МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

ИНСТИТУТ НЕПРЕРЫВНОГО И ДИСТАНЦИОННОГО ОБРАЗОВАНИЯ

|  |
| --- |
| КАФЕДРА компьютерных технологий и программной инженерии |

ОЦЕНКА

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| «ОБРАБОТКА ДАННЫХ В ФАЙЛАХ» |
| по дисциплине: Основы программирования |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | Z9431 |  |  |  | Андреев Д.И. |
|  | номер группы |  | подпись, дата |  | инициалы, фамилия |
| Студенческий билет № | 2019/3781 | |  |  |  |

Санкт-Петербург 2020

Оглавление

[1. Цель работы 3](#_Toc55649402)

[2. Задание 3](#_Toc55649403)

[3. Описание созданных функций 3](#_Toc55649404)

[4. Листинг программы 5](#_Toc55649405)

[5. Пример выполнения программы 6](#_Toc55649406)

[6. Анализ результатов и выводы 7](#_Toc55649413)

1. Цель работы

Целью работы является изучение файловых операций ввода-вывода.

1. Задание

Вариант 1

Написать программу, которая считывает из текстового файла три предложения и записывает их в другой файл в обратном порядке.

1. Описание созданных функций

Для решения поставленной в лабораторной работе задачи потребуются следующие функции:

Имя: readWrite

Назначение: Считывает из текстового файла три предложения и записывает их в другой файл в обратном порядке

Входные данные:

* inFile – полный путь к входному файлу
* outFile – полный путь к выходному файлу

Выходные данные: отсутствуют.

Побочный эффект: записывает в выходной файл считанные из входного файла 3 первых предложения в обратном порядке.

Тестовые данные:

Everyone could notice that people often complain and say that it was better before.

It may be a little nostalgia; the old people would like to return in the 50s, forty-years-old people prefer the 70s.

It can be explained of the life period when they were younger.

As it’s known people like their youth time. In general living is better now.

We live when there aren’t as many wars in the world as before.

And people live longer; the level of lifespan is higher than it was 40 years ago.

People work less and they can afford to go to the vacation and buy more things.

Результат выполнения функции:

It can be explained of the life period when they were younger.

It may be a little nostalgia; the old people would like to return in the 50s, forty-years-old people prefer the 70s.

Everyone could notice that people often complain and say that it was better before.

Прототип: void readWrite(const char \*inFile, const char \*outFile);

Алгоритм:

Псевдокод:

Открыть файл inFile для чтения

Создать двумерный массив символов string для хранения трех предложения

Для трех предложений

Для каждого символа в предложении и пока из файла есть что читать

Помещаем символ в массив string

Если этот символ – точка

Начинаем читать новое предложение

Выйти из цикла

Закрыть файл

Открыть файл outFile для записи

Для каждого предложения начиная с последнего

Записать предложние в файл

Закрыть файл

Блок-схема:



1. Листинг программы

#include <iostream>

#include <fstream>

#include <Windows.h>

using namespace std;

//определения функций

void showHint();

short getCommand();

void setFileName(char\* inFile, char\* outFile, size\_t nameSize);

void readWrite(const char\* inFile, const char\* outFile);

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

cout << "Лабораторная работа №6.\nОбработка числовых матриц.\nВариант " << (1 - 1) % 20 + 1 << endl;

short command = -1;

const size\_t fileNameSize = 100;

char inFile[fileNameSize]{};

char outFile[fileNameSize]{};

do {

command = getCommand();

switch (command)

{

case 0:

setFileName(inFile, outFile, fileNameSize);

break;

case 1:

readWrite(inFile, outFile);

break;

case 2:

showHint();

break;

case 3:

break;

default:

cout << "Неизвестная команда\n";

break;

}

} while (command != 3);

return 0;

}

short getCommand()

{

#undef max

short com;

cout << "Введите команду для продолжения (2 - показать все команды): ";

cin >> com;

while (!cin) {

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

cout << "Введите корректное значение (3 для выхода): ";

cin >> com;

}

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

return com;

#define max

}

void setFileName(char\* inFile, char\* outFile, size\_t nameSize)

{

#undef max

cout << "Введите полный путь к входному файлу (до 100 символов): ";

cin.getline(inFile, nameSize);

cout << "Введите полный путь к выходному файлу (до 100 символов): ";

cin.getline(outFile, nameSize);

#define max

}

void readWrite(const char\* inFile, const char\* outFile)

{

fstream fs;

fs.open(inFile, ios\_base::in);

if (!fs.is\_open()) {

cerr << "Невозможно открыть файл " << inFile << " для чтения\n";

return;

}

char string[3][1000]{};

for (int count = 0; count < 3;) {

for (int i = 0; i < 1000 && !fs.eof(); ++i) {

fs.get(string[count][i]);

if (string[count][i] == '.') {

++count;

break;

}

}

}

fs.close();

fs.open(outFile, ios\_base::out);

if(!fs.is\_open()) {

cerr << "Невозможно открыть файл " << outFile << " для записи\n";

return;

}

for (int i = 2; !(i < 0); --i)

fs << string[i];

fs.close();

}

void showHint()

{

cout << "0 - Ввести имена файлов\n";

cout << "1 - Запустить функцию чтения - записи\n";

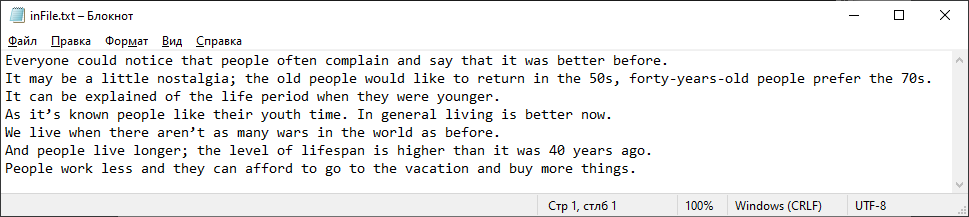
cout << "2 - Показать все команды\n";

cout << "3 - Выход\n";

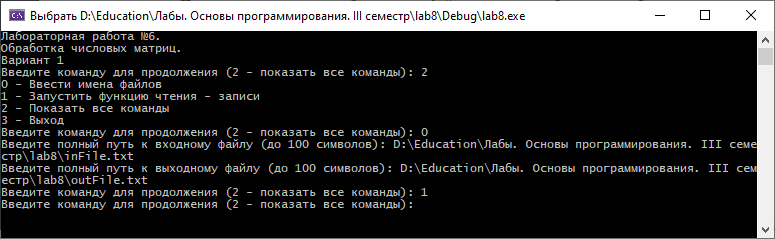
}

1. Пример выполнения программы

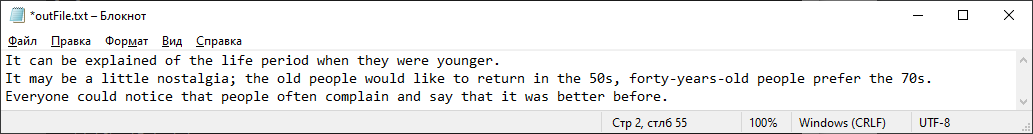
Входной файл



Выполнение программы



Выходной файл



1. Анализ результатов и выводы

К достоинствам программы можно отнести:

Программа работает без ошибок и верно выполняет поставленную задачу

Простота реализации и легко воспринимаемый код программы, а также реализация заданий в виде функций дает возможность повторного использования кода в других проектах

К недостаткам можно отнести:

Максимальная длина предложений и имен файлов ограничена 1000 и 100 символами соответственно, что снижает гибкость и понятность для пользователя