**ПРИЛОЖЕНИЕ А**

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

# КАФЕДРА ИНТЕЛЛЕКТУАЛЬНЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

ОБРАБОТКА МАССИВОВ СТРУКТУРИРОВАННЫХ ДАННЫХ

**«**ШКОЛЬНИК**»**

**ТЕКСТ ПРОГРАММЫ**

КР.АС56.190261-01 12 00

## Листов 15

|  |  |
| --- | --- |
| Руководитель | Т.Н. Гирель |
| Выполнил | М.В. Карпенко |
| Консультант |  |
| по ЕСПД | Т.Н. Гирель |
|  |  |

2020

СОДЕРЖАНИЕ

main.cpp – основной модуль в котором содержится меню для работы с программой.

#include <iostream>

#include <fstream>

#include<windows.h>

#include<cstring>

#include<string>

using namespace std;

struct Schooler

{

char surname[20];

char name[20];

int age;

int class\_number;

int mark;

};

Schooler\* fromFile(string path, int st);

void add(string path);

void out(Schooler\* data, int st);

void toFile(string path, Schooler\* data, int st);

void sort(Schooler\* data, int st, string pathi);

void correct(Schooler\* data);

void deleting(Schooler\* data, int st,string path);

void insert(Schooler\* data, int st, string path);

void filter(Schooler\* data, int st);

void average(Schooler\* data, int st);

void menu(Schooler\* data, int st, string path);

void find(Schooler\* data, int st);

int f(char\* s);

int main()

{

setlocale(0, "");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

bool t = true;

int st=0;

string path = "R:\\forKurs.txt";

string pathi = "R:\\KursIndecses.txt";

ifstream file(path, ios\_base::in);

ofstream index(pathi, ios\_base::out);

if (!file.is\_open() || !index.is\_open())

{

cout << "Ошибка открытия файлов" << endl;

}

else

{

string s;

ifstream f1;

f1.open(path);

while (getline(f1, s))

st++;

st = st / 5;

f1.close();

file.close();

index.close();

Schooler\* p;

p = fromFile(path, st);

cout << "Файлы открыты" << endl;

cout << "Массив заполнен" << endl;

Sleep(1000);

while (t)

{

system("cls");

cout << "Количество структур:" << st << endl;

cout << "Выберете: " << endl;

cout << " 0 Перезаполнение массива" << endl;

cout << " 1 Вывод массива структур" << endl;

cout << " 2 Добавление записи в файл" << endl;

cout << " 3 Перезапись файла" << endl;

cout << " 4 Сортировка" << endl;

cout << " 5 Изменение заданной структуры" << endl;

cout << " 6 Удаление структуры из массива" << endl;

cout << " 7 Поиск структуры по заданному полю" << endl;

cout << " 8 Дополнительное меню" << endl;

cout << " 9 Выход. " << endl;

int n;

cin >> n;

system("cls");

switch (n)

{

case 0:

p = fromFile(path, st);

system("pause");

break;

case 1:

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

out(p, i);

system("pause");

break;

case 2:

add(path);

st++;

system("pause");

break;

case 3:

toFile(path, p, st);

system("pause");

break;

case 4:

sort(p,st,pathi);

system("pause");

break;

case 5:

correct(p);

system("pause");

break;

case 6:

deleting(p, st,path);

st--;

system("pause");

break;

case 7:

find(p, st);

system("pause");

break;

case 8:

menu(p, st, path);

system("pause");

break;

case 9:

t = false;

break;

default:

cout << "ERROR!!!" << endl;

break;

}

}

}

system("pause");

return 0;

};

int f(char\* s)

{

int temp = 0; // число

int i = 0;

while (s[i] >= 0x30 && s[i] <= 0x39)

{

temp = temp + (s[i] & 0x0F);

temp = temp \* 10;

i++;

}

temp = temp / 10;

return(temp);

}

Schooler\* fromFile(string path, int st)

{

ifstream file;

file.open(path);

char\* str = new char[20];

Schooler\* data = new Schooler[st];

if (!file.is\_open())

{

cout << "ERROR!!!" << endl;

}

else

{

char buff[20];

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

{

file >> data[i].surname;

file >> data[i].name;

file >> buff;

data[i].age = f(buff);

file >> buff;

data[i].class\_number = f(buff);

file >> buff;

data[i].mark = f(buff);

}

}

file.close();

cout << "Перезаполнение массива выполнено" << endl;

return data;

};

void out(Schooler\* data, int i)

{

cout << data[i].surname << endl;

cout << data[i].name << endl;

cout << data[i].age << endl;

cout << data[i].class\_number << endl;

cout << data[i].mark << endl;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

};

void add(string path)

{

ofstream f;

f.open(path,ios::app);

string str;

cout << "Введите фамилию" << endl;

cin >> str;

f << str;

f << "\n";

cout << "Введите имя" << endl;

cin >> str;

f << str;

f << "\n";

cout << "Введите возраст" << endl;

cin >> str;

f << str;

f << "\n";

cout << "Введите номер класса " << endl;

cin >> str;

f << str;

f << "\n";

cout << "Введите средний балл" << endl;

cin >> str;

f << str;

f << "\n";

system("cls");

cout << "Введите 0 для перезаполнения массива" << endl;

f.close();

};

void toFile(string path, Schooler\* data, int st)

{

ofstream file;

file.open(path);

if (!file.is\_open())

{

cout << "ERROR!!!" << endl;

}

else

{

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

{

file << data[i].surname << "\n";

file << data[i].name << "\n";

file << data[i].age << "\n";

file << data[i].class\_number << "\n";

file << data[i].mark << "\n";

}

file.close();

}

cout << "Перезапись файла выполнена" << endl;

};

void sort(Schooler\* data, int st, string pathi)

{

Schooler temp;

cout << "Выберете параметр сортировки: " << endl;

cout << " 1 Фамилия" << endl;

cout << " 2 Имя" << endl;

cout << " 3 Возраст" << endl;

cout << " 4 Класс" << endl;

cout << " 5 Средний балл" << endl;

int c;

cin >> c;

switch (c)

{

case 1:

for (int k = 0; k < st; k++)

{

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

{

if ((int)data[i].surname[0] > (int)data[i + 1].surname[0])

{

temp = data[i];

data[i] = data[i + 1];

data[i + 1] = temp;

}

}

}

break;

case 2:

for (int k = 0; k < st; k++)

{

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

{

if ((int)data[i].name[0] > (int)data[i + 1].name[0])

{

temp = data[i];

data[i] = data[i + 1];

data[i + 1] = temp;

}

}

}

break;

case 3:

for (int k = 0; k < st; k++)

{

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

{

if (data[i].age > data[i + 1].age)

{

temp = data[i];

data[i] = data[i + 1];

data[i + 1] = temp;

}

}

}

break;

case 4:

for (int k = 0; k < st; k++)

{

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

{

if (data[i].class\_number > data[i + 1].class\_number)

{

temp = data[i];

data[i] = data[i + 1];

data[i + 1] = temp;

}

}

}

break;

case 5:

for (int k = 0; k < st; k++)

{

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

{

if (data[i].mark > data[i + 1].mark)

{

temp = data[i];

data[i] = data[i + 1];

data[i + 1] = temp;

}

}

}

break;

default:

cout << "ERROR!!!" << endl;

break;

}

ofstream index;

index.open(pathi);

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

{

index << data[i].surname << "\n";

index << data[i].name << "\n";

index << data[i].age << "\n";

index << data[i].class\_number << "\n";

index << data[i].mark << "\n";

}

index.close();

};

void correct(Schooler\* data)

{

cout << "Введите номер структуры" << endl;

int a;

cin >> a;

a--;

cout << "Что изменить?" << endl;

cout << "1-Фамилию" << endl;

cout << "2-Имя" << endl;

cout << "3-Возраст" << endl;

cout << "4-Класс" << endl;

cout << "5-Средний балл" << endl;

int n;

cin >> n;

switch (n)

{

case 1:

cin >> data[a].surname;

break;

case 2:

cin >> data[a].name;

break;

case 3:

cin >> data[a].age;

break;

case 4:

cin >> data[a].class\_number;

break;

case 5:

cin >> data[a].mark;

break;

default:

cout << "ERROR!!!" << endl;

break;

}

cout << "Введите 3 для перезаписи файла" << endl;

};

void deleting(Schooler\* data, int st,string path)

{

ofstream file;

file.open(path);

if (!file.is\_open())

{

cout << "ERROR!!!" << endl;

}

else

{

cout << "Введите номер структуры" << endl;

int k;

cin >> k;

k--;

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

{

if (i == k)continue;

file << data[i].surname << "\n";

file << data[i].name << "\n";

file << data[i].age << "\n";

file << data[i].class\_number << "\n";

file << data[i].mark << "\n";

}

}

cout << "Введите 0 для перезаполнения массива" << endl;

};

void find(Schooler\* data, int st)

{

cout << "Выберете параметр для поиска" << endl;

cout << " 1 Фамилия" << endl;

cout << " 2 Имя" << endl;

cout << " 3 Возраст" << endl;

cout << " 4 Класс" << endl;

cout << " 5 Средний балл" << endl;

int f;

cin >> f;

int find2, findi;

char finds[20];

switch (f)

{

case 1:

cin >> finds;

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

{

if (!strcmp(data[i].surname,finds))

{

find2 = i;

break;

}

}

break;

case 2:

cin >> finds;

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

{

if (!strcmp(data[i].name,finds))

{

find2 = i;

break;

}

}

break;

case 3:

cin >> findi;

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

{

if (data[i].age == findi)

{

find2 = i;

break;

}

}

break;

case 4:

cin >> findi;

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

{

if (data[i].class\_number == findi)

{

find2 = i;

break;

}

}

break;

case 5:

cin >> findi;

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

{

if (data[i].mark == findi)

{

find2 = i;

break;

}

}

break;

default:

cout << "ERROR!!!" << endl;

break;

}

out(data, find2);

};

void menu(Schooler\* data, int st, string path)

{

cout << "Выберете " << endl;

cout << "1 Вставка новой записи перед выбранной" << endl;

cout << "2 Выборка" << endl;

cout << "3 Фильтр" << endl;

int c;

cin >> c;

switch (c)

{

case 1:

insert(data, st, path);

st++;

cout << "Введите 0 для перезаполнения массива" << endl;

break;

case 2:

average(data, st);

break;

case 3:

filter(data, st);

break;

default:

cout << "ERROR!!!" << endl;

break;

}

};

void insert(Schooler\* data, int st, string path)

{

cout << "Введите номер позиции, в которую надо вписать структуру" << endl;

int n;

cin >> n;

ofstream file;

file.open(path);

Schooler flag;

if (!file.is\_open())

{

cout << "ERROR!!!" << endl;

}

else

{

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

{

if (i == n - 1)

{

string str;

cout << "Введите фамилию" << endl;

cin >> str;

file << str;

file << "\n";

cout << "Введите имя" << endl;

cin >> str;

file << str;

file << "\n";

cout << "Введите возраст" << endl;

cin >> str;

file << str;

file << "\n";

cout << "Введите номер класса " << endl;

cin >> str;

file << str;

file << "\n";

cout << "Введите средний балл" << endl;

cin >> str;

file << str;

file << "\n";

system("cls");

}

file << data[i].surname << "\n";

file << data[i].name << "\n";

file << data[i].age << "\n";

file << data[i].class\_number << "\n";

file << data[i].mark << "\n";

}

file.close();

}

};

void average(Schooler\* data, int st)

{

int a, b;

cout << "Введите левую границу диапазона" << endl;

cin >> a;

cout << "Введите правую границу диапазона" << endl;

cin >> b;

cout << "Выберете " << endl;

cout << "1 Среднее по полю \"Средний балл\"" << endl;

cout << "2 Среднее по полю \"Возраст\"" << endl;

int k;

cin >> k;

double av=0;

switch (k)

{

case 1:

for (int i = a - 1; i < (b - 1); i++)

{

av += data[i].mark;

}

av = av / (b - a);

cout << "Среднее значение равно:" << av << endl;

break;

case 2:

for (int i = a - 1; i < (b - 1); i++)

{

av += data[i].age;

}

av = av / (b - a);

cout << "Среднее значение равно:" << av << endl;

break;

default:

cout << "ERROR!!!" << endl;

break;

}

};

void filter(Schooler\* data, int st)

{

cout << "Выберете " << endl;

cout << "1 Фильтр по полю \"Средний балл\"" << endl;

cout << "2 Фильтр по полю \"Возраст\"" << endl;

cout << "3 Фильтр по полю \"Класс\"" << endl;

int k;

cin >> k;

cout << "Выберете " << endl;

cout << "1 Больше, чем выбранное Вами число" << endl;

cout << "2 Меньше, чем выбранное Вами число" << endl;

int q;

cin >> q;

cout << "Введите число" << endl;

int r;

cin >> r;

switch (k)

{

case 1:

switch (q)

{

case 1:

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

{

if (data[i].mark > r)

out(data, i);

}

break;

case 2:

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

{

if (data[i].mark < r)

out(data, i);

}

break;

default:

cout << "ERROR!!!" << endl;

break;

}

break;

case 2:

switch (q)

{

case 1:

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

{

if (data[i].age > r)

out(data, i);

}

break;

case 2:

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

{

if (data[i].age < r)

out(data, i);

}

break;

default:

cout << "ERROR!!!" << endl;

break;

}

break;

case 3:

switch (q)

{

case 1:

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

{

if (data[i].class\_number > r)

out(data, i);

}

break;

case 2:

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

{

if (data[i].class\_number < r)

out(data, i);

}

break;

default:

cout << "ERROR!!!" << endl;

break;

}

break;

default:

cout << "ERROR2!!!" << endl;

break;

}

};