Министерство образования и науки Российской Федерации

Севастопольский государственный университет

Кафедра ИС

Отчет

по лабораторной работе № 6(9)

«ПРОГРАММИРОВАНИЕ ОПЕРАЦИЙ НАД СТРУКТУРАМИ И БИНАРНЫМИ ФАЙЛАМИ»

Выполнил

ст. гр. ИТ-12д

Воронин И.Ю.

Проверил:

асс. Забаштанский А.К.

Севастополь

2015

1.ЦЕЛЬ РАБОТЫ

Изучение способов описания структур данных на языке С. Исследование особенностей обработки бинарных файлов, хранящих структурные типы данных.

2.ВАРИАНТ ЗАДАНИЯ

Вариант 12 Описать структуру с именем NOTE, содержащую следующие поля:

- фамилия, имя;

- номер телефона;

- дата рождения (массив из трех чисел).

Написать программу, выполняющую следующие действия с помощью функций:

- ввод с клавиатуры данных в файл, состоящий из элементов типа NOTE; записи должны быть упорядочены по датам рождения;

- чтение данных из этого файла;

- вывод на экран информации о человеке, номер телефона которого введен с клавиатуры;

- если такого нет, выдать на дисплей соответствующее сообщение

3.СТРУКТУРНЫЕ СХЕМЫ

Алгоритм работы данной программы представлен в следующей схеме:



Рисунок 3.1-Ввод данных



Рисунок 3.2-Вывод данных



Рисунок 3.3-Поиск по номеру телефона

4.ТЕКСТ ПРОГРАММЫ

#include <iostream>

#include <conio.h>

#include <iomanip>

const int FIOLEN = 15;

const int MOBILELEN = 10;

struct birthday{

int day;

int month;

int year;

};

struct NOTE{

char fio[FIOLEN];

char mobile[MOBILELEN];

birthday birth;

};

extern int GetSize();

extern void input();

extern void sort(NOTE\* s, int size);

extern void output();

extern void telephone();

using namespace std;

int main(){

setlocale(LC\_ALL,"Russian");

char c;

do{

cout<<'\n'<<'\n';

cout<<"1-Добавить новые данные. "<<'\n';

cout<<"2-Печать данных файла. "<<'\n';

cout<<"3-Поиск по номеру телефона. "<<'\n';

cout<<"0-Выход. "<<'\n';

cout << " Выберите пункт меню: ", cin >> c;

cin.clear();

switch (c){

case '1': input(); break;

case '2': output(); break;

case '3': telephone(); break;

default: continue;

}

}

while (c!='0');

return 0;

}

NOTE GetNote(){

NOTE a;

cin.clear();

cin.ignore();

cout << "FIO: ", cin.getline(a.fio, FIOLEN);

cout << "Mobile: ", cin.getline(a.mobile, MOBILELEN);

cout << "Birthday: \n";

cout << " Day: ", cin >> a.birth.day;

cout << " Month: ", cin >> a.birth.month;

cout << " Year: ", cin >> a.birth.year;

return a;

}

int GetSize(){

int size = 0;

FILE\* f = fopen("Data.dat", "r+");

//подсчёт количества элементов

for(NOTE s; fread(&s, sizeof(NOTE), 1, f) ; size++);

fclose(f);

return size;

}

void input(){

FILE\* f;

int gs = GetSize()+1;

NOTE temp, \*s = new NOTE[gs];

//добавление новых данных в конец массива

s[gs-1] = GetNote();

//считывание данных в массив

f = fopen("Data.dat","r+");

for (int i = 0; fread(&temp, sizeof(NOTE), 1, f); i++)

s[i] = temp;

fclose(f);

//сортировка массива

sort(s, gs-1);

//запись нового сортированного массива в файл

f = fopen("Data.dat","w+");

for(int j = 0; j<gs; j++)

fwrite(&s[j], sizeof(NOTE), 1, f);

fclose(f);

}

void sort(NOTE\* s, int size) {

NOTE tmp;

if (size == 1) return; //1 элемент

for (int i(0); i < size - 1; i++){

for (int j(1); j < size; j++){

//младшие уходят в конец массива, старшие-первые

if ((s[i].birth.year > s[j].birth.year)||

(s[i].birth.year == s[j].birth.year) && (s[i].birth.month > s[j].birth.month)||

(s[i].birth.year == s[j].birth.year) && (s[i].birth.month == s[j].birth.month)&&(s[i].birth.day > s[j].birth.day))

swap(s[i], s[j]);

}

}

}

void output(){

NOTE t;

FILE\* f = fopen("Data.dat","r+");

cout<<'\n'<<'\n';

cout<<'|'<<setw(FIOLEN)<<"FIO"<<'|'<<setw(MOBILELEN)<<"Mobile"<<'|'<<setw(12)<<"Birthday"<<'|'<<'\n';

while(fread(&t,sizeof(NOTE),1,f)){

cout<<'|'<<setw(FIOLEN)<<t.fio<<'|'<<setw(MOBILELEN)<<t.mobile<<'|'<<setw(8)<<

t.birth.day<<'.'<<t.birth.month<<'.'<<t.birth.year<<'|'<<'\n';

}

fclose(f);

}

void telephone(){

FILE\* f = fopen("Data.dat","R");

NOTE t,node;

cout<<"Enter number of telephone: ";

cin.clear();

cin.ignore();

while(fread(&t,sizeof(NOTE),1,f))

if(node.mobile == t.mobile) {

cout<<'|'<<setw(FIOLEN)<<"FIO"<<'|'<<setw(MOBILELEN)<<"Mobile"<<'|'<<setw(12)<<"Birthday"<<'|'<<'\n';

cout<<'|'<<setw(FIOLEN)<<t.fio<<'|'<<setw(MOBILELEN)<<t.mobile<<'|'<<setw(8)<<

t.birth.day<<'.'<<t.birth.month<<'.'<<t.birth.year<<'|'<<'\n';

fclose(f);

return;

}

cout<<"Matches no found...";

fclose(f);

}

5.ТЕСТОВЫЕ ПРИМЕРЫ

Запуск программы выглядит следующим образом (рис.5.1.)



Рисунок 5.1.:Запуск программы.

Печать массива изображена на рисунке 5.2. Данные отсортированы по возрасту.

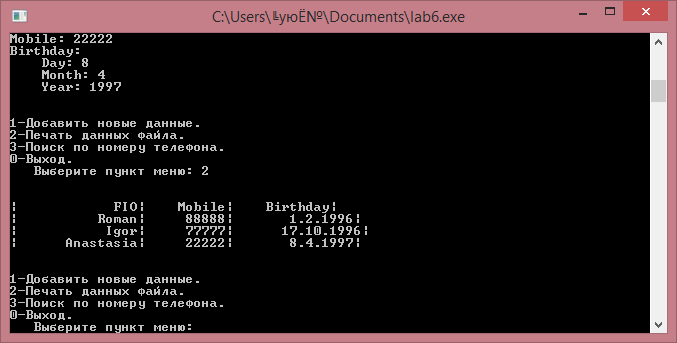


Рисунок 5.2.:Печать массива

ВЫВОДЫ

Была написана программа, работающая с бинарными файлами, которое основаны на использовании заданной нам структуры. Также методом пузырька данный файл был отсортирован по одному из полей структуры (дата рождения). Для ввода данных, все записи файла копировались в массив, после туда добавлялся новый элемент, проводилась сортировка, а по окончанию, данные в новом порядке записывались в файл.