Министерство образования и науки РФ

Севастопольский государственный университет

Кафедра информатики и управления в технических системах

ОТЧЕТ

по лабораторной работе №8

ИССЛЕДОВАНИЕ средств управления потоками ввода-вывода. Исследование механизма обработки исключений.

по дисциплине «Объектно-ориентированное программирование»

Выполнил:

Студент группы ИВТ/б 22-о

Черняев Н.Г.

Проверил:

Тимофеев И.С.

г. Севастополь 2018

**Цель работы:**

Изучить способы реализации и особенности управления потоками ввода/вывода, исследовать способы генерации и обработки исключений.

**Вариант №10**

1. Написать программу, печатающую символы от “a” до введенного с клавиатуры символа (последний возможный “z”). Для каждого символа вывести номер, сам символ, шестнадцатеричный, восьмеричный и двоичный код этого символа. Сохранить результат работы программы в файле. Предусмотреть обработку ошибок.

2. Создать класс МойФайл, содержащий строку – путь к файлу. Описать перехват ошибок, связанных с некорректной работой с файлом. Пример: попытка открыть файл, не существующий по данному пути.

**Текст программы:**

#include "stdafx.h"

#include <string>

#include <iostream>

#include <fstream>

#include <stdio.h>

#include <iomanip>

#include <stdlib.h>

using namespace std;

class MyFile

{

private:

string path;

public:

MyFile()

{

path = "";

}

void InputPath()

{

cout << "Enter path to file" << endl;

cin >> path;

}

string GetPath()

{

return path;

};

};

string inBinary(int num)

{

int copy, a = 0;

string str, result;

while (num != 0)

{

copy = num;

copy = copy / 2;

a = num % 2;

num = copy;

str += to\_string(a);

}

for (int i = str.length() - 1;i >= 0;i--)

{

result += str[i];

}

return result;

}

int main()

{

string path;

fstream file;

MyFile ob;

int i = 1;

char start\_symb;

char in\_symb;

ob.InputPath();

path = ob.GetPath();

file.exceptions(fstream::badbit | fstream::failbit);

try

{

cout << "Trying to open the file" << endl;

file.open(path, ios::in | ios::out | ios::trunc);

cout << "File opened successfully" << endl;

cout << "Enter the Symbol from a to z: " << endl;

cin >> in\_symb;

if (in\_symb<'a' || in\_symb>'z')

{

throw exception("Invalid character entered");

}

else

{

for (start\_symb = 'a'; start\_symb <= in\_symb; start\_symb++) {

file << i << " - " << start\_symb << "\t 10:" << dec << int(start\_symb);

file << "\t 16:" << hex << int(start\_symb);

file << "\t 8:" << oct << int(start\_symb);

file << "\t 2:" << inBinary(int(start\_symb)) << endl;

i++;

}

file.close();

cout << "Data is located in the file" << endl;

}

}

catch (const fstream::failure & ex)

{

cout << ex.what() << endl;

cout << ex.code() << endl;

cout << "Error in opening file" << endl;

}

catch (const exception &ex)

{

cout << ex.what() << endl;

}

return 0;

}

**Тестовые примеры:**

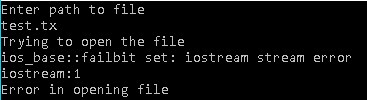


рис.1 – пример работы программы №1

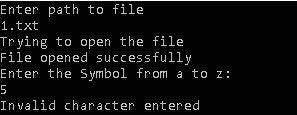


рис.2 – пример работы программы №2

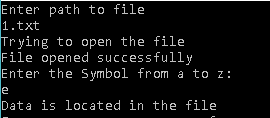


рис.3 – пример работы программы №3

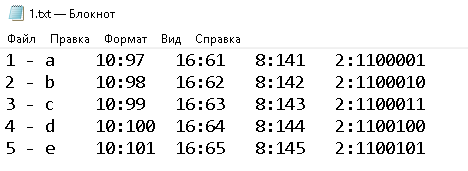


рис.4 – содержание файла 1.txt для примера №3

В примере №1 происходит попытка открыть несуществующий файл.

В примере №2 вводится символ, который не подходит условию (от a до z).

Пример №3 работает верно.

**Вывод:**

В ходе данной лабораторной работы были изучены способы реализации и особенности управления потоками ввода/вывода, исследованы способы генерации и обработки исключений.