|  |  |
| --- | --- |
| C:\Documents and Settings\admin\Рабочий стол\МИРЭА_ЭМБЛЕМА_приказ.JPG | |
| МИНОБРАНАУКИ РОССИИ | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  «**МОСКОВСКИЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»**  **МИРЭА** | |
| Институт информационных технологий  Кафедра математического обеспечения и стандартизации информационных технологий | |
| **Отчет по лабораторной работе №7** **Тема работы «Файлы. Работа с файлами»** | |
| Выполнил: |  |
| Студент группы ИКБО-07-17  Отчет принял:  Преподаватель кафедры МОСИТ | Данилов Д.Е.  Миронов Антон Николаевич |
| Москва 2018 | |

# **Задание 1**

# **Условие задания**

* Создание модуля с операциями над текстовым файлом.

1. Создание текстового файла, содержащего числовые значения по одному на строку;
2. Вывод содержимого текстового файла;
3. Добавление новой записи в конец файла:

* Создать новый файл из значений исходного, поделив каждое число на число, равное количеству строк в файле.

# **Алгоритмы функций**

void Create\_txt(int &c) //Создание нового текстового файла

{

ofstream f("myfile.txt");

cout << "Введите количество строк" << endl;

cin >> c;

for (int i = 1; i <= c; i++)

{

f << i << endl;

}

f.close();

}

void Out\_File\_txt() //Вывод записей в файле

{

char ch = 0;

ifstream in("myfile.txt");

while ((ch = in.get()) != EOF)

{

cout << ch;

}

in.close();

}

void Update\_File(int &c) //Добавление новой записи в конец файла

{

ofstream f;

string a;

getchar();

getline(cin, a, '\n');

f.open("myfile.txt", ios::out | ios::app);

f << a << endl;

c++;

f.close();

}

void New\_txt(int c) //Новый из исходного с доп.операцией

{

int a;

ofstream f("myfile\_1.txt");

ifstream in("myfile.txt");

for (int i = 1; i <= c; i++)

{

in >> a;

a /= c;

f << a << endl;

a = 0;

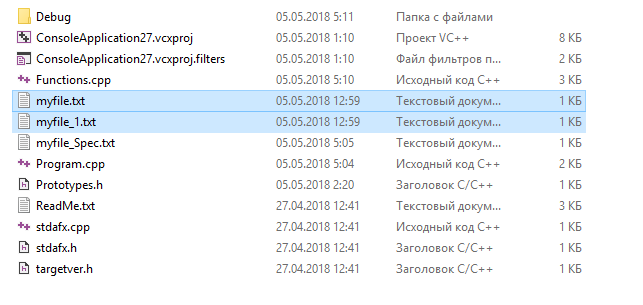
}

f.close();

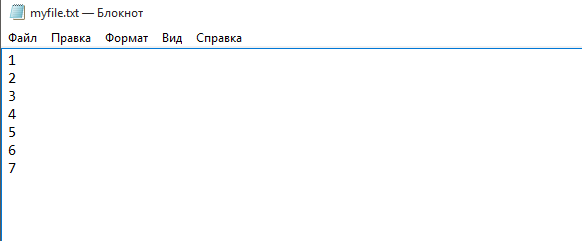
in.close();

}

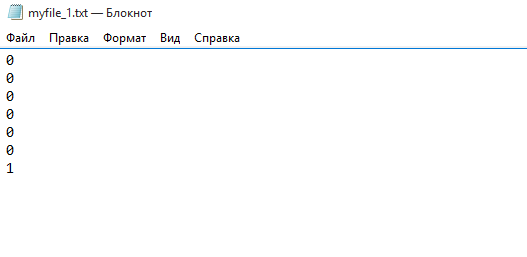
**Файлы, полученные при работе программы**



Заполнение файла 6-ю строками и добавление в конец файла 7 строки:



При делении всех чисел на 7 получаем новый файл:



# **Задание 2**

# **Условие задания**

Создание модуля с операциями над двоичными файлами.

1. Подготовьте тестовые данные в текстовом файле с кодировкой ASCII.
2. Разработайте структуру записи двоичного файла согласно варианту задания:

|  |  |
| --- | --- |
| Структура записи | Специализация вуза: код специальности, название вуза, название специальности |

1. Разработайте функции для операций:

* преобразование тестовых данных в двоичный файл
* отображение всех записей двоичного файла
* манипулирование записями в двоичном файле: согласно операциям определенных в варианте:

|  |  |
| --- | --- |
| Доп. операция | 1. Сформировать список вузов, которые ведут подготовку специалистов по специальности с указанным номером. Список представить в текстовом файле. 2. Изменить код специальности по названию специальности |

1. Сохраните функции в новом модуле.

Разработать приложение, демонстрирующее выполнение всех операций, подключив к нему модуль с функциями.

# **Алгоритмы функций**

void Create\_bin\_file(ifstream &ft, ofstream &fb, vector<Specialization> &s) //Создание бинарного файла

{

ft.open("myfile\_Spec.txt", ios::out);

fb.open("myfile\_Spec\_2.txt", ios::out | ios::binary);

Specialization x;

while (!ft.eof())

{

getline(ft, x.Name);

getline(ft, x.Spec);

ft >> x.Code;

ft.get();

fb.write((char \*)&x, sizeof(Specialization));

fb.clear();

s.push\_back(x);

}

ft.close();

fb.close();

}

void Out\_bin\_file(ifstream &fb, int a) //Вывод бинарного файла

{

fb.open("myfile\_Spec\_2.txt", ios::binary);

Specialization x;

fb.read((char \*)&x, sizeof(Specialization));

while (!fb.eof())

{

cout << x.Name << endl;

cout << x.Spec << endl;

cout << x.Code << endl;

fb.read((char \*)&x, sizeof(Specialization));

}

}

void New\_txt\_High\_School(vector<Specialization> s) //Новый текстовый файл с ВУЗами введенной специальности

{

int a;

ofstream f("High\_School.txt");

cout << "Введите код специальности" << endl;

cin >> a;

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

if (a == s[i].Code)

f << s[i].Name << endl;

}

void Change\_Code(vector<Specialization> &s) //Смена специальности по названию ВУЗа

{

ofstream ft("myfile\_Spec.txt", ios::out);

string ss;

int a;

int temp;

cout << "Введите название вуза" << endl;

cin >> ss;

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

{

if (ss == s[i].Name)

{

cout << "Введите новый код специальности" << endl;

cin >> a;

temp = s[i].Code;

for (int j = 0; j < s.size(); j++)

{

if (s[j].Code == temp)

{

s[j].Code = a;

}

}

}

}

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

{

ft << s[i].Name << endl;

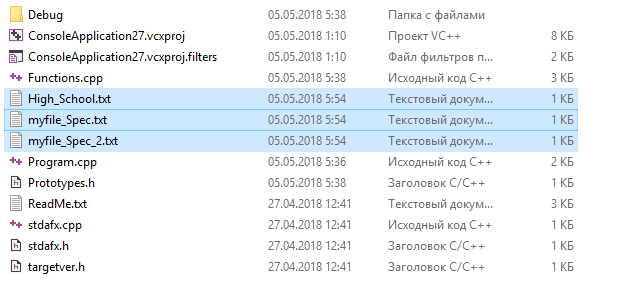
ft << s[i].Spec << endl;

ft << s[i].Code << endl;

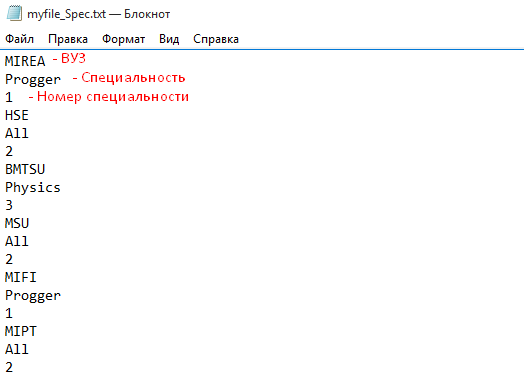
}

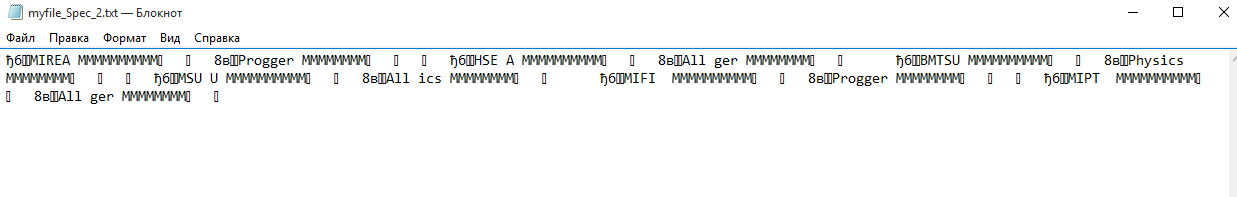
}

**Файлы, полученные при работе программы**

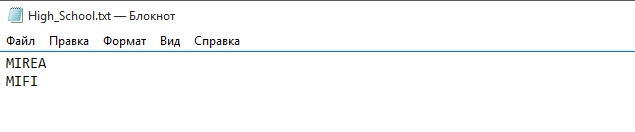


Созданный заранее файл с элементами структуры:

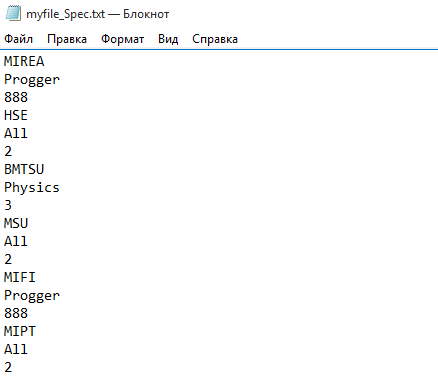


Созданный бинарный файл: 

Созданный файл с названиями ВУЗов по введенному номер специальности (1):



Измененный файл с элементами структур (Были изменены номера специальностей у ВУЗов с таким же номером специальности, как и у MIREA)



**Кодировка алгоритма программы:**

**Functions.cpp**

#include "stdafx.h"

#include "Prototypes.h"

#include <fstream>

#include <iostream>

#include <string>

#include <vector>

#include <string.h>

using namespace std;

void Create\_txt(int &c)

{

ofstream f("myfile.txt");

cout << "Введите количество строк" << endl;

cin >> c;

for (int i = 1; i <= c; i++)

{

f << i << endl;

}

f.close();

}

void Out\_File\_txt()

{

char ch = 0;

ifstream in("myfile.txt");

while ((ch = in.get()) != EOF)

{

cout << ch;

}

in.close();

}

void Update\_File(int &c)

{

ofstream f;

string a;

getchar();

getline(cin, a, '\n');

f.open("myfile.txt", ios::out | ios::app);

f << a << endl;

c++;

f.close();

}

void New\_txt(int c)

{

int a;

ofstream f("myfile\_1.txt");

ifstream in("myfile.txt");

for (int i = 1; i <= c; i++)

{

in >> a;

a /= c;

f << a << endl;

a = 0;

}

f.close();

in.close();

}

void Create\_bin\_file(ifstream &ft, ofstream &fb, vector<Specialization> &s)

{

ft.open("myfile\_Spec.txt", ios::out);

fb.open("myfile\_Spec\_2.txt", ios::out | ios::binary);

Specialization x;

while (!ft.eof())

{

getline(ft, x.Name);

getline(ft, x.Spec);

ft >> x.Code;

ft.get();

fb.write((char \*)&x, sizeof(Specialization));

fb.clear();

s.push\_back(x);

}

ft.close();

fb.close();

}

void Out\_bin\_file(ifstream &fb, int a)

{

fb.open("myfile\_Spec\_2.txt", ios::binary);

Specialization x;

fb.read((char \*)&x, sizeof(Specialization));

while (!fb.eof())

{

cout << x.Name << endl;

cout << x.Spec << endl;

cout << x.Code << endl;

fb.read((char \*)&x, sizeof(Specialization));

}

}

void New\_txt\_High\_School(vector<Specialization> s)

{

int a;

ofstream f("High\_School.txt");

cout << "Введите код специальности" << endl;

cin >> a;

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

if (a == s[i].Code)

f << s[i].Name << endl;

}

void Change\_Code(vector<Specialization> &s)

{

ofstream ft("myfile\_Spec.txt", ios::out);

string ss;

int a;

int temp;

cout << "Введите название вуза" << endl;

cin >> ss;

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

{

if (ss == s[i].Name)

{

cout << "Введите новый код специальности" << endl;

cin >> a;

temp = s[i].Code;

for (int j = 0; j < s.size(); j++)

{

if (s[j].Code == temp)

{

s[j].Code = a;

}

}

}

}

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

{

ft << s[i].Name << endl;

ft << s[i].Spec << endl;

ft << s[i].Code << endl;

}

}

**Prototypes.h**

#pragma once

#include "stdafx.h"

#include <string>

#include <fstream>

#include <iostream>

#include <vector>

using namespace std;

struct Specialization

{

string Name;

string Spec;

int Code;

};

void Create\_txt(int &c);

void Out\_File\_txt();

void Update\_File(int &c);

void New\_txt(int c);

void Create\_bin\_file(ifstream &ft, ofstream &fb, vector<Specialization> &s);

void Out\_bin\_file(ifstream &fb, int a);

void New\_txt\_High\_School(vector<Specialization> s);

void Change\_Code(vector<Specialization> &s);

**Program.cpp**

#include "stdafx.h"

#include "Prototypes.h"

#include <Windows.h>

#include <vector>

using namespace std;

int main()

{

vector<Specialization> s;

ifstream ft;

ofstream fb;

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int a = 0;

int count = 0;

int ans = 10;

while (ans != 0)

{

system("cls");

cout << "Лабораторная работа №7 ИКБО-07-17 Данилов Д.Е. Вариант 7" << endl << endl;

cout << "Задание 1:" << endl;

cout << "Меню" << endl;

cout << "1) Создание текстового файла" << endl;

cout << "2) Вывод содержимого текстового файла" << endl;

cout << "3) Добавление новой записи в конец" << endl;

cout << "Дополнительные операции:" << endl;

cout << "4) Создание нового файла из исходного, поделив все числа на количество строк" << endl << endl;

cout << "Задание 2:" << endl;

cout << "Меню" << endl;

cout << "5) Преобразование данных в двоичный файл" << endl;

cout << "6) Отображение записей двоичного файла" << endl;

cout << "Дополнительные операции:" << endl;

cout << "7) Сформировать список вузов, ведущих подготовку спец-ов по спец-ти с указанным номером" << endl;

cout << "8) Изменить код специальности по названию" << endl << endl;

cout << "0) Выход" << endl;

cout << "Ваш выбор: ";

cin >> ans;

system("cls");

cout << "Лабораторная работа №7 ИКБО-07-17 Данилов Д.Е. Вариант 7" << endl << endl;

switch (ans)

{

case 1: Create\_txt(count); system("pause"); break;

case 2: Out\_File\_txt(); system("pause"); break;

case 3: Update\_File(count); system("pause"); break;

case 4: New\_txt(count); system("pause"); break;

case 5: Create\_bin\_file(ft, fb, s); system("pause"); break;

case 6: Out\_bin\_file(ft, a); system("pause"); break;

case 7: New\_txt\_High\_School(s); system("pause"); break;

case 8: Change\_Code(s); system("pause"); break;

default: break;

}

}

return 0;

}