Пермский филиал федерального государственного автономного образовательного учреждения высшего образования  
«Национальный исследовательский университет   
Высшая школа экономики»

*Факультет социально-экономических и компьютерных наук*

Мусихин Данил Михайлович

**Работа с файлами**

*Лабораторная работа №4*

студента образовательной программы «Разработка информационных систем для бизнеса» по направлению подготовки *09.03.04 Программная инженерия*

Руководитель, к.т.н., Доцент кафедры ИТБ.

Л.Н. Лядова

Пермь, 2023 год

**Оглавление**

[1 Постановка задачи 4](#_Toc152166223)

[2 Анализ 5](#_Toc152166224)

[3 Код 9](#_Toc152166241)

# Постановка задачи

Цель работы – изучение возможностей работы с файлами.

**Рекомендации по выполнению**: при выполнении заданий используйте разделы ([***https://learn.microsoft.com/ru-ru/cpp/standard-library/iostream-programming?view=msvc-170***](https://learn.microsoft.com/ru-ru/cpp/standard-library/iostream-programming?view=msvc-170)) – подразделы, посвящённые работе с файлами – справочной системы (или файл PDF с полным руководством, доступные по ссылке: [***https://learn.microsoft.com/ru-ru/cpp/standard-library/***](https://learn.microsoft.com/ru-ru/cpp/standard-library/)).

При выполнении работы необходимо рассмотреть возможности:

1)      создания файлов;

2)      чтения из файлов;

3)      записи в файлы.

Разработайте консольное приложение, в котором должны быть выполнены следующие задачи:

1 – Создание файла (в формате .txt) и запись в файл одного значения (строки например: «В файл выведена информация»). После записи файл нужно закрыть. Проверьте, создан ли файл, записана ли в него строка. Какие ошибки могут произойти при создании файла? Как их перехватить?

2 – Открытие созданного файла для чтения и считывание записанной строки. Считанная из файла строка должна быть выведена на экран. После выполнения операции файл должен быть закрыт. Какие ошибки могут произойти при чтении из файла? Как их перехватить?

3 – Открытие файла для записи и вывод в файл последовательности строк, которые пользователь вводит с клавиатуры (строки могут содержать целые числа и пр.). После ввода последней строки файл закрывается. Как вывести в существующий файл новые строки, добавив их в конец? Как очистить файл перед записью? Рассмотрите оба варианта. Какие ошибки могут произойти? Как их перехватить?

4 – Открытие файла для чтения и считывание всех записанных в него значений. Считанные из файла значения должны быть занесены в линейный список в порядке считывания и выведены на экран. После выполнения операции файл должен быть закрыт. Какие ошибки могут произойти? Как их перехватить? Рассмотрите варианты, когда файл содержит только введённые числа (см. выше) без первой строки (после очистки), и когда в файл записана первая строка (в примере – «В файл выведена информация»), которую нужно пропустить при считывании значений чисел и т. п. (см. выше).

Выполнение заданий 1–4 необходимо для получения оценки «хорошо».

Для получения оценки «отлично» рассмотрите следующие вопросы:

5 – Есть ли возможность при вводе данных из файла получить прямой доступ к нужному значению? Поясните ответ. Приведите пример кода с решением задачи (смещения в нужному значению в файле).

6 – Есть ли возможность записывать в файлы и считывать из файлов структурированную информацию (записей с заданным пользователем набором атрибутов)? Поясните ответ. Приведите пример кода с решением задачи вывода в файл значений атрибутов записей и считывания этих значений.

7 – Какие возможности имеются для управления доступом к файлам в многопрограммной среде? Поясните ответ, приведите примеры.

# Анализ

## Создание файла и запись в файл:

Используемые функции: ofstream, open(), is\_open(), close().

Потенциальные ошибки: Ошибка доступа, недостаток прав, отсутствие свободного места.

Обработка ошибок: Проверка открытия файла с использованием is\_open(), обработка исключений.

## Открытие файла для чтения и считывание строки:

Используемые функции: ifstream, open(), is\_open(), getline(), close().

Потенциальные ошибки: Отсутствие файла, недостаток прав на чтение.

Обработка ошибок: Проверка открытия файла с использованием is\_open(), обработка исключений.

## Открытие файла для записи и вывод последовательности строк:

Используемые функции: ofstream, open(), is\_open(), getline(), close().

Добавление строк в конец файла: использование флага ios::app.

Очистка файла перед записью: использование флага ios::trunc.

Потенциальные ошибки: Ошибка доступа, недостаток прав, отсутствие свободного места.

Обработка ошибок: Проверка открытия файла с использованием is\_open(), обработка исключений.

## Чтение из файла в линейный список:

Используемые функции: ifstream, open(), is\_open(), getline(), stoi(), list.

Потенциальные ошибки: Отсутствие файла, недостаток прав на чтение, некорректные данные в файле.

Обработка ошибок: Проверка открытия файла с использованием is\_open(), обработка исключений.

Дополнительные вопросы для оценки "отлично":

## Прямой доступ к значению в файле:

Используемые функции: seekg().

Пример кода: Установка позиции с помощью seekg() и чтение данных.

## Структурированная информация в файлах:

Используемые структуры данных: struct или классы.

Пример кода: Определение структуры, запись и чтение объектов.

## Управление доступом в многопрограммной среде:

Используемые функции/методы: Системные вызовы, семафоры, мьютексы.

Примеры: Обеспечение синхронизации доступа к файлу между несколькими процессами.

# Код

#include <iostream>

#include <fstream>

#include <string>

#include <vector>

#include <filesystem>

using namespace std;

void WriteFile(string file\_name)

{

ofstream out(file\_name);

if (out.is\_open())

out << "Hello" << endl;

out.close();

}

void ReadFile(string file\_name)

{

ifstream in(file\_name);

string s;

if (in.is\_open())

while (getline(in, s))

cout << s << endl;

in.close();

}

void UpdateFile(string file\_name)

{

ofstream out(file\_name, ios::app);

string s;

if (out.is\_open())

do {

cout << s;

out << s << endl;

} while (s.empty());

out.close();

}

void GetValueFromFile(int index, string file\_name)

{

}

struct Point

{

Point(double x, double y) : x{ x }, y{ y } {}

double x;

double y;

};

void WriteStructuredDataInFile(string file\_name)

{

vector<Point> points{ Point{0, 0}, Point{4, 5}, Point{-5, 7} };

ofstream out(file\_name);

if (out.is\_open())

{

// записываем все объекты Point в файл

for (const Point& point : points)

{

out.write((char\*)&point, sizeof(point));

}

}

out.close();

vector<Point> new\_points(3);

ifstream in(file\_name); // окрываем файл для чтения

if (in.is\_open())

{

in.read((char\*)&new\_points[0], sizeof(points[2]));

in.read((char\*)&new\_points[1], sizeof(points[1]));

in.read((char\*)&new\_points[2], sizeof(points[0]));

}

in.close();

for (const Point& point : new\_points)

{

cout << "Point X: " << point.x << "\tY: " << point.y << endl;

}

}

void ChangeLaws(string file\_name)

{

// Изменение прав доступа на редактирование для владельца

filesystem::permissions(file\_name,

filesystem::perms::owner\_write,

filesystem::perm\_options::add);

cout << "Права доступа изменены" << endl;

}

int main()

{

cout << "Program started" << endl;

string file\_name = "hello.txt";

ReadFile(file\_name);

// UpdateFile(file\_name);

// ReadFile(file\_name);

}