

Задание. Составить и выполнить программу, которая сохраняет данные о студентах (фамилия, имя, отчество, номер группы) в файл, считывает данные из файла и выводит на экран, упорядочивает данные в алфавитном порядке.

Решение

1. Постановка задачи

Составить программу, которая сохраняет данные о студентах (фамилия, имя, отчество, номер группы) в файл, считывает данные из файла и выводит на экран, упорядочивает данные в алфавитном порядке.

2. Алгоритм решения задачи

Действие 1. Выполнять действие 2, действие 3, действие 4, действие 5 в бесконечном цикле.

Действие 2. Если избранное действие 2, тогда

Действие 2.1. Определить массив структур `sd`.

Действие 2.2. Ввести значение полей структуры.

Действие 2.3. Ввести имя файла.

Действие 2.4. Добавить данные к существующему файлу или создать новый файл.

Действие 3. Если избранное действие 3, тогда

Действие 3.1. Ввести имя файла.

Действие 3.2. Если файл открывается перейти к действию 3.3, иначе к главному меню.

Действие 3.3. Определить размер массива структур `sd`.

Действие 3.4. Считать данные из файла в массив структур `sd`.

Действие 3.5. Ввести значение полей массива структур `sd`.

Действие 4. Если избранное действие 4, тогда

Действие 4.1. Ввести имя файла.

Действие 4.2. Если файл открывается перейти к действию 4.3, иначе к главному меню.

Действие 4.3. Определить размер массива структур `sd`.

Действие 4.4. Считать данные из файла в массив структур `sd`.

Действие 4.5. Отсортировать массив структур по фамилии.

Действие 4.6. Вывести элементы массива после сортировки.

Действие 4.7. Спросить у пользователя необходимость сохранения упорядоченных данных, если так, тогда перейти к действию 4.8, иначе к главному меню.

Действие 4.8. Перезаписать файл

Действие 5. Если избранное действие 5, тогда конец работы программы.

3. Текст программы

```
#include <iostream>
#include <cstring>
#include <fstream>
#include <windows.h>
using namespace std;
struct stud{
    char surname[30];
    char name[30];
    char name2[30];
    char grupa[10];};
void enterStruct(stud sd[], int kol){
    for(int i=0; i < kol; i++){
        cout << "\nФамилия: ";cin >> sd[i].surname;
        cout << "Имя: "; cin >> sd[i].name;
        cout << "Отчество: "; cin >> sd[i].name2;
        cout << "Группа: "; cin >> sd[i].grupa; }
}
void fileWrite(char* txt,stud sd[], int kol) {
    ofstream t(txt,ios::binary | ios::app);
    for(int i=0;i<kol;i++){
        t.write((char*)&sd[i], sizeof sd[i]); }
    t.close(); }
void printStruct(stud* sd,int kol){
    for(int i = 0; i < kol; i++){
        cout << "\nФамилия: "<< sd[i].surname;
        cout << "\nИмя: "<< sd[i].name;
```

```

        cout << "\nОтчество: "<<sd[i].name2;
        cout << "\nГруппа:  "<< sd[i].grupa<<"\n";}}

int fileOfSize (char *txt) {
    ifstream t(txt,ios::binary);
    if (!t.is_open()){
        cout << "Файл не может быть открытым !\n";return 0; }
    t.seekg(0,ios_base::end);
    int k = t.tellg()/sizeof (stud);
    t.close();
    return k;}

void readFile(char *txt,stud* sd,int kol) {
    ifstream t(txt,ios::binary );
    for(int i=0;i<kol;i++){
        t.read((char*)&sd[i], sizeof sd[i]); }
    t.close();}

void rewriteFile(char* txt,stud* sd,int kol) {
    ofstream t(txt,ios::binary);
    for(int i=0;i<kol;i++){
        t.write((char*)&sd[i], sizeof sd[i]);}
    t.close();}

void sortStud(stud* sd,int kol){
    stud s;
    for(int i=1;i<kol;i++)
        for(int j=kol-1;j>=i;j--){
            if(stricmp(sd[j].surname, sd[j-1].surname) < 0){
                s = sd[j];
                sd[j]=sd[j-1];
                sd[j-1]=s;
            }
            else if(stricmp(sd[j].surname, sd[j-1].surname) ==0){
                if(stricmp(sd[j].name, sd[j-1].name) < 0){
                    s = sd[j];
                    sd[j]=sd[j-1];

```

```

        sd[j-1]=s;
    }
}}

int main() {
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    char fName[80];
    int k = 0; stud * sd;
    while(1) {
        cout<<"0: Exit\n";
        cout<<"1: Добавить записи в файл\n";
        cout<<"2: Вывести содержимое файла на экран\n";
        cout<<"3: Сортировка данных по фамилии\n ";
        cin >> k;
        switch(k) {
            case 1:
                cout<<"\nВведите количество студентов: ";
                int zn;
                cin>>zn; sd = new stud[zn];
                enterStruct(sd, zn);
                cin.ignore();
                cout << "\nВведите имя файла: ";
                cin.getline(fName, 80);
                fileWrite(fName, sd, zn);
                break;
            case 2:
                cout << "\nВведите имя файла: "; cin.ignore();
                cin.getline(fName, 80);
                zn=fileOfSize(fName);
                if (zn!=0){sd = new stud[zn];
                readFile(fName, sd, zn);
                cout << "\n Содержимое файла: \n";
                printStruct(sd, zn); cin.ignore();}
                break;
            case 3:

```

```
    cout << "\n Сортировка данных по фамилии: \n";
    cout << "\nВведите имя файла: "; cin.ignore();
        cin.getline(fName,80);
    zn=fileOfSize(fName);
    if (zn!=0){sd = new stud[zn];
    readFile(fName,sd,zn);
    sortStud(sd,zn);
    printStruct(sd,zn);
    cout << "\nСохранить отсортированные данные? (Y/N): ";
    char tak;cin>>tak;
    if (tak=='Y' || tak =='y')
        rewriteFile(fName,sd,zn);}
        break;
    case 0:
        exit(0);
        break;
    }}
return 0;
}
```