МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Лабораторная работа № 1

по дисциплине «Информатика»

Структурированный тип

Факультет: АВТФ

Группа: АВТ - 310

Студент: Богомолов Д.А.

Преподаватель: Сагитов Ю.И.

Новосибирск

2014

**Задание**

Определить структурированный тип и набор функций для работы с таблицей записей, реализованной в массиве структур. В перечень функций входят:

* ввод записи таблицы с клавиатуры;
* загрузка и сохранение таблицы в текстовом файле;
* просмотр таблицы;
* сортировка таблицы в порядке возрастания заданного поля;
* поиск в таблице элемента с заданным значением поля или с наиболее близким к нему по значению;
* удаление записи;
* изменение (редактирование) записи;
* вычисление с проверкой и использованием всех значений по заданному условию и формуле (например, общая сумма на всех счетах).

**Вариант**

Название экзамена, дата экзамена, фамилия преподавателя, количество оценок, оценки.

**Текст программы с комментариями**

#define \_CRT\_SECURE\_NO\_WARNINGS

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <iostream>

#include <stdio.h>

using namespace std;

struct date{

int dd,mm,yy;

void setDate(int dd0, int mm0, int yy0){ // встроенная функция установки даты

dd=dd0; mm=mm0; yy=yy0;

cout << dd << " " << mm <<" " << yy << endl;

}

void getDate(){ // встроенная функция чтения даты

printf("\nДень:"); scanf("%d",&dd);

printf("Месяц:"); scanf("%d",&mm);

printf("Год:"); scanf("%d",&yy); // непосредственный доступ по имени поля

}

int cmpDate(date &T){ // встроенная функция сравнения дат

if (yy!=T.yy) return yy-T.yy;

if (mm!=T.mm) return mm-T.mm;

return dd-T.dd;

}

// встроенные функции работы с файлом

void loadDate(FILE \*fd){ fscanf(fd,"%d%d%d",&dd,&mm,&yy); }

void saveDate(FILE \*fd){ fprintf(fd,"%d %d %d ",dd,mm,yy); }

void showDate(){ printf("%02d.%02d.%04d ",dd,mm,yy); }

};

const int NP = 100;

const int N = 100;

struct exam

{

char name[32],profName[32];

date \_date;

int marks[NP];

int np;

void loadMark(FILE \*fd,int i)

{

int temp;

fscanf(fd,"%d",&temp);

marks[i] = temp;

}

void sortMarks()

{

//для упрощения взята сама тупая сортировка, пузырьком

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

for(int j=i;j<np;j++)

if(marks[i]>marks[j])

{

int temp = marks[i];

marks[i] = marks[j];

marks[j] = temp; // LOL

}

}

double GetMiddle()

{

double s =0;

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

{

s+=marks[i];

}

//lcout << s <<" " << np << " \\" << (double)s/(double)np;

return (double)((double)s/(double)np);

}

void GetCountMarks()

{

int one = 0,two = 0, three = 0,four = 0, five = 0;

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

{

if(marks[i]==1) one++;

if(marks[i]==2) two++;

if(marks[i]==3) three++;

if(marks[i]==4) four++;

if(marks[i]==5) five++;

}

cout << "Кол-во колов " << one << endl;

cout << "Кол-во двоек " << two << endl;

cout << "Кол-во троек " << three << endl;

cout << "Кол-во четверок " << four << endl;

cout << "Кол-во пятерок " << five << endl;

}

void saveMarks(FILE \*fd)

{

fprintf(fd, "%d ",np);

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

fprintf(fd, "%d ",marks[i]);

fprintf(fd, "\n");

}

void showMarks()

{

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

cout << marks[i]<< " ";

}

void getMarks(int n)

{

np = n;

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

cin >> marks[i];

}

void showExam()

{

cout << "Имя преподавателя: " << profName << ", имя экзамена: " << name << ", дата экзамена: ";

\_date.showDate();

cout << ", кол-во оценок: " << np << ", оценки: ";

showMarks();

cout << endl;

}

void addExam(char name0[],char profName0[])

{

strcpy(name,name0);

strcpy(profName, profName0);// profName = profName0;

//cout << "You type name exam:" << name0 << " Имя преподавателя:" << profName0 << endl;

\_date.getDate();

cout << "Введи кол-во оценок у преподавателя: ";

int temp;

cin >> temp;

getMarks(temp);

}

void loadExam(FILE \*fd)

{

fscanf(fd,"%s %s",profName,name);

\_date.loadDate(fd); // вызов встроенной функции для поля birth

fscanf(fd,"%d",&np);

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

{

loadMark(fd,i);

}

}

void saveExam(FILE \*fd)

{

fprintf(fd, "%s %s ",name,profName);

\_date.saveDate(fd);

saveMarks(fd);

}

void specFunc(int func)

{

switch(func)

{

case 1:

{

}

break;

default: break;

}

}

int cmpUser(exam &T,int mode){ // встроенная функция сравнения записей

switch (mode){

case 3: return strcmp(name,T.name); // сравнение по имени

case 4: return \_date.cmpDate(T.\_date); // сравнение по дате регистрации

case 2: return np-T.np;

}

cout << "problem" << endl;

return 0;

}

};

struct table

{

exam \_exams[N];

int nn = 0;

int curInFile = 0;

void loadTable(char nm[]){

FILE \*fd=fopen(nm,"rw+");

if (fd==NULL) return;

fscanf(fd,"%d",&curInFile);

cout << "in file " << nn+curInFile << " exams" << endl;

if (nn>=N) return;

for (int i=nn;i<nn+curInFile;i++) \_exams[i].loadExam(fd);

nn+=curInFile;

fclose(fd);

}

void addExam(char name[],char profName[])

{

\_exams[nn].addExam(name, profName);

nn++;

}

void deleteExam(int id)

{

for(int j=id;j<nn-1;j++)

\_exams[j]= \_exams[j+1];

nn--;

}

void editExam(char name[])

{

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

{

if(strcmp(\_exams[i].name, name)==0)

{

cout << "Я нашел вашу запись =3, что хотите отредактировать?\n 1 - имя преподавателя\n2 - имя экзамена,\n3 - изменить дату экзамена\n4-изменить оценки" << endl;

int mode; cin >> mode;

switch(mode)

{

case 1:

{

cout << "Введите имя преподавателя" << endl;

cin >> \_exams[i].profName;

}

break;

case 2:

{

cout << "Введите имя экзамена" << endl;

cin >> \_exams[i].name;

}

break;

case 3:

{

cout << "Введите дату" << endl;

\_exams[i].\_date.getDate();

}

break;

case 4:

{

cout << "Введите кол-во оценок" << endl;

int count; cin >> count;

cout << "Вы молодец, теперь введите ваши оценки через пробел или каждую через enter" << endl;

\_exams[i].getMarks(count);

}

break;

}

break;

}

}

}

void sortMarks(char name[])

{

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

{

if(strcmp(\_exams[i].name, name)==0)

{

\_exams[i].sortMarks();

break;

}

}

}

void sortTable(int mode)

{

switch(mode)

{

case 1:

{

//для упрощения взята сама тупая сортировка, пузырьком

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

for(int j=i;j<nn;j++)

if(\_exams[i].np>\_exams[j].np)

{

exam temp = \_exams[i];

\_exams[i] = \_exams[j];

\_exams[j] = temp; // LOL

}

}

break;

case 3:

{

int i,j,k;

for (i=0;i<nn;i++){ // вызов встроенной функции сравнения записей

for(j=k=i; j<nn;j++)

if (\_exams[j].cmpUser(\_exams[k], 3)<0) k=j;

exam cc=\_exams[i]; \_exams[i]=\_exams[k]; \_exams[k]=cc;

}

}

break;

case 4:

{

int i,j,k;

for (i=0;i<nn;i++){ // вызов встроенной функции сравнения записей

for(j=k=i; j<nn;j++)

if (\_exams[j].cmpUser(\_exams[k], 4)<0) k=j;

exam cc=\_exams[i]; \_exams[i]=\_exams[k]; \_exams[k]=cc;

}

}

break;

}

}

void FindExam(int mode)

{

switch(mode)

{

case 1:

{

cout << "input surname" << endl;

char name[32]; cin >> name;

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

{

if(strcmp(\_exams[i].profName,name)==0)

\_exams[i].showExam();

}

}

break;

case 2:

{

cout << "input name exam" << endl;

char name[32]; cin >> name;

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

{

if(strcmp(\_exams[i].name,name)==0)

\_exams[i].showExam();

}

}

break;

case 3:

{

cout << "input id" << endl;

int temp; cin >> temp;

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

{

if(temp==i) \_exams[i].showExam();

}

}

break;

}

}

void specFunc(int func)

{

switch(func)

{

case 1:

{

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

{

\_exams[i].showExam();

cout << "Среднее значение оценок " << \_exams[i].GetMiddle() << endl;

}

}

break;

case 2:

{

cout << "Статистика оценок(кол-во)" << endl;

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

{

\_exams[i].showExam();

\_exams[i].GetCountMarks();

}

}

break;

default: break;

}

}

void viewExams()

{

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

{

printf("%d %s",i,": ");

\_exams[i].showExam();

}

}

void saveTable(char nm[])

{

FILE \*fd=fopen(nm,"wb+");

if (fd==NULL) return;

cout << " Write " << nn << " exams into file. Begin" << endl;

fprintf(fd, "%d\n",nn);

for (int j=0;j<nn;j++) \_exams[j].saveExam(fd);

fclose(fd);

}

};

table TT;

int main(int argc, const char \* argv[])

{

while(true)

{

cout << "\na(dd),v(iew),l(oad),E(dit),s(ave),F(ind),d(elete),S(ort),c(ompare),f(unctions),e(xit)\nwhat to do" << endl;

char t;

cin >> t;

switch (t) {

case 'a':

{

cout << "a(dd) mode" << endl;

cout << "Введите название экзамена и фамилию преподавателя" << endl;

char name[32],name2[32];

cin >> name >> name2;

TT.addExam(name, name2);

}

break;

case 'v':

{

cout << "v(iew) mode" << endl;

TT.viewExams();

break;

}

break;

case 'c':

{

cout << "c(ompare) mode" << endl;

cout << "";

break;

}

break;

case 'l':

{

cout << "l(oad) mode" << endl;

TT.loadTable("file.txt");

}

break;

case 'E':

{

cout << "E(dit) mode " << endl;

cout << "Вы можете изменить поле экзамена по имени,введите его" << endl;

char name[32]; cin >> name;

TT.editExam(name);

}

break;

case 'f':

{

cout << "special final f(unctions)\n1 - Средняя оценка за экзамен\n2 - общее кол - во оценок " << endl;

int mode; cin >> mode;

switch (mode) {

case 1:

{

TT.specFunc(1);

}

break;

case 2:

{

TT.specFunc(2);

}

break;

default:

break;

}

}

break;

case 'F':

{

cout << "search 1-surname 2 -name exam 3 - id" << endl;

int temp; cin >> temp;

switch (temp) {

case 1:

{

TT.FindExam(1);

}

break;

case 2:

{

TT.FindExam(2);

}

break;

case 3:

{

TT.FindExam(3);

}

break;

default:

break;

}

}

break;

case 'S':

{

cout << "S(ort) mode " << endl;

cout << "Сортировка по:\n 1 - кол-ву оценок\n 2 - внутри экзамена будут сортиоваться оценки\n 3 - по имени \n 4 - по дате" << endl;

int mode; cin >> mode;

switch (mode) {

case 1:

{

TT.sortTable(1);

}

break;

case 2:

{

cout << "Введите имя экзамена" << endl;

char name[32]; cin >> name;

TT.sortMarks(name);

}

break;

case 3:

{

TT.sortTable(3);

}

break;

case 4:

{

TT.sortTable(4);

}

break;

default:

break;

}

}

break;

case 'd':

{

cout << "d(elete) mode" << endl;

cout << "Введите название экзамена для удаления" << endl;

int id;

cin >>id;

TT.deleteExam(id);

}

break;

case 's':

{

cout << "s(ave) mode" << endl;

cout << "r(ewrite) or w(rite) file with another name?" << endl;

char c; cin >> c;

switch (c) {

case 'r':

{

TT.saveTable("file.txt");

}

break;

case 'a':

{

cout << "Введите имя файла,на английском" << endl;

char name[32]; cin >> name;

TT.saveTable(name);

}

default:

break;

}

//TT.saveTable("file.txt");

}

break;

case 'e':

{

cout << "Press enter to exit:" << endl;

getchar();

getchar();

return 0;

}

break;

default:

cout << "non implemented mode" << endl;

break;

}

}

getchar();

return 0;

}

**Пример работы программы**

**a(dd),v(iew),l(oad),E(dit),s(ave),d(elete),S(ort),c(ompare),f(unctions),e(xit)**

**what to do**

a

**a(dd) mode**

**Введите название экзамена и фамилию преподавателя**

Математика Резников

**День:**24

**Месяц:**01

**Год:**2014

**Введи кол-во оценок у преподавателя:** 5

5 4 4 3 5

**a(dd),v(iew),l(oad),E(dit),s(ave),d(elete),S(ort),c(ompare),f(unctions),e(xit)**

**what to do**

a

**a(dd),v(iew),l(oad),E(dit),s(ave),d(elete),S(ort),c(ompare),f(unctions),e(xit)**

**what to do**

**a(dd) mode**

**Введите название экзамена и фамилию преподавателя**

ЛАиАГ Судоплатов

**День:**14

**Месяц:**01

**Год:**2014

**Введи кол-во оценок у преподавателя:** 4

3

4 5 3

**a(dd),v(iew),l(oad),E(dit),s(ave),d(elete),S(ort),c(ompare),f(unctions),e(xit)**

**what to do**

v

**a(dd),v(iew),l(oad),E(dit),s(ave),d(elete),S(ort),c(ompare),f(unctions),e(xit)**

**what to do**

**v(iew) mode**

**Имя преподавателя: Резников, имя экзамена: Математика, дата экзамена: 24.01.2014 , кол-во оценок: 5, оценки: 5 4 4 3 5**

**Имя преподавателя: Судоплатов, имя экзамена: ЛаАиАГ, дата экзамена: 14.01.2014 , кол-во оценок: 4, оценки: 3 4 5 3**

**Выводы**Использование структур в программах облегчает понимание и увеличивает скорость написания программы в целом. Структуры служат аналогом таблицы данных в этой ЛР.