МИНИСТЕРСВО ОБРАЗОВАНИЯ И НАУКИ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное

учреждение высшего профессионального образования  
«САНКТ-ПЕТЕРБУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА № 2

ОТЧЕТ ЗАЩИЩЕН С ОЦЕНКОЙ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ПРЕПОДАВАТЕЛЬ

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

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 2 |
| Запись данных в файл |
| По дисциплине: Информатика |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4710 |  |  |  |  |
|  |  |  | подпись, дата |  | (Фамилия, инициалы) |

Санкт-Петербург 2017

Цель работы:

-Научиться читать из файла  
-Научиться записывать в файл

-Научится сортировать данные

Условия задачи: Ваниант 15

Составить список, записать его в файл, закрыть файл, снова открыть данный файл и считать данные из него, отсортировать и записать отсортированный список в новый файл

Код программы:

#include <iostream>

#include <string>

#include <fstream>

#define COUNT 6

using namespace std;

struct Date{

int day;

int manth;

int year;

};

struct Stud{

string name;

char gendr;

char finish;

char adress;

char grand;

Date date;

int marks[5];

int point;

};

void openReadFile(string src, ifstream \*file){

while(true){

  cout << "input name of file\n";

  string name;

  cin >> name;

  string path = src + name;

file->open(path, ios\_base::app);

if(file->is\_open())

break;

cout << "complete" << endl;

}

}

void openWriteFile(string src, ofstream \*file){

while(true){

   cout << "input name of file\n";

   string name;

   cin >> name;

   string path = src + name;

file->open(path, ios\_base::app);

if(file->is\_open())

return;

std::cout << "Create in src folder? : ";

bool qs;

cin >> qs;

if(qs){

file->open(name);

return;

}

}

cout << "File is open\n";

}

void readFile(ifstream &file, Stud \*students){

file >> students->name >> students->gendr >> students->finish >>

students->adress >> students->grand >> students->date.day >>

students->date.manth >> students->date.year;

for(int k = 0; k < 5; k++)

file >> students->marks[k];

file >> students->point;

}

string writeString(Stud students){

string data("");

data += students.name + " " + students.gendr + " " + students.finish + " " +

students.adress + " " + students.grand + " " + to\_string(students.date.day) + " " +

to\_string(students.date.manth) + " " + to\_string(students.date.year) + " ";

for(int k = 0; k<5;k++)

data += to\_string(students.marks[k]) + " ";

data += to\_string(students.point) + "\n";

return data;

}

void input(Stud \*students){

cin >> students->name >> students->gendr >>

    students->finish >> students->adress >>

    students->grand >> students->date.day >>

    students->date.manth >> students->date.year;

for(int k = 0; k<5;k++)

cin >> students->marks[k];

cin >> students->point;

}

void output(Stud students){

cout << students.name << ' ' << students.gendr <<

' ' << students.finish << ' ' << students.adress <<

' ' << students.grand << ' ' << students.date.day <<

' ' << students.date.manth << ' ' << students.date.year << ' ';

for(int k = 0; k<5;k++)

cout << students.marks[k] << ' ';

cout << students.point << endl;

}

int main(){

string src = "/home/nikita/Documents/Coding/c++/Labki/Self/LabaFive/";

ifstream iFile; //read

ofstream oFile; //write

ofstream sOFile; //write

string data("");

Stud students[COUNT];

cout << "Open file for write" << endl;

openWriteFile(src, &oFile);

cout << "input students" << endl;

for(int i = 0; i < COUNT; i++){

input(&students[i]);

data += writeString(students[i]);

}

cout << "Writing to file" << endl;

oFile << data;

oFile.close();

cout << "Open file for read" << endl;

openReadFile(src, &iFile);

cout << "Read and output students" <<endl;

for(int i = 0; i < COUNT; i++){

readFile(iFile, &students[i]);

output(students[i]);

}

iFile.close();

cout << "input name of file for writing sorted list" << endl;

openWriteFile(src, &sOFile);

cout << "Sorted" << endl;

data = "";

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

if((students[i].grand == 'y') &&

(students[i].date.year >= 1989)&&

(students[i].date.year <= 1990)){

output(students[i]);

data += writeString(students[i]);

}

cout << "Write to file" << endl;

sOFile << data;

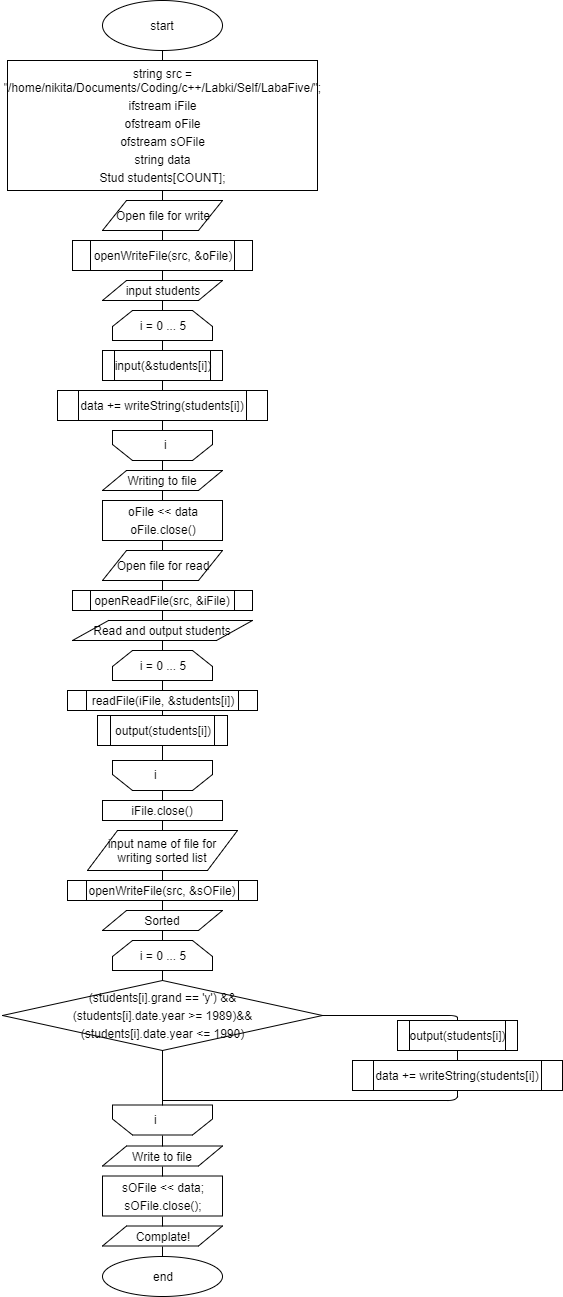
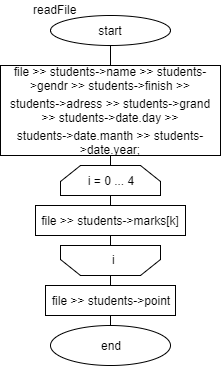
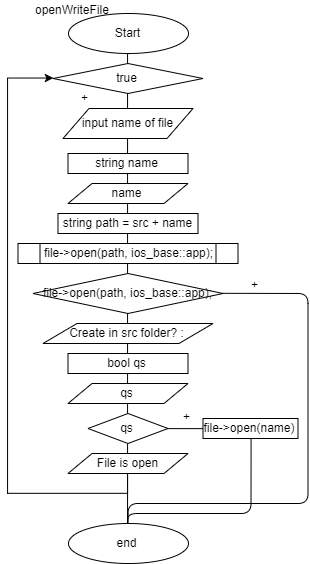
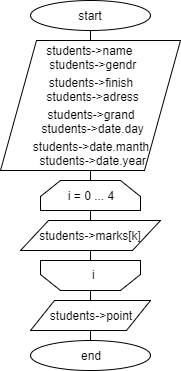
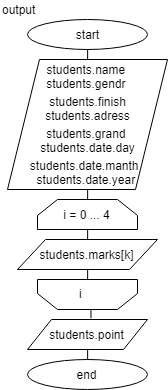
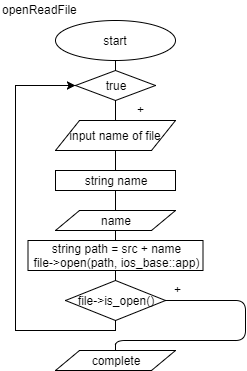
sOFile.close();

cout << "Complate!" << endl;

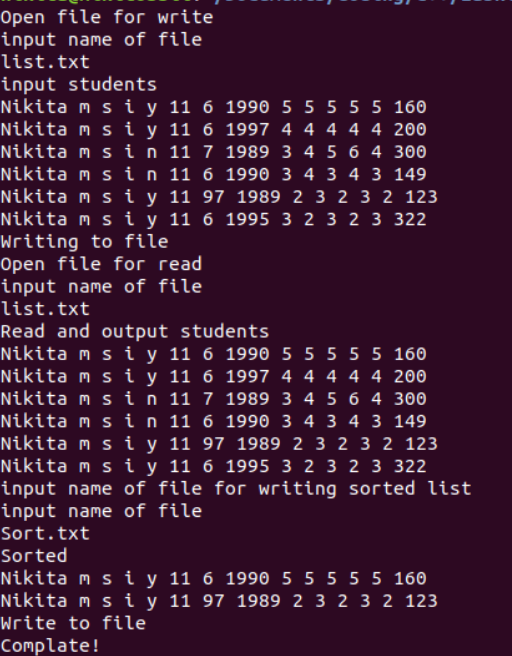
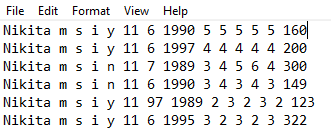
return 0;

}

Алгоритмы:



Пример работы кода:

Вывод:

-Научился Записывать данные в файл

-Научился читать данные из файла

-Научился сортировать списки