для написания больших вычислительных программ и программ обработки текстов и баз данных. Его также называют языком системного программирования, так как он удобен при написании операционных систем. С является языком относительно низкого уровня, поскольку имеет дело с объектами того же вида, что и большинство ЭВМ, а именно с символами, числами и адресами. Они могут объединяться и пересылаться посредством обычных арифметических и логических операций, осуществляемых реальными ЭВМ. Поэтому язык С крайне удобен для разработки различных приложений и реализации алгоритмов, которые, например, могут использоваться для сортировки данных.

**1 Постановка задачи**

**1.1 Достоинства алгоритма**

Метод сортировки «пузырьком» является наиболее простым и понятным в реализации. Он легко реализуется на всех языках программирования и подходит для начинающих. «Пузырёк» также лежит в основе других более сложных методов сортировки.

**1.2 Недостатки алгоритма**

Алгоритм считается учебным и практически не применяется вне учебной литературы, а также не эффективен для больших массивов. Такая ситуация наблюдается из-за большого количества проходов и сравнений на каждом из проходов алгоритма, на что тратится большое количество времени. Время выполнения алгоритма прямо пропорционально квадрату количества элементов (N2), что делает сортировку «пузырьком» самым медленным методом сортировки.

**1.3 Типичные сценарии использования**

Данный метод сортировки актуален для использования в учебных целях(для реализации начинающим программистом), а также для небольшого массива входных данных. Алгоритм можно применять для сортировки с целью ускорения поиска каких-либо элементов и для упорядочивания элементов.

**2 Выбор решения**

В качестве среды разработки программы и реализации алгоритма была выбрана программа Microsoft Visual Studio 2022, а в качестве языка программирования был выбран С. Для отладки программы использовались несколько возможностей данной среды, а именно трассировка, точка останова, просмотр значений переменных.

Все действия, совершаемые программой, происходят в одном файле.

С помощью пользовательского интерфейса пользователь может выбрать действие, которое он хочет осуществить.

**3 Описание программы**

В данной работе студент Николаев А.А. выполнял практическое задание «Сортировка пузырьком». В процессе разработки студент осуществил запись данных в файл и разработал пользовательский интерфейс.

Запись данных в файл осуществляется в функции *writeArrayToFile:*

void writeArrayToFile(int arr[], int size) {

printf("Введите имя файла для загрузки данных в него: ");

scanf("%s", n\_file);

if (sizeof(n\_file) > 49) {

printf("Имя файла превышает доступное значение (49)!\n");

return;

}

if ((file = fopen(n\_file, "w")) == NULL)

{

printf("\nНевозможно открыть для записи файл: %s\n", n\_file);

\_getch();

return;

}

fprintf(file, "%d\n", size);

for (int i = 0; i < size; i++)

fprintf(file, "%d ", arr[i]);

fclose(file);

printf("Данные успешно записаны!\n");

}

Подробный алгоритм выполнения этой функции представлен в пункте 4(рисунок 1).

В функции *main* был реализован пользовательский интерфейс:

void main() {

setlocale(LC\_ALL, "RUS");

srand(time(NULL));

helloScreen();

clock\_t start, end;

double time\_spent = 0.0;

system("cls");

do {

printf("\tМеню\n");

printf("1. Просмотреть массив\n");

printf("2. Отсортировать массив\n");

printf("3. Записать массив в файл\n");

printf("4. Загрузить массив из файла\n");

printf("5. Посмотреть время выполнения алгоритма\n");

printf("6. Создать новый массив\n");

printf("7. Выход\n");

printf("Выберите действие: ");

scanf("%d", &choice);

switch (choice) {

case 1: //Просмотр

system("cls");

printf("Массив:\n ");

for (int i = 0; i < size; i++) {

printf("%d) %d\n ", i+1, arr[i]);

}

printf("\n");

break;

case 2: // Сортировка

system("cls");

start = clock();

bubbleSort(arr, size);

end = clock();

printf("Массив отсортирован!\n");

time\_spent = (double)(end - start) / CLOCKS\_PER\_SEC;

printf("\n");

break;

case 3: // Запись в файл

system("cls");

writeArrayToFile(arr, size);

break;

case 4: // Запись в файл

system("cls");

if (arr != NULL)

{

free(arr);

}

readArrayFromFile();

break;

case 5: // Время

system("cls");

if (time\_spent == 0) {

printf("Сначала отсортируйте массив!\n");

break;

}

printf("Время выполнения алгоритма: %f\n", time\_spent);

break;

case 6: //Ввод данных

system("cls");

free(arr);

printf("Введите размер массива: ");

scanf("%d", &size);

arr = (int\*)malloc(size \* sizeof(int));

if (arr == NULL) {

printf("Не удалось выделить память!\n");

return;

}

for (int i = 0; i < size; i++) {

arr[i] = rand();

}

break;

case 7: // Выход

system("cls");

printf("До свидания!\n");

free(arr);

break;

default:

system("cls");

printf("Некорректный выбор!\n");

break;

}

} while (choice != 7);

getchar();

}

Самой первой вызывается функция *helloScreen*. Это функция начала программы, она выводит в консоль заставку программы.

Дальнейший алгоритм выполнения этой функции будет представлен в пункте 4(рисунок 2).

**4. Схема алгоритма**

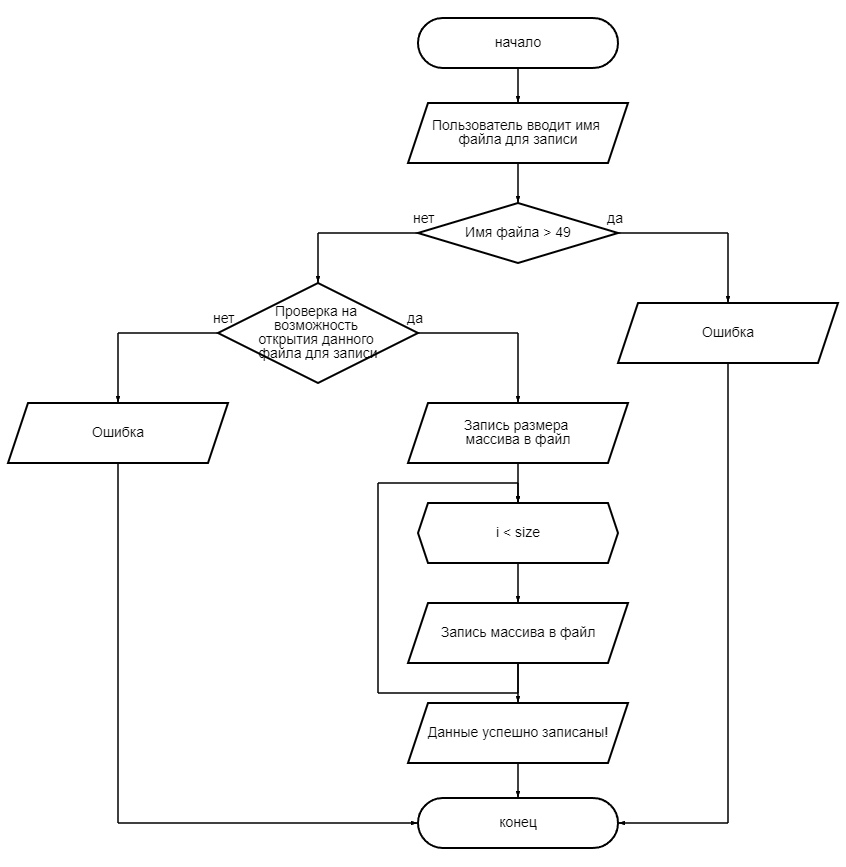


Рисунок 1 - блок-схема функции *writeArrayToFile*

**5. Отладка**

В качестве среды разработки была выбрана программа Microsoft Visual

Studio , которая содержит в себе все необходимые средства для разработки и

отладки модулей и программ.

Для отладки программы использовались точки остановки и пошаговое

выполнение кода программы, анализ содержимого глобальных и локальных

переменных.

Тестирование проводилось в рабочем порядке, в процессе разработки,

после завершения написания программы. После завершения написания

программы, человеком, выполнявшим тестирование программы, были

выявлены и исправлены ошибки.

**6. Совместная разработка**

**Заключение**

За период выполнения данной учебной практической работы были освоены основные понятия и технологии пузырьковой сортировки, реализован метод работы с файлами. Продемонстрирована работа алгоритма сортировки «пузырьком». Конечная программа реализовывает непосредственно алгоритм, а также выполняет считывание входных данных из файла и запись результатов в файл. В ходе тестирование была произведена отладка, алгоритм выполняется верно, предусмотрены ограничения формата ввода чисел и ситуация ошибки чтения файлов.

В результате тестирования мы пришли к выводу, что данный алгоритм подходит для сортировки небольших массивов, так как при увеличении числа элементов, подвергающихся сортировке, в геометрической прогрессии растёт время выполнения программы. Сортировка «пузырьком» является довольно простым и эффективным алгоритмом в случае, если применять его при работе с небольшими последовательностями чисел, либо с частично отсортированными последовательностями, либо в ознакомительных и учебных целях. Реализация алгоритма проста, однако высокая вычислительная сложность и большие временные затраты на реализацию сортировки сильно ограничивают сферу применения данного метода.

**Список источников**

**Приложение А**

**Файл main.c**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <conio.h>

#include <locale.h>

#include <stdlib.h>

#include <time.h>

int size, choice;

int\* arr = 0;

char n\_file[49];

FILE\* file;

void bubbleSort(int arr[], int size) {

int k = 0;

for (int i = 0; i < size - 1; i++) {

for (int j = 0; j < size - i - 1; j++) {

if (arr[j] > arr[j + 1]) {

int temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

k++;

}

if (k == 0)

break;

}

}

}

void writeArrayToFile(int arr[], int size) {

printf("Введите имя файла для загрузки данных в него: ");

scanf("%s", n\_file);

if (sizeof(n\_file) > 49) {

printf("Имя файла превышает доступное значение (49)!\n");

return;

}

if ((file = fopen(n\_file, "w")) == NULL)

{

printf("\nНевозможно открыть для записи файл: %s\n", n\_file);

\_getch();

return;

}

fprintf(file, "%d\n", size);

for (int i = 0; i < size; i++)

fprintf(file, "%d ", arr[i]);

fclose(file);

printf("Данные успешно записаны!\n");

}

void readArrayFromFile() {

printf("Введите имя файла для загрузки данных из него: ");

scanf("%s", n\_file);

if (sizeof(n\_file) > 49) {

printf("Имя файла превышает доступное значение (49)!");

return;

}

if ((file = fopen(n\_file, "r+")) == 0)

{

printf("Невозможно открыть файл для чтения %s\n", n\_file);

\_getch();

return;

}

fscanf(file, "%d", &size);

arr = (int\*)malloc(size \* sizeof(int));

for (int i = 0; i < size; i++) {

fscanf(file, "%d", &arr[i]);

}

fclose(file);

printf("Данные успешно прочитаны!\n");

}

void helloScreen() {

printf("\tПрактическая работа\nНа тему: \"Сортировка пузырьком (bubble sort)\"\n\nВыполнили студенты группы 22ВВВ1: Курушин Ярослав, Николаев Александр, Ганин Иван.\n\n");

system("PAUSE");

}

void main() {

setlocale(LC\_ALL, "RUS");

srand(time(NULL));

helloScreen();

clock\_t start, end;

double time\_spent = 0.0;

system("cls");

do {

printf("\tМеню\n");

printf("1. Просмотреть массив\n");

printf("2. Отсортировать массив\n");

printf("3. Записать массив в файл\n");

printf("4. Загрузить массив из файла\n");

printf("5. Посмотреть время выполнения алгоритма\n");

printf("6. Создать новый массив\n");

printf("7. Выход\n");

printf("Выберите действие: ");

scanf("%d", &choice);

switch (choice) {

case 1: //Просмотр

system("cls");

printf("Массив:\n ");

for (int i = 0; i < size; i++) {

printf("%d) %d\n ", i+1, arr[i]);

}

printf("\n");

break;

case 2: // Сортировка

system("cls");

start = clock();

bubbleSort(arr, size);

end = clock();

printf("Массив отсортирован!\n");

time\_spent = (double)(end - start) / CLOCKS\_PER\_SEC;

printf("\n");

break;

case 3: // Запись в файл

system("cls");

writeArrayToFile(arr, size);

break;

case 4: // Запись в файл

system("cls");

if (arr != NULL)

{

free(arr);

}

readArrayFromFile();

break;

case 5: // Время

system("cls");

if (time\_spent == 0) {

printf("Сначала отсортируйте массив!\n");

break;

}

printf("Время выполнения алгоритма: %f\n", time\_spent);

break;

case 6: //Ввод данных

system("cls");

free(arr);

printf("Введите размер массива: ");

scanf("%d", &size);

arr = (int\*)malloc(size \* sizeof(int));

if (arr == NULL) {

printf("Не удалось выделить память!\n");

return;

}

for (int i = 0; i < size; i++) {

arr[i] = rand();

}

break;

case 7: // Выход

system("cls");

printf("До свидания!\n");

free(arr);

break;

default:

system("cls");

printf("Некорректный выбор!\n");

break;

}

} while (choice != 7);

getchar();

}