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 № 5 |
| Обработка ведомости студенческой группы |
| По дисциплине: Информатика |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

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

Санкт-Петербург 2018

Цель работы:

-Научиться читать из файла  
-Научиться записывать в файл

-Научится сортировать данные

Условия задачи: Ваниант 6

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;

};

ifstream 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::out);

if(file.is\_open())

break;

cout << "complete" << endl;

}

return file;

}

ofstream 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::in);

if(file.is\_open())

break;

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

bool qs;

cin >> qs;

if(qs){

file.open(name);

break;

}

}

cout << "File is open\n";

return file;

}

Stud 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;

return students;

}

void Sort(Stud \*students){

    bool ex = false;

    while(!ex){

        ex = true;

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

float tmp1 = 0;

float tmp2 = 0;

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

tmp1 += students[i].marks[k];

tmp2 += students[i+1].marks[k];

}

if(tmp2 > tmp1){

Stud tmp = students[i];

students[i] = students[i + 1];

students[i + 1] = tmp;

ex = false;

}

}

    }

}

int main(){

int centerPoint = 0;

string src = "C:\\Users\\Nikitka\\Documents\\Coding\\C++\\Labki\\LabaMarkin\\LabaFive\\";

ifstream forRead; //read //iFile

ofstream forWrite; //write oFile

ofstream forWrite2; //write sOFile

string data("");

Stud students[COUNT];

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

forWrite = openWriteFile(src);

cout << "input students" << endl;

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

cin >> students[i].name >> students[i].gendr >>

    students[i].finish >> students[i].adress >>

    students[i].grand >> students[i].date.day >>

    students[i].date.manth >> students[i].date.year;

data += students[i].name + " " + students[i].gendr + " " + students[i].finish + " " +

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

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

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

cin >> students[i].marks[k];

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

}

cin >> students[i].point;

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

}

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

forWrite << data;

forWrite.close();

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

forRead = openReadFile(src);

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

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

students[i] = readFile(forRead);

forRead.close();

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

cout << students[i].name << ' ' << students[i].gendr <<

' ' << students[i].finish << ' ' << students[i].adress <<

' ' << students[i].grand << ' ' << students[i].date.day <<

' ' << students[i].date.manth << ' ' << students[i].date.year << ' ';

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

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

cout << students[i].point << endl;

}

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

forWrite2 = openWriteFile(src);

cout << "Sorted" << endl;

data = "";

Sort(students);

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

cout << students[i].name << ' ' << students[i].gendr <<

' ' << students[i].finish << ' ' << students[i].adress <<

' ' << students[i].grand << ' ' << students[i].date.day <<

' ' << students[i].date.manth << ' ' << students[i].date.year << ' ';

data += students[i].name + " " + students[i].gendr + " " + students[i].finish + " " +

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

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

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

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

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

}

cout << students[i].point << endl;

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

}

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

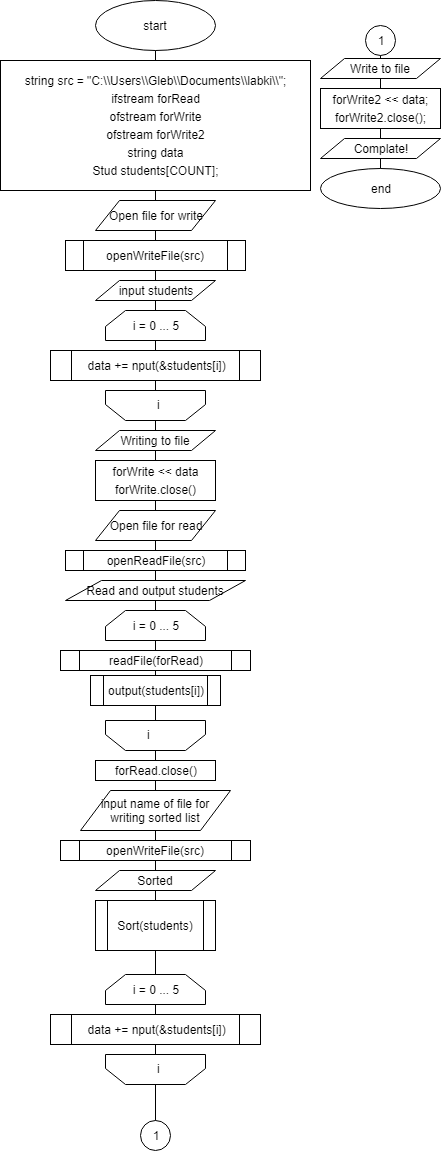
forWrite2 << data;

forWrite2.close();

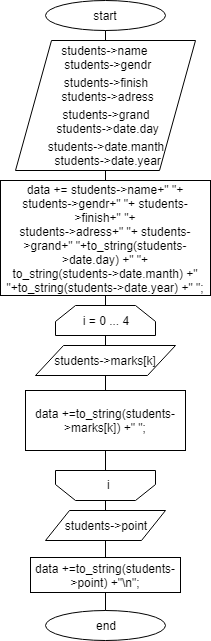
cout << "Complate!" << endl;

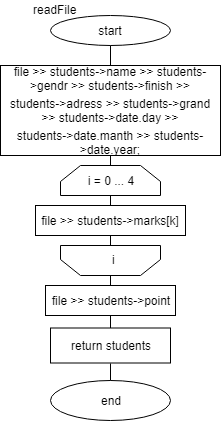
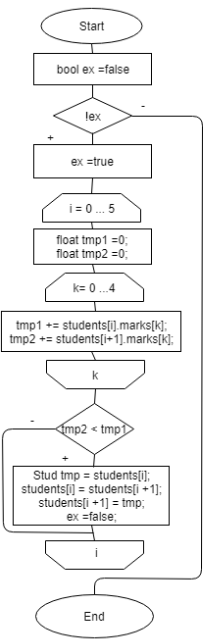
return 0;

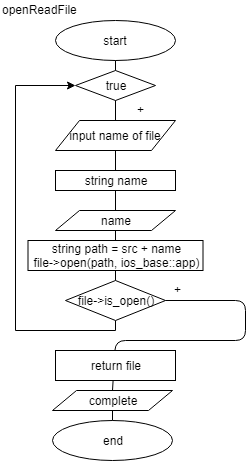
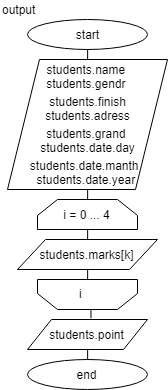
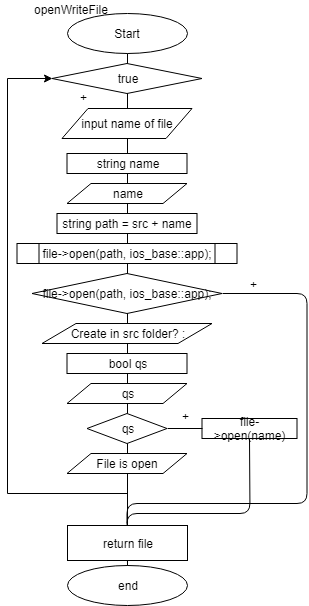
}



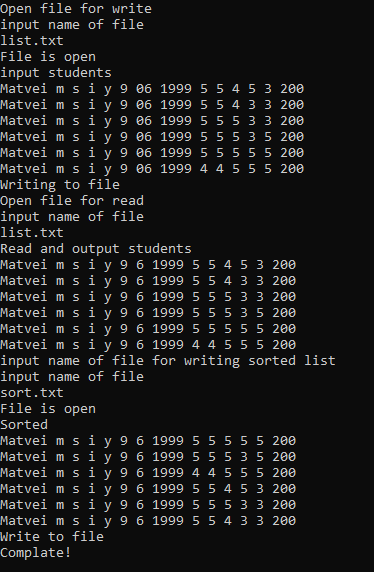
Алгоритмы:

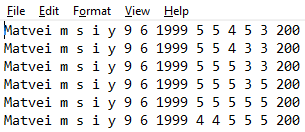


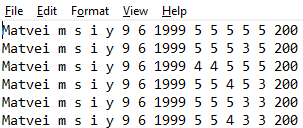




Пример работы кода:







Вывод:

-Научился Записывать данные в файл

-Научился читать данные из файла

-Научился сортировать списки