МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

## Факультет информационных технологий и робототехники

Кафедра программного обеспечения вычислительной техники

и автоматизированных систем

**Отчет по лабораторной работе № 7**

по дисциплине: ”Основы алгоритмизации и программирования”

на тему: ***”Структуры, объединения,перечисления ”***

Вариант 3

Выполнил**:** студент группы *10701118* Воробей И.А.

Приняла**:** ст.пр. Борисова И.М

Минск 2019

# Лабораторная работа № 7. *Структуры, объединения, перечисления*.

**Задание 1. Структуры.**

Описать структуру с именем STUDENT, содержащую следующие поля:

|  |  |
| --- | --- |
| • | фамилия и инициалы; |
| • | номер группы; |

|  |  |
| --- | --- |
| • | успеваемость (массив из пяти элементов). |

Написать программу, выполняющую следующие действия:

|  |  |
| --- | --- |
| • | ввод с клавиатуры данных в массив, состоящий из десяти структур типа STUDENT; записи должны быть упорядочены по алфавиту; |
| • | вывод на дисплей фамилий и номеров групп для всех студентов, имеющих хотя бы одну неудовлетворительную оценку; |

|  |  |
| --- | --- |
| • | если таких студентов нет, вывести соответствующее сообщение. |

 #include <stdio.h>

#include<iostream>

using namespace std;

const int n = 10;

struct STUDENT

{

char name[20];

int numofgr;

int performance[5];

};

void sort(STUDENT ms[n])

{

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

{

for (int j = i + 1; j<n; j++)

{

if (strcmp(ms[i].name, ms[j].name)>0)

{

STUDENT c = ms[i];

ms[i] = ms[j];

ms[j] = c;

}

}

}

}

void output(STUDENT ms[n])

{

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

{

cout << "name : "; cout << ms[i].name << endl;

cout << "number of grup : "; cout << ms[i].numofgr << endl;

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

{

cout << ms[i].performance[y] << endl;

}

cout << endl;

}

}

void blakclist(STUDENT ms[n]);

void main()

{

STUDENT ms[n];

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

{

cout << "Enter name : "; cin >> ms[i].name;

cout << "Enter nuber of groap : "; cin >> ms[i].numofgr;

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

{

cout << "Enter mark["; cout << j; cout << "] : "; cin >> ms[i].performance[j];

}

cout << endl;

}

output(ms);

cout << endl;

sort(ms);

output(ms);

cout << endl;

blakclist(ms);

system("pause");

}

void blakclist(STUDENT ms[n])

{

bool fl = 0;

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

{

bool flag = 0;

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

{

if (ms[i].performance[j]<4)

flag = fl = 1;

}

if (flag)

{

cout << ms[i].name << endl;

cout << ms[i].numofgr << endl;

}

}

if (!fl)

cout << "Students with bad marks do not exist." << endl;

}

**Задание 2. Объединения и перечисления.**

Описать структуру для хранения следующей информации (при необходимости используйте перечисления):

данные о клиентах:

|  |  |
| --- | --- |
| • | студенты (№ зачетки, фамилия, пол(м/ж), средний балл); |
| • | преподаватели (табельный номер, фамилия, должность (4 должности)); |

|  |  |
| --- | --- |
| • | слушатели (№ паспорта, фамилия, специальность (4 специальности)). |

Создать массив из 5 таких структур.

Написать программу, выполняющую следующие действия:

|  |  |
| --- | --- |
| • | добавление записи в массив; если в массиве нет места - вывести соответствующее сообщение; |
| • | удаление записи из массива; если такой записи нет - вывести соответствующее сообщение; |

|  |  |
| --- | --- |
| • | вывод всех записей массива. |

#include<iostream>

#include<stdio.h>

using namespace std;

const int n = 5;

union KLIENT{

struct st{

int numofz;

char famils[20];

enum sex{

WOMEN,

MEN

};

int midlmark;

};

struct tch{

int tabel;

char familt[20];

enum position{

YONGER,

MIDL,

HIGHER,

DIRECTOR

};

};

struct lrs{

int pasport;

char famill[20];

enum specialty{

DRIVER,

BILDER,

COOK,

KILLER

};

};

};

void output(KLIENT::st kls[n],

KLIENT::st::sex sx[n],

KLIENT::tch klt[n],

KLIENT::tch::position pos[n],

KLIENT::lrs kll[n],

KLIENT::lrs::specialty spec[n], int x, int y, int z)

{

for (int j = 0; j < x; j++){

cout << "Name of record book : "<<kls[j].numofz << endl;

cout << "Sername of student : "<<kls[j].famils << endl;

switch (sx[j]){

case 0:

cout << "Sex of student : WOMEN\n\n";

break;

case 1:

cout << "Sex of student : MEN\n\n";

break;

}

}

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

{

cout <<"Personnel number : " <<klt[j].tabel << endl;

cout <<"Sername of teacher : " <<klt[j].familt << endl;

switch (pos[j])

{

case 0:

cout << "Position of teacher : YUNGER\n\n";

break;

case 1:

cout << "Position of teacher : MIDL\n\n";

break;

case 2:

cout << "Position of teacher : HIGHER\n\n";

break;

case 3:

cout << "Position of teacher : DIRECTOR\n\n";

break;

}

}

for (int j = 0; j < z; j++){

cout << "Pasport number of listener : "<<kll[j].pasport << endl;

cout <<"Sername of listener : "<< kll[j].famill << endl;

switch (spec[j])

{

case 0:

cout << "Profession of listener : DRIVER\n\n";

break;

case 1:

cout << "Profession of listener : BILDER\n\n";

break;

case 2:

cout << "Profession of listener : COOK\n\n";

break;

case 3:

cout << "Profession of listener : KILLER\n\n";

break;

}

cout <<"profession : "<< spec[j] << endl;

}

}

void main()

{

char sername[20];

int c,c1,x=0,y=0,z=0;

KLIENT::st kls[n];

KLIENT::st::sex sx[n];

KLIENT::tch klt[n];

KLIENT::tch::position pos[n];

KLIENT::lrs kll[n];

KLIENT::lrs::specialty spec[n];

while (1)

{

cout << "Enter 1-STUDENT 2-TEACHER 3-LISTENER to create record,\n4-to del a record, 5-to view all records : ";

cin >> c1; cout << "\n";

switch (c1)

{

case 1:

if (x < n)

{

cout << "Enter number of record book : "; cin >> kls[x].numofz;

cout << "Enter name of student : "; cin >> kls[x].famils;

cout << "Enter 1-MEN or 0-WOMEN : ";

cin >> c;

if (c)

sx[x] = KLIENT::st::MEN;

else

sx[x] = KLIENT::st::WOMEN;

x++;

}

else

{

cout << "Empty places have been occupied!\n ";

break;

}

break;

case 2:

if (y < n)

{

cout << "Enter personnel number of teacher : "; cin >> klt[y].tabel;

cout << "Enter sername of teacher : "; cin >> klt[y].familt;

cout << "Enter 0-YOUNGER, 1-MIDLE, 2-HUGHER, 3-DIRECTOR : ";

cin >> c;

switch (c)

{

case 0:

pos[y] = KLIENT::tch::YONGER;

break;

case 1:

pos[y] = KLIENT::tch::MIDL;

break;

case 2:

pos[y] = KLIENT::tch::HIGHER;

break;

case 3:

pos[y] = KLIENT::tch::DIRECTOR;

break;

}

y++;

}

else

{

cout << "Empty places have been occupied!\n ";

break;

}

break;

case 3:

if (z < n)

{

cout << "Enter number of pasport : "; cin >> kll[z].pasport;

cout << "Enter sername of listener : "; cin >> kll[z].famill;

cout << "Enter 0-DRIVER, 1-BILDER, 2-COOK, 3-KILLER : ";

cin >> c;

switch (c)

{

case 0:

spec[z] = KLIENT::lrs::DRIVER;

break;

case 1:

spec[z] = KLIENT::lrs::BILDER;

break;

case 2:

spec[z] = KLIENT::lrs::COOK;

break;

case 3:

spec[z] = KLIENT::lrs::KILLER;

break;

}

z++;

}

else

{

cout << "Empty places have been occupied!\n ";

break;

}

break;

case 4:

cout << "Enter a sername, why you want to del :";

cin >> sername;

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

{

if (strcmp(kls[j].famils, sername) == 0)

{

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

{

kls[i]=kls[i+1];

}

x--;

}

}

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

{

if (strcmp(klt[j].familt, sername) == 0)

{

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

{

kls[i] = kls[i + 1];

}

y--;

}

}

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

{

if (strcmp(kll[j].famill, sername) == 0)

{

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

{

kls[i] = kls[i + 1];

}

z--;

}

}

break;

case 5:

output(kls, sx, klt, pos, kll, spec,x,y,z);

}

}

system("pause");

}

**Контрольные вопросы.**

1. Опишите битовое поле для хранения следующей информации (размер задавать по минимуму):

|  |  |
| --- | --- |
| ◆ | номер поезда (4 десятичных цифры); |
| ◆ | время отправления; |

|  |  |
| --- | --- |
| ◆ | тип поезда (4 типа). |

#include <stdio.h>

#include<iostream>

using namespace std;

struct trains{

short number : 14;

short hh : 4;

short mm : 6;

short type : 3;

};

void main()

{

trains tr;

tr.number = 9999;

tr.hh = 23;

tr.mm = 59;

tr.type = 1;

char c;

int b;

short t;

cout << sizeof(c);

cout << sizeof(b);

cout << sizeof(t);

cout << sizeof(tr);

system("pause");

}