Министерство образования и науки Кыргызской Республики

Кыргызский государственный технический университет

им. И. Раззакова

Факультет информационных технологий

Кафедра «Программное обеспечение компьютерных систем»

Направление:710400 «Программная инженерия»

ОТЧЕТ

По дисциплине: «Функционально-ориентированное проектирование»

Лабораторная работа №5

Тема**: «**Структуры**»**

Выполнила: студентка группы

ПИ(б)-5-19 Ажиходжоева Каныкей

Проверил: доцент Искаков Р.Т

Бишкек – 2020

1. Описать структуру с именем Student, содержащую следующие поля: name, group, ses(успеваемость - массив из 5 элементов). Ввести с клавиатуры данные в массив stud1, состоящий из 10 структур типа student. Записи упорядочить по алфавиту

вывести на экран фамилии и номера групп для всех студентов, имеющих оценки 4 и 5, если таких нет, то вывести соответствующее сообщение.

• Блок-схема



Ввод с консоли. Вывод.

 Сортировка.

Поиск(2).

Главная функция (Main).

• Код программы

#include <iostream>

#include <iomanip>

using namespace std;

struct Student {

char name[30];

char group[10];

int ses[5];

};

void input(Student\*, int, int);

void output(Student\*, int, int);

void sort(Student\*, int, int);

void best(Student\*, int, int);

void good(Student\*, int, int);

int main() {

setlocale(LC\_ALL, "rus");

int x, n;

cout << "Введите кол-во студентов : ";

cin >> n;

const int m = 5;

Student\* stud1 = new Student[n];

restart:

cout << "-----------------------------------------" << endl;

cout << "| Главное Меню: |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 1.Ввод студентов. |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 2.Вывод данных. |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 3.Сортировка. |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 4.Поиск Отличников. |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 5.Поиск Ударников. |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 6.Выход. |" << endl;

cout << "-----------------------------------------" << endl;

cout << endl;

cout << "Введите операцию : ";

cin >> x;

cout << endl;

switch (x) {

case 1:

input(stud1, n, m);

break;

case 2:

output(stud1, n, m);

break;

case 3:

sort(stud1, n, m);

break;

case 4:

best(stud1, n, m);

break;

case 5:

good(stud1, n, m);

break;

case 6:

exit(1);

default:

cout << "\tОперация НЕ НАЙДЕНА!\n";

break;

}

goto restart;

}

void input(Student\* stud1, int n, int m) {

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

cout << "\n\tСтудент №" << i + 1 << endl

<< "\tИмя: ";

cin >> stud1[i].name;

cout << "Группа: ";

cin >> stud1[i].group;

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

res:

cout << "Оцкенка по предмету №" << j + 1 << ": ";

cin >> stud1[i].ses[j];

if (stud1[i].ses[j] < 2 || stud1[i].ses[j] > 5) {

cout << "Такой оценки нет! Введите заново." << endl;

goto res;

}

}

cout << endl;

}

}

void output(Student\* stud1, int n, int m) {

cout << "\n\t№\tИМЯ\tГРУППА\t\tОЦЕНКИ\n";

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

cout << "\t" << i + 1 << "\t" << stud1[i].name

<< "\t" << stud1[i].group

<< "\t\t";

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

cout << stud1[i].ses[j] << " ";

}

cout << endl;

}

cout << endl;

}

void sort(Student\* stud1, int n, int m) {

Student temp;

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

for (int j = 0; j < n - 1; j++) {

if (strcmp(stud1[j].name, stud1[j + 1].name) > 0) {

temp = stud1[j];

stud1[j] = stud1[j + 1];

stud1[j + 1] = temp;

}

}

}

}

void best(Student\* stud1, int n, int m) {

int temp, temp1 = 0;

cout << "\n\tОТЛИЧНИКИ!\n"

<< "\t№\tИМЯ\tГРУППА\t\tОЦЕНКИ\n" << endl;

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

temp = 0;

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

if (stud1[i].ses[j] == 5) {

temp++;

}

}

if (temp == 5) {

cout << "\t" << i + 1 << "\t" << stud1[i].name

<< "\t" << stud1[i].group

<< "\t\t";

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

cout << stud1[i].ses[j] << " ";

}

cout << endl;

temp1++;

}

}

if (temp1 == 0) {

cout << "ОТЛИЧНИКИ НЕ НАЙДЕНЫ!\n\n";

}

}

void good(Student\* stud1, int n, int m) {

int temp1, temp2, temp3;

temp2 = 0;

cout << "\n\tУДАРНИКИ"

<< "\t№\tИМЯ\tГРУППА\t\tОЦЕНКИ\n" << endl;

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

temp1 = temp3 = 0;

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

if (stud1[i].ses[j] == 4)

temp1++;

if (stud1[i].ses[j] == 5)

temp3++;

}

if (temp1 + temp3 == 5 && temp1 != 0) {

cout << "\t" << i + 1 << "\t" << stud1[i].name

<< "\t" << stud1[i].group

<< "\t\t";

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

cout << stud1[i].ses[j] << " ";

}

cout << endl;

temp2++;

}

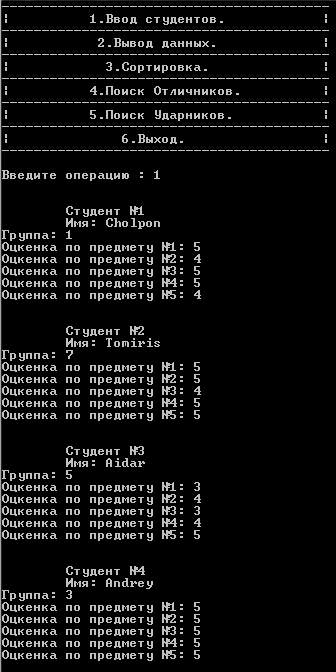
}

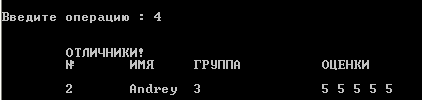
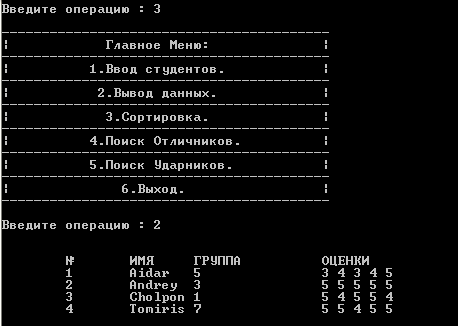
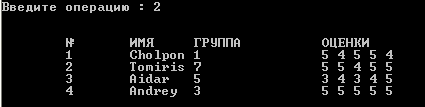
if (temp2 == 0)

cout << "УДАРНИКИ НЕ НАЙДЕНЫ!\n\n";

}

•Результат вычислений





1. Описать структуру с именем Train, содержащую следующие поля: nazn(название пункта назначения), num(номер поезда), time(время отправления). Ввести с клавиатуры данные в массив rasp, состоящий из 6 структур типа Train. Записи упорядочить по номерам поездов. Вывести на экран информацию о поезде, номер которого введен с клавиатуры, если такого поезда нет, то вывести соответствующее сообщение.

• Блок-схема

Главная функция (Main).





• Код программы

#include<iostream>

#include <iomanip>

using namespace std;

struct Vr {

int h;

int m;

};

struct Train {

string nazn;

int num;

Vr time;

};

void input(Train\*, int);

void output(Train\*, int);

void sort(Train\*, int);

void och(Train\*, int);

int main() {

setlocale(LC\_ALL, "rus");

int n, x;

cout << "Введите количество поездов:";

cin >> n;

Train\* tr1 = new Train[n];

metka:

cout << "-----------------------------------------" << endl;

cout << "| Главное Меню: |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 1.Ввод поездов. |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 2.Вывод данных. |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 3.Сортировка. |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 4.Время отпрвления поезда. |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 5.Выход. |" << endl;

cout << "-----------------------------------------" << endl;

cout << endl;

cout << "Введите номер опреации:";

cin >> x;

switch (x)

{

case 1:

input(tr1, n);

goto metka; break;

case 2:

output(tr1, n);

goto metka; break;

case 3:

sort(tr1, n);

goto metka; break;

case 4:

och(tr1, n);

goto metka; break;

case 5:

exit; break;

default:

cout << "Операция не найдена." << endl;

goto metka; break;

}

return 0;

}

void input(Train\* train, int n) {

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

{

cout << "Место назначения: ";

cin >> train[i].nazn;

cout << "Поезд № : ";

cin >> train[i].num;

t:

cout << "Введитие часы(0:23):";

cin >> train[i].time.h;

if (train[i].time.h < 0 || train[i].time.h > 23) {

cout << "Неправильный формат времени.Введите заново." << endl;

goto t;

}

cout << "Введитие минуты(0:59):";

cin >> train[i].time.m;

if (train[i].time.m < 0 || train[i].time.m > 59) {

cout << "Неправильный формат времени.Введите заново." << endl;

goto t;

}

}

}

void output(Train\* train, int n)

{

cout << "-----------------------------------------" << endl;

cout << "| " << "Место назначения" << " " << "Поезд №" << " " << "Время" << " |" << endl;

cout << "-----------------------------------------" << endl;

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

{

cout << " " << train[i].nazn << " " << train[i].num << " " << train[i].time.h << ":" << train[i].time.m;

cout << endl;

}

cout << endl;

}

void sort(Train\* train, int n) {

Train temp;

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

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

if (train[i].num > train[i + 1].num)

{

temp = train[i];

train[i] = train[i + 1];

train[i + 1] = temp;

}

}

}

}

void och(Train\* train, int n) {

int p;

cout << "Введите номер поезда:";

cin >> p;

bool f = false;

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

if (train[i].num == p) {

f = true;

cout << "-----------------------------------------" << endl;

cout << "| " << "Место назначения" << " " << "Поезд №" << " " << "Время" << " |" << endl;

cout << "-----------------------------------------" << endl;

cout << " " << train[i].nazn << " " << train[i].num << " " << train[i].time.h << ":" << train[i].time.m << endl;

break;

}

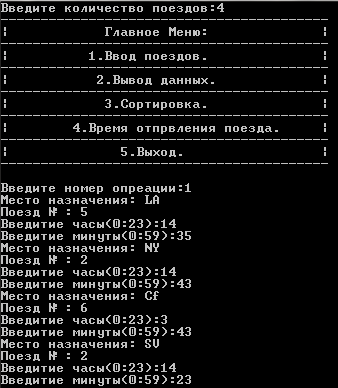
if (f = false) cout << "Нет такого номера поезда";

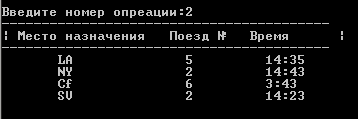
cout << endl;

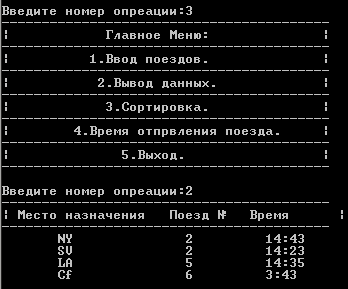
}

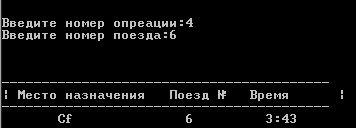
}

• Результат вычислений









1. Описать структуру с именем Train, содержащую следующие поля: nazn(название пункта назначения), num(номер поезда), time(время отправления). Ввести с клавиатуры данные в массив rasp, состоящий из 6 структур типа Train. Записи упорядочить по номерам поездов. Вывести на экран информацию о поезде, номер которого введен с клавиатуры, если такого поезда нет, то вывести соответствующее сообщение.

•Блок-схема

Главная функция(Main)



• Код программы

# include <iostream>

# include < iomanip >

# include <Windows.h>

using namespace std;

struct Worker {

string name;

string pos;

int year;

};

void input(Worker\*, int);

void output(Worker\*, int);

void sort(Worker\*, int);

void vyvod(Worker\*, int);

int main() {

setlocale(LC\_ALL, "RUS");

int n, x;

cout << "Введите количество работников:";

cin >> n;

Worker\* w = new Worker[n];

metka:

cout << "-----------------------------------------" << endl;

cout << "| Главное Меню: |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 1.Ввод работников. |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 2.Вывод данных. |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 3.Сортировка. |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 4.Поиск по стажу работы. |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 5.Выход. |" << endl;

cout << "-----------------------------------------" << endl;

cout << endl;

cout << "Введите номер опреации:";

cin >> x;

switch (x)

{

case 1:

input(w, n);

goto metka; break;

case 2:

output(w, n);

goto metka; break;

case 3:

sort(w, n);

goto metka; break;

case 4:

vyvod(w, n);

goto metka; break;

case 5:

exit; break;

default:

cout << "Операция не найдена." << endl;

goto metka; break;

}

return 0;

}

void input(Worker\* worker, int n)

{

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

cout << "Работник №" << i + 1 << ":";

cin >> worker[i].name;

cout << "Должность:";

cin >> worker[i].pos;

t:

cout << "Год:";

cin >> worker[i].year;

if (worker[i].year < 1950 || worker[i].year >2020) {

cout << "Введите корректный год." << endl;

goto t;

}

cout << endl;

}

}

void output(Worker\* worker, int n)

{

cout << "-----------------------------------------" << endl;

cout << "| " << "Имя:" << " " << "Должность:" << " " << "Год:" << " |" << endl;

cout << "-----------------------------------------" << endl;

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

cout << "|" << worker[i].name << " " << worker[i].pos << " " << worker[i].year << " |" << endl;

}

cout << "-----------------------------------------" << endl;

}

void sort(Worker\* worker, int n) {

Worker temp;

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

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

if (worker[i].year > worker[i + 1].year)

{

temp = worker[i];

worker[i] = worker[i + 1];

worker[i + 1] = temp;

}

}

}

}

void vyvod(Worker\* worker, int n)

{

int p;

int g;

cout << "Введите стаж работы :";

cin >> p;

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

g = 2020 - worker[i].year;

if (g >= p) {

cout << "-----------------------------------------" << endl;

cout << "| " << "Имя:" << " " << "Должность:" << " " << "Год:" << " |" << endl;

cout << "-----------------------------------------" << endl;

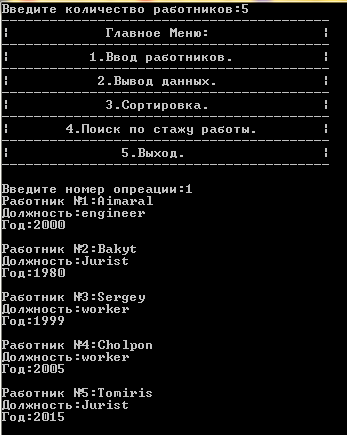
cout << "|" << worker[i].name << " " << worker[i].pos << " " << worker[i].year << " |" << endl;

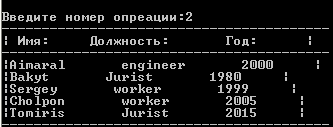
}

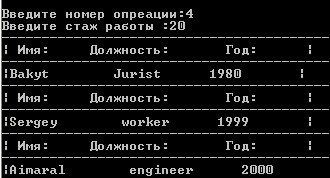
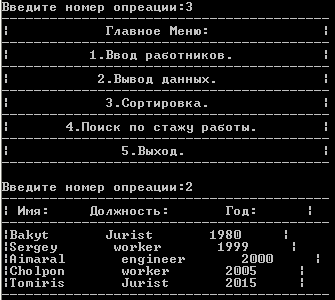
}

}

• Результат вычислений







1. Создайте структуру с именем time. Три ее поля, имеющие тип int, будут называться hours, minutes и seconds. Напишите программу, которая просит пользователя ввести время в формате часы, минуты, секунды. Можно запрашивать на ввод как три значения сразу, так и выводить для каждой величины отдельное приглашение. Программа должна хранить время в структурной переменной типа time и выводить количество секунд в введенном времени.

• Блок-схема



• Код программы

#include <iostream>

#include <iomanip>

using namespace std;

struct Time {

int hours;

int minutes;

int seconds;

};

int s(Time);

int main() {

setlocale(LC\_ALL, "rus");

Time t;

int total;

h:

cout << "Введите часы (0:23) :";

cin >> t.hours;

if (t.hours < 0 || t.hours > 23) {

cout << "Неправильный формат часов.Введите заново." << endl;

goto h;

}

m:

cout << "Введите минуты (0:59) :";

cin >> t.minutes;

if (t.minutes < 0 || t.minutes > 59) {

cout << "Неправильный формат минут.Введите заново." << endl;

goto m;

}

s:

cout << "Введите секунды (0:23) :";

cin >> t.seconds;

if (t.seconds < 0 || t.seconds > 59) {

cout << "Неправильный формат секунд.Введите заново." << endl;

goto s;

}

cout << "В введеном времени " << s(t) << " секунд" << endl;

system("pause");

return 0;

}

int s(Time time) {

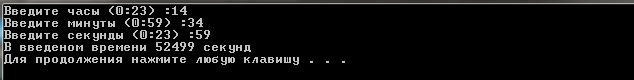
int total;

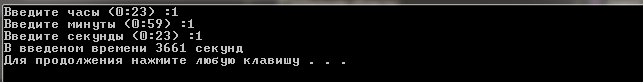
total = (time.hours \* 60\*60) + (time.minutes \* 60) + time.seconds;

return total;

}

• Результат вычислений





1. Используя структуру time из предыд. упражнения, напишите программу, которая получает от пользователя два значения времени в формате 23:59:59, сохраняет их в переменных типа struct time, затем переводит оба значения в секунды, складывает их, переводит сумму в исходный формат, сохраняет его в переменной типа time и выводит полученный результат на экран в формате 23:59:59.

• Блок-схема

Главная функция (Main). Ввод данных.



Функция подсчета секунд и перевода их в формат (23:59:59).

• Код программы

#include <iostream>

using namespace std;

struct time {

int hours;

int minutes;

int seconds;

};

void input(time\*);

int output1(time\*);

void output2(time\*, int&, int);

int main() {

setlocale(LC\_ALL, "rus");

int x, sec1, sec2, sec;

time t1;

time t2;

time time;

int d = 0;

cout << "Ввод времени №1:" << endl;

input(&t1);

sec1 = output1(&t1);

cout << "\n" << t1.hours << ":" << t1.minutes << ":" << t1.seconds << " = " << sec1 << " секунд" << endl;

cout << "Ввод времени №2 :" << endl;

input(&t2);

sec2 = output1(&t2);

cout << "\n" << t2.hours << ":" << t2.minutes << ":" << t2.seconds << " = " << sec2 << " секунд\n";

cout << "Сумма = " << sec1 + sec2 << endl;

sec = sec1 + sec2;

output2(&time, d, sec);

if (d) {

cout << "\n1 день и " << time.hours << ":" << time.minutes << ":" << time.seconds << "\n\n\n";

}

else {

cout << "\n" << time.hours << ":" << time.minutes << ":" << time.seconds << "\n\n\n";

}

system("pause");

return 0;

}

void input(time\* time) {

t:

cout << "Введите часы (0:23) : ";

cin >> time->hours;

if (time->hours < 0 or time->hours > 23) {

cout << "Неправильный формат часов.Введите заново." << endl;

goto t;

}

m:

cout << "Введите минуты (0:59) : ";

cin >> time->minutes;

if (time->minutes < 0 or time->minutes > 59) {

cout << "Неправильный формат минут.Введите заново." << endl;

goto m;

}

s:

cout << "Введите секунды (0:59) : ";

cin >> time->seconds;

if (time->seconds < 0 or time->seconds > 59) {

cout << "Неправильный формат секунд.Введите заново." << endl;

goto s;

}

}

int output1(time\* time) {

int a;

a = time->hours \* 60 \* 60 + time->minutes \* 60 + time->seconds;

return a;

}

void output2(time\* time, int& d, int a) {

int hour, sec, min;

if (a >= (24 \* 60 \* 60)) {

d = 1;

a = a - (24 \* 60 \* 60);

}

hour = a / 3600;

min = (a - hour \* 3600) / 60;

sec = a - hour \* 3600 - min \* 60;

time->hours = hour;

time->minutes = min;

time->seconds = sec;

}

• Результат вычислений

