

Задание:

Придумать и кратко описать некоторую прикладную задачу для решения которой обрабатываемые объекты разумно представлять в виде структур (не менее 5 шт. + связи); Выделить набор значимых свойств этих объектов и разработать набор соответствующих структур языка C для их представления; Разработать набор функций на языке C для работы с этими структурами; В main продемонстрировать работу со статическими и динамическими переменными этих типов.

***При выполнении лабораторной работы использование Git + github обязательно.***

Решение:

Предметная область:

Необходимо составить модель учебного класса. В ней будут связи между частями.

Учитель обладает ФИО. Урок содержит название и учителя, который ведет этот урок. Оценка состоит только из значения. У ученика есть ФИО, набор уроков и соответствующих оценок. Класс содержит название класса, год обучения и набор учеников.

**Структуры**

1. **Учитель (U)**

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | F | char(15) | Фамилия |
| 2 | I | char(15) | Имя |
| 3 | o | char(15) | Отчество |

1. **Предметы (P)**

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | Name | char(50) | Название предмета |
| 2 | **Teacher** | **U Teach** | **Учитель для каждого предмета** |

1. **Ученик (Uch)**

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | F | char(15) | Фамилия |
| 2 | I | char(15) | Имя |
| 3 | o | char(15) | Отчество |
| 4 | **Mas\_Less** | **P mas\_p[20]** | **Массив уроков для каждого ученика** |
| 5 | **Mas\_Marks** | **O mas\_o[20]** | **Массив оценок для каждого ученика** |

1. **Оценки (O)**

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | Value | Integer | Оценка |

1. **Класс (С)**

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | Name | char(10) | Название класса |
| 2 | Year | char(4) | Год обучения |
| 3 | **Mas\_stud** | **Uch mas\_uch[32]** | **Массив учеников** |

*Жирным указаны другие структуры*

1. Лучшие ученики в классе
2. Худшие ученики в классе

Связи:

Оценки

Учителя

Предметы

Ученики

класс

Файл Teacher.h

#pragma once

#include <iostream>

#include <stdio.h>

#include <string>

#include <locale.h>

using namespace std;

// Структура Учитель

struct Teacher

{

string Fam = ""; // Фамилия

string Name = ""; // Имя

string Otch = ""; // Отчество

};

// Инициализация структуры Учитель

Teacher init\_t(string Fam, string Name, string Otch);

// Ввод учителя, передаем адрес, где будет храниться информация

Teacher inp\_Teach();

// Ввод учителя, передаем адрес, где будет храниться информация

void out\_Teach(Teacher t);

Файл Teacher.cpp

#include "Teacher.h"

// Инициализация структуры Учитель

Teacher init\_t(string Fam, string Name, string Otch)

{

Teacher t;

t.Fam = Fam;

t.Name = Name;

t.Otch = Otch;

return t;

}

// Ввод учителя, передаем адрес, где будет храниться информация

Teacher inp\_Teach()

{

Teacher t;

t = init\_t("", "", "");

cout << "Введите Фамилию учителя: ";

cin >> t.Fam;

cout << "Введите имя учителя: ";

cin >> t.Name;

cout << "Введите отчество учителя: ";

cin >> t.Otch;

return t;

}

// Ввод учителя, передаем адрес, где будет храниться информация

void out\_Teach(Teacher t)

{

cout << "ФИО учителя: " << t.Fam << " " << t.Name << " " << t.Otch << endl;

}

Файл Lesson.h

#pragma once

#include "Teacher.h"

// Структура Уроки

struct Lessons

{

string Name = ""; // Название предмета

Teacher Teach; // Учитель

};

// Инициализация структуры Уроки

Lessons init\_l(string Name\_Less);

// Ввод уроков, передаем адрес, где будет храниться информация и адрес учителя

Lessons inp\_Less(Teacher t);

// Вывод уроков, передаем адрес, где будет храниться информация

void out\_Less(Lessons ls);

Файл Lesson.cpp

#include "Lesson.h"

// Инициализация структуры Уроки

Lessons init\_l(string Name\_Less)

{

Lessons ls;

ls.Name = Name\_Less;

Teacher t;

t = init\_t("", "", "");

ls.Teach = t;

return ls;

}

// Ввод уроков, передаем адрес, где будет храниться информация и адрес учителя

Lessons inp\_Less(Teacher t)

{

Lessons ls;

ls = init\_l("");

cout << "Введите название предмета: ";

cin >> ls.Name;

cout << endl;

ls.Teach = t;

return ls;

}

// Вывод уроков, передаем адрес, где будет храниться информация

void out\_Less(Lessons ls)

{

cout << "Название предмета: " << ls.Name << endl;

out\_Teach(ls.Teach);

}

Файл Mark.h

#pragma once

#include "Lesson.h"

// Структура Оценка

struct Mark

{

int Value = 0; // Оценка

};

// Инициализация структуры Оценка

Mark init\_m(int v);

// Ввод данных в структуру Оценка

Mark inp\_Mark();

// Вывод данных из структуры

void out\_Mark(Mark m);

Файл Mark.cpp

#include "Mark.h"

// Инициализация структуры Оценка

Mark init\_m(int v)

{

Mark m;

m.Value = v;

return m;

}

// Ввод данных в структуру Оценка

Mark inp\_Mark()

{

Mark m;

m = init\_m(0);

string mark;

cout << "Введите оценку: ";

cin >> mark;

m.Value = stoi(mark);

cout << endl;

return m;

}

// Вывод данных из структуры

void out\_Mark(Mark m)

{

cout << "Оценка: " << m.Value << endl;

}

Файл Student.h

#pragma once

#include "Mark.h"

// Структура Ученик

struct Student

{

string Fam = ""; // Фамилия

string Name = ""; // Имя

string Otch = ""; // Отчество

Mark mas\_Marks[20]; // Массив оценок

Lessons mas\_Less[20]; // Массив уроков

int i = 0; // Текущее кол-во уроков

};

// Очищение всех полей структуры Ученик (Student)

Student null\_s();

// Инициализация структуры Ученик

Student init\_s(string Fam, string Name, string Otch, Mark mas\_m[20], Lessons mas\_l[20]);

// Ввод ФИО ученика в структуру

Student inp\_FIO\_Stud();

// Добавление к ученику урока

int add\_Less\_to\_Stud(Student& st, Lessons& ls);

// Добавление к ученику урока и оценки

int add\_LM\_to\_Stud(Student& st, Lessons& ls, Mark& m);

// Добавление к ученику оценки к конкретному предмету

int add\_Mark\_to\_Stud(Student& st, Lessons& ls, Mark& m);

// Вывод ученика

// Передаем ученика и флаг для различного вывода уроков и оценок

// fl\_out (0 или 1)

void out\_Stud(Student st, int fl\_out);

Файл Student.cpp

#include "Student.h"

// Очищение всех полей структуры Ученик (Student)

Student null\_s()

{

Student st;

st.Fam = "";

st.Name = "";

st.Otch = "";

Lessons L;

Mark M;

L = init\_l("");

M = init\_m(0);

st.i = 0;

for (int i = 0; i < 20; i++)

{

st.mas\_Less[i] = L;

}

for (int i = 0; i < 20; i++)

{

st.mas\_Marks[i] = M;

}

return st;

}

// Инициализация структуры Ученик

Student init\_s(string Fam, string Name, string Otch, Mark mas\_m[20], Lessons mas\_l[20])

{

Student st;

st = null\_s();

st.Fam = Fam;

st.Name = Name;

st.Otch = Otch;

st.i = 0;

int i = 0;

while (st.mas\_Less[i].Name != "")

{

st.mas\_Less[i] = mas\_l[i];

i++;

st.i++;

}

for (int i = 0; i < 20; i++)

{

st.mas\_Marks[i] = mas\_m[i];

}

return st;

}

// Ввод ФИО ученика в структуру

Student inp\_FIO\_Stud()

{

Student st;

st = null\_s();

cout << "Введите Фамилию ученика: ";

cin >> st.Fam;

cout << "Введите Имя ученика: ";

cin >> st.Name;

cout << "Введите Отчество ученика: ";

cin >> st.Otch;

return st;

}

// Добавление к ученику урока

// 0 - не удачно, 1 - удачно

int add\_Less\_to\_Stud(Student& st, Lessons& ls)

{

int fl = 0;

for (int i = 0; (i < 20) && (fl == 0); i++)

{

if (st.mas\_Less[i].Name == "")

{

st.mas\_Less[i] = ls;

fl = 1;

}

}

return fl;

}

// Добавление к ученику урока и оценки

// 0 - не удачно, 1 - удачно

int add\_LM\_to\_Stud(Student& st, Lessons& ls, Mark& m)

{

int fl = 0;

for (int i = 0; (i < 20) && (fl == 0); i++)

{

if (st.mas\_Less[i].Name == "")

{

st.mas\_Less[i] = ls;

st.mas\_Marks[i] = m;

fl = 1;

}

}

return fl;

}

// Добавление к ученику оценки к конкретному предмету

// 0 - не удачно, 1 - удачно

int add\_Mark\_to\_Stud(Student& st, Lessons& ls, Mark& m)

{

int fl = 0;

for (int i = 0; (i < 20) && (fl == 0); i++)

{

if (st.mas\_Less[i].Name == ls.Name)

{

if (st.mas\_Less[i].Teach.Fam == ls.Teach.Fam)

{

if (st.mas\_Less[i].Teach.Name == ls.Teach.Name)

{

if (st.mas\_Less[i].Teach.Otch == ls.Teach.Otch)

{

st.mas\_Marks[i] = m;

fl = 1;

}

}

}

}

}

return fl;

}

// Вывод ученика

// Передаем ученика и флаг для различного вывода уроков и оценок

// fl\_out (0 или 1)

void out\_Stud(Student st, int fl\_out)

{

cout << "ФИО ученика: " << st.Fam << " " << st.Name << " " << st.Otch << endl;

if (fl\_out == 0)

{

cout << "Уроки:" << endl;

for (int i = 0; i < 20; i++)

{

out\_Less(st.mas\_Less[i]);

}

cout << "Оценки:" << endl;

for (int i = 0; i < 20; i++)

{

st.mas\_Marks[i];

}

}

else

{

cout << "Уроки:" << endl;

int i = 0;

while (st.mas\_Less[i].Name != "")

{

cout << " \* " << st.mas\_Less[i].Name << " \* " << st.mas\_Marks[i].Value << " \* " << endl;

i++;

}

}

}

Файл Class.h

#pragma once

#include "Student.h"

// Структура Класс

struct Class\_

{

string Name = ""; // Название класса

string year = ""; // Год обучения

int i = 0; // Текущее кол-во учеников

Student mas\_Stud[32]; // Массив учеников

};

// Очищение полей структуры Класс

Class\_ null\_c();

// Инициализация структуры Класс

Class\_ init\_c(string Name, string Year, Student\* mas[32]);

// Ввод данных в структуру Класс

Class\_ inp\_Class\_();

// Добавление к классу ченика

int add\_St\_to\_Cl(Class\_& cl, Student& st);

// Выод данных из структуры Класс

// fl\_out (0 или 1)

void out\_Class\_(Class\_ cl, int fl\_out);

// Функция по нахождению лучших учеников в классе

int Best\_Stud\_inClass(Class\_ cl);

// Функция по нахождению худших учеников в классе

int Bed\_Stud\_inClass(Class\_ cl);

Файл Class.cpp

#include "Class.h"

// Очищение полей структуры Класс

Class\_ null\_c()

{

Class\_ cl;

cl.Name = "";

cl.year = "";

cl.i = 0;

Student S;

S = null\_s();

for (int i = 0; i < 32; i++)

{

cl.mas\_Stud[i] = S;

}

return cl;

}

// Инициализация структуры Класс

Class\_ init\_c(string Name, string Year, Student\* mas[32])

{

Class\_ cl;

cl.Name = Name;

cl.year = Year;

cl.i = 0;

Student S;

S = null\_s();

for (int i = 0; i < 32; i++)

{

cl.mas\_Stud[i] = \*mas[i];

}

return cl;

}

// Ввод данных в структуру Класс

Class\_ inp\_Class\_()

{

Class\_ cl;

cl = null\_c();

cout << "Введите название класса: ";

cin >> cl.Name;

cout << endl;

cout << "Введите год обучения: ";

cin >> cl.year;

cout << endl;

return cl;

}

// Добавление к классу ченика

int add\_St\_to\_Cl(Class\_& cl, Student& st)

{

int fl = 0;

if (cl.i < 20)

{

cl.mas\_Stud[cl.i] = st;

cl.i++;

fl = 1;

}

return fl;

}

// Вывод данных из структуры Класс

// fl\_out (0 или 1)

void out\_Class\_(Class\_ cl, int fl\_out)

{

cout << "Название класса: " << cl.Name << endl;;

cout << "Год обучения: " << cl.year << endl;

cout << endl;

cout << "Ученики:" << endl;

if (fl\_out == 0)

{

int i = 0;

while (cl.mas\_Stud[i].Fam != "")

{

out\_Stud(cl.mas\_Stud[i], 0);

i++;

}

}

else

{

int i = 0;

while (cl.mas\_Stud[i].Fam != "")

{

out\_Stud(cl.mas\_Stud[i], 1);

i++;

}

}

}

// Функция по нахождению лучших учеников в классе

int Best\_Stud\_inClass(Class\_ cl)

{

double mas\_m[32];

int exit = 0;

for (int i = 0; i < 32; i++)

{

mas\_m[i] = 0;

}

int i = 0;

int k = 0;

int sum = 0;

while (cl.mas\_Stud[i].Fam != "")

{

int j = 0;

k = 0;

sum = 0;

while (cl.mas\_Stud[i].mas\_Less[j].Name != "")

{

sum += cl.mas\_Stud[i].mas\_Marks[j].Value;

k++;

j++;

}

mas\_m[i] = double(sum) / k;

i++;

}

if (i == 0)

exit = -1;

else

{

double maxM = 0;

maxM = mas\_m[0];

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

{

if (maxM < mas\_m[j])

maxM = mas\_m[j];

}

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

{

if (mas\_m[j] == maxM)

out\_Stud(cl.mas\_Stud[j], 1);

}

}

return exit;

}

// Функция по нахождению худших учеников в классе

int Bed\_Stud\_inClass(Class\_ cl)

{

double mas\_m[32];

int exit = 0;

for (int i = 0; i < 32; i++)

{

mas\_m[i] = 0;

}

int i = 0;

int k = 0;

int sum = 0;

while (cl.mas\_Stud[i].Fam != "")

{

int j = 0;

k = 0;

sum = 0;

while (cl.mas\_Stud[i].mas\_Less[j].Name != "")

{

sum += cl.mas\_Stud[i].mas\_Marks[j].Value;

k++;

j++;

}

mas\_m[i] = double(sum) / k;

i++;

}

if (i == 0)

exit = -1;

else

{

double minM = 0;

minM = mas\_m[0];

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

{

if (minM > mas\_m[j])

minM = mas\_m[j];

}

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

{

if (mas\_m[j] == minM)

out\_Stud(cl.mas\_Stud[j], 1);

}

}

return exit;

}

Файл lab2\_t1.cpp

#include "Class.h"

int main()

{

setlocale(LC\_ALL, "Russian");

system("chcp 1251");

system("cls");

cout << "========================Тест структуры Учитель===========================" << endl;

Teacher t\_Chem;

Teacher\* t\_Biology = new Teacher;

t\_Chem = init\_t("", "", "");

\*t\_Biology = init\_t("", "", "");

cout << "Введите ФИО 2-ух учителей:" << endl;

cout << endl;

t\_Chem = inp\_Teach();

out\_Teach(t\_Chem);

cout << endl;

\*t\_Biology = inp\_Teach();

out\_Teach(\*t\_Biology);

cout << "===================================================" << endl;

cout << endl;

cout << endl;

cout << "========================Тест структуры Уроки===========================" << endl;

Lessons Chem;

Lessons\* Biology = new Lessons;

Chem = init\_l("");

\*Biology = init\_l("");

cout << "Введите 2 урока:" << endl;

cout << endl;

cout << "Для учителя - ";

out\_Teach(t\_Chem);

Chem = inp\_Less(t\_Chem);

cout << "Вывод:" << endl;

out\_Less(Chem);

cout << endl;

cout << "Для учителя - ";

out\_Teach(\*t\_Biology);

\*Biology = inp\_Less(\*t\_Biology);

cout << "Вывод:" << endl;

out\_Less(\*Biology);

cout << "===================================================" << endl;

cout << endl;

cout << endl;

cout << "=======================Тест структуры Ученик и Оценки============================" << endl;

Student st1;

Student\* st2 = new Student;

Student st3;

st1 = null\_s();

\*st2 = null\_s();

st3 = null\_s();

cout << "Введите 3 ученика:" << endl;

cout << endl;

cout << "Первый ученик:" << endl;

st1 = inp\_FIO\_Stud();

cout << endl;

cout << "Выберите, какие уроки у первого ученика:" << endl;

cout << endl;

out\_Less(Chem);

string str = "";

cout << "Добавить этот предмет ученику? (д - добавить, другой набор символов - нет): ";

cin >> str;

if (str == "д")

{

if (add\_Less\_to\_Stud(st1, Chem))

{

cout << "Урок добавлен!" << endl;

cout << endl;

cout << "Добавить оценку (1,2,3,4,5 - добавить, другой набор символов - нет):" << endl;

str = "";

cin >> str;

Mark mr;

mr = init\_m(0);

mr.Value = stoi(str);

if ((mr.Value >= 1) || (mr.Value <= 5))

{

if (add\_Mark\_to\_Stud(st1, Chem, mr))

cout << "Оценка добавлена!" << endl;

else

cout << "Оценка не добавлена!" << endl;

}

}

else

cout << "Урок не добавлен!" << endl;

}

else

cout << "Урок не добавлен!" << endl;

cout << endl;

cout << endl;

cout << "Вывод:" << endl;

out\_Stud(st1, 1);

cout << endl;

cout << "Выберите, какие уроки у первого ученика:" << endl;

cout << endl;

out\_Less(\*Biology);

str = "";

cout << "Добавить этот предмет ученику? (д - добавить, другой набор символов - нет): ";

cin >> str;

if (str == "д")

{

if (add\_Less\_to\_Stud(st1, \*Biology))

{

cout << "Урок добавлен!" << endl;

cout << endl;

cout << "Добавить оценку (1,2,3,4,5 - добавить, другой набор символов - нет):" << endl;

str = "";

cin >> str;

Mark mr;

mr = init\_m(0);

mr.Value = stoi(str);

if ((mr.Value >= 1) || (mr.Value <= 5))

{

if (add\_Mark\_to\_Stud(st1, \*Biology, mr))

cout << "Оценка добавлена!" << endl;

else

cout << "Оценка не добавлена!" << endl;

}

}

else

cout << "Урок не добавлен!" << endl;

}

else

cout << "Урок не добавлен!" << endl;

cout << endl;

cout << endl;

cout << "Вывод:" << endl;

out\_Stud(st1, 1);

cout << endl;

cout << "Второй ученик:" << endl;

\*st2 = inp\_FIO\_Stud();

cout << endl;

cout << "Выберите, какие уроки у первого ученика:" << endl;

out\_Less(Chem);

str = "";

cout << "Добавить этот предмет ученику? (д - добавить, другой набор символов - нет): ";

cin >> str;

if (str == "д")

{

if (add\_Less\_to\_Stud(\*st2, Chem))

{

cout << "Урок добавлен!" << endl;

cout << endl;

cout << "Добавить оценку (1,2,3,4,5 - добавить, другой набор символов - нет):" << endl;

str = "";

cin >> str;

Mark mr;

mr = init\_m(0);

mr.Value = stoi(str);

if ((mr.Value >= 1) || (mr.Value <= 5))

{

if (add\_Mark\_to\_Stud(\*st2, Chem, mr))

cout << "Оценка добавлена!" << endl;

else

cout << "Оценка не добавлена!" << endl;

}

}

else

cout << "Урок не добавлен!" << endl;

}

else

cout << "Урок не добавлен!" << endl;

cout << endl;

cout << endl;

cout << "Вывод:" << endl;

out\_Stud(\*st2, 1);

cout << endl;

cout << "Выберите, какие уроки у первого ученика:" << endl;

cout << endl;

out\_Less(\*Biology);

str = "";

cout << "Добавить этот предмет ученику? (д - добавить, другой набор символов - нет): ";

cin >> str;

if (str == "д")

{

if (add\_Less\_to\_Stud(\*st2, \*Biology))

{

cout << "Урок добавлен!" << endl;

cout << endl;

cout << "Добавить оценку (1,2,3,4,5 - добавить, другой набор символов - нет):" << endl;

str = "";

cin >> str;

Mark mr;

mr = init\_m(0);

mr.Value = stoi(str);

if ((mr.Value >= 1) || (mr.Value <= 5))

{

if (add\_Mark\_to\_Stud(\*st2, \*Biology, mr))

cout << "Оценка добавлена!" << endl;

else

cout << "Оценка не добавлена!" << endl;

}

}

else

cout << "Урок не добавлен!" << endl;

}

else

cout << "Урок не добавлен!" << endl;

cout << endl;

cout << endl;

cout << "Вывод:" << endl;

out\_Stud(\*st2, 1);

cout << endl;

cout << "Третий ученик:" << endl;

st3 = inp\_FIO\_Stud();

cout << endl;

cout << "Выберите, какие уроки у первого ученика:" << endl;

out\_Less(Chem);

str = "";

cout << "Добавить этот предмет ученику? (д - добавить, другой набор символов - нет): ";

cin >> str;

if (str == "д")

{

if (add\_Less\_to\_Stud(st3, Chem))

{

cout << "Урок добавлен!" << endl;

cout << endl;

cout << "Добавить оценку (1,2,3,4,5 - добавить, другой набор символов - нет):" << endl;

str = "";

cin >> str;

Mark mr;

mr = init\_m(0);

mr.Value = stoi(str);

if ((mr.Value >= 1) || (mr.Value <= 5))

{

if (add\_Mark\_to\_Stud(st3, Chem, mr))

cout << "Оценка добавлена!" << endl;

else

cout << "Оценка не добавлена!" << endl;

}

}

else

cout << "Урок не добавлен!" << endl;

}

else

cout << "Урок не добавлен!" << endl;

cout << endl;

cout << endl;

cout << "Вывод:" << endl;

out\_Stud(st3, 1);

cout << endl;

cout << "Выберите, какие уроки у первого ученика:" << endl;

cout << endl;

out\_Less(\*Biology);

str = "";

cout << "Добавить этот предмет ученику? (д - добавить, другой набор символов - нет): ";

cin >> str;

if (str == "д")

{

if (add\_Less\_to\_Stud(st3, \*Biology))

{

cout << "Урок добавлен!" << endl;

cout << endl;

cout << "Добавить оценку (1,2,3,4,5 - добавить, другой набор символов - нет):" << endl;

str = "";

cin >> str;

Mark mr;

mr = init\_m(0);

mr.Value = stoi(str);

if ((mr.Value >= 1) || (mr.Value <= 5))

{

if (add\_Mark\_to\_Stud(st3, \*Biology, mr))

cout << "Оценка добавлена!" << endl;

else

cout << "Оценка не добавлена!" << endl;

}

}

else

cout << "Урок не добавлен!" << endl;

}

else

cout << "Урок не добавлен!" << endl;

cout << endl;

cout << endl;

cout << "Вывод:" << endl;

out\_Stud(st3, 1);

cout << "===================================================" << endl;

cout << endl;

cout << endl;

cout << "=======================Тест структуры Класс============================" << endl;

cout << "Создайте 2 класса учеников:" << endl;

cout << endl;

Class\_ cl1;

Class\_\* cl2 = new Class\_;

cl1 = null\_c();

\*cl2 = null\_c();

cout << "Введите данные для первого класса" << endl;

cl1 = inp\_Class\_();

cout << endl;

out\_Stud(st1, 1);

cout << "Добавить данного ученика в класс? (д - добавить, другой набор символов - нет): ";

str = "";

cin >> str;

if (str == "д")

{

if (add\_St\_to\_Cl(cl1, st1))

{

cout << "Ученик добавлен!" << endl;

cout << endl;

}

else

cout << "Ученик не добавлен!" << endl;

}

else

cout << "Ученик не добавлен!" << endl;

cout << endl;

cout << endl;

out\_Stud(\*st2, 1);

cout << "Добавить данного ученика в класс? (д - добавить, другой набор символов - нет): ";

str = "";

cin >> str;

if (str == "д")

{

if (add\_St\_to\_Cl(cl1, \*st2))

{

cout << "Ученик добавлен!" << endl;

cout << endl;

}

else

cout << "Ученик не добавлен!" << endl;

}

else

cout << "Ученик не добавлен!" << endl;

cout << endl;

cout << endl;

out\_Stud(st3, 1);

cout << "Добавить данного ученика в класс? (д - добавить, другой набор символов - нет): ";

str = "";

cin >> str;

if (str == "д")

{

if (add\_St\_to\_Cl(cl1, st3))

{

cout << "Ученик добавлен!" << endl;

cout << endl;

}

else

cout << "Ученик не добавлен!" << endl;

}

else

cout << "Ученик не добавлен!" << endl;

cout << endl;

cout << endl;

cout << "Вывод класаа №1" << endl;

out\_Class\_(cl1, 1);

cout << endl;

cout << endl;

cout << endl;

cout << "Введите данные для второго класса" << endl;

\*cl2 = inp\_Class\_();

cout << endl;

out\_Stud(st1, 1);

cout << "Добавить данного ученика в класс? (д - добавить, другой набор символов - нет): ";

str = "";

cin >> str;

if (str == "д")

{

if (add\_St\_to\_Cl(\*cl2, st1))

{

cout << "Ученик добавлен!" << endl;

cout << endl;

}

else

cout << "Ученик не добавлен!" << endl;

}

else

cout << "Ученик не добавлен!" << endl;

cout << endl;

cout << endl;

out\_Stud(\*st2, 1);

cout << "Добавить данного ученика в класс? (д - добавить, другой набор символов - нет): ";

str = "";

cin >> str;

if (str == "д")

{

if (add\_St\_to\_Cl(\*cl2, \*st2))

{

cout << "Ученик добавлен!" << endl;

cout << endl;

}

else

cout << "Ученик не добавлен!" << endl;

}

else

cout << "Ученик не добавлен!" << endl;

cout << endl;

cout << endl;

out\_Stud(st3, 1);

cout << "Добавить данного ученика в класс? (д - добавить, другой набор символов - нет): ";

str = "";

cin >> str;

if (str == "д")

{

if (add\_St\_to\_Cl(\*cl2, st3))

{

cout << "Ученик добавлен!" << endl;

cout << endl;

}

else

cout << "Ученик не добавлен!" << endl;

}

else

cout << "Ученик не добавлен!" << endl;

cout << endl;

cout << endl;

cout << "Вывод класаа №2" << endl;

cout << endl;

out\_Class\_(\*cl2, 1);

cout << "===================================================" << endl;

cout << endl;

cout << endl;

cout << endl;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Функция поиска лучших учеников в классе\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Лучшие ученики класса №1:" << endl;

Best\_Stud\_inClass(cl1);

cout << endl;

cout << endl;

cout << "Лучшие ученики класса №2:" << endl;

Best\_Stud\_inClass(\*cl2);

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << endl;

cout << endl;

cout << endl;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Функция поиска худших учеников в классе\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Худшие ученики класса №1:" << endl;

Bed\_Stud\_inClass(cl1);

cout << endl;

cout << endl;

cout << "Худшие ученики класса №2:" << endl;

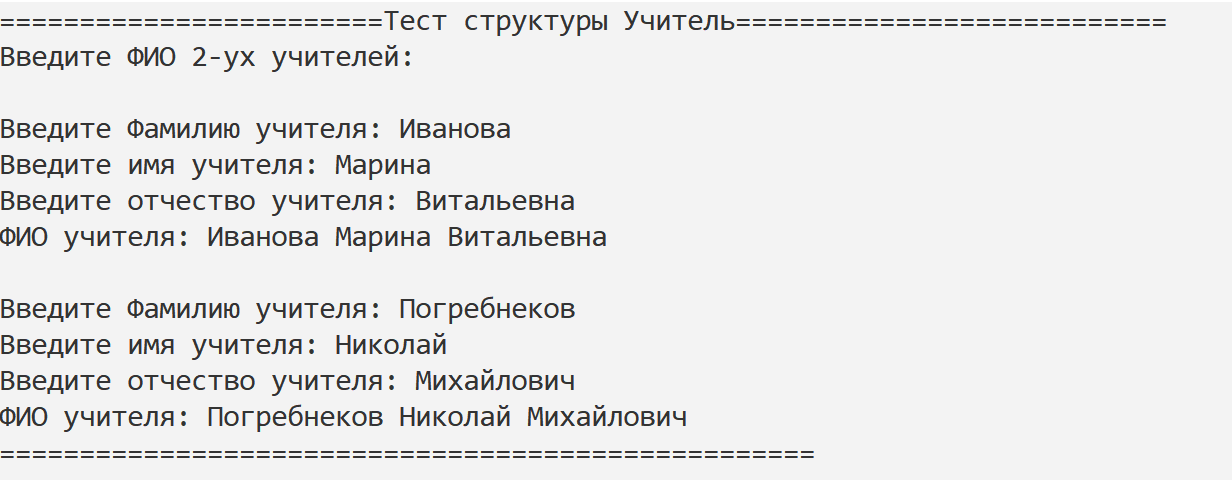
Bed\_Stud\_inClass(\*cl2);

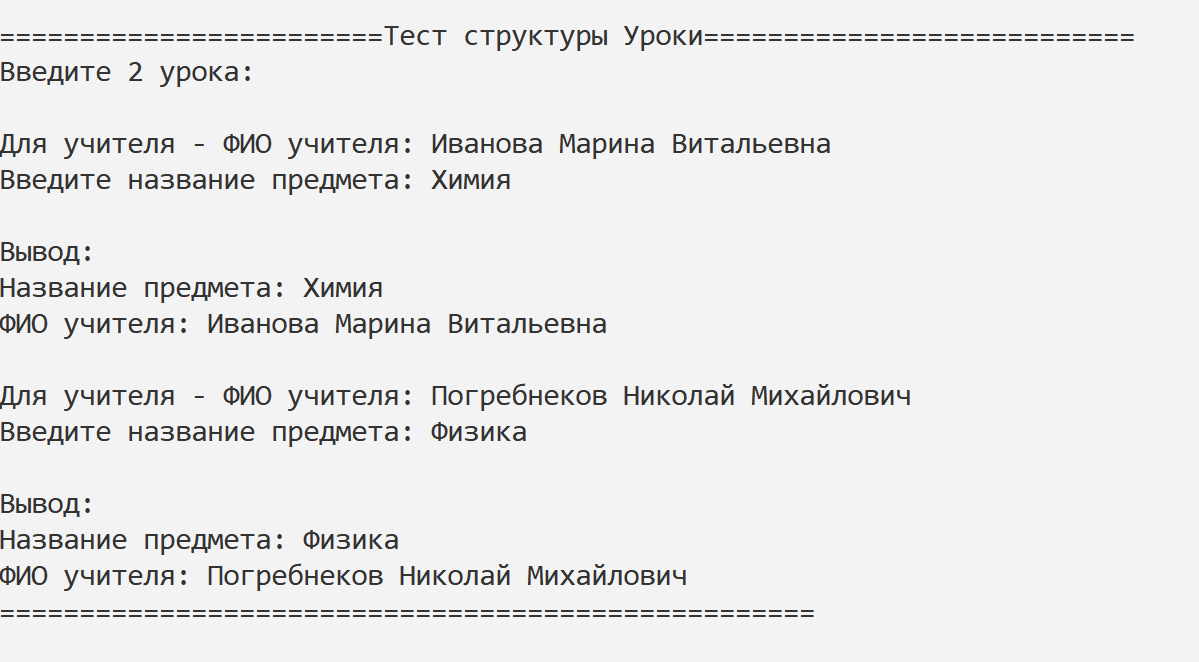
cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

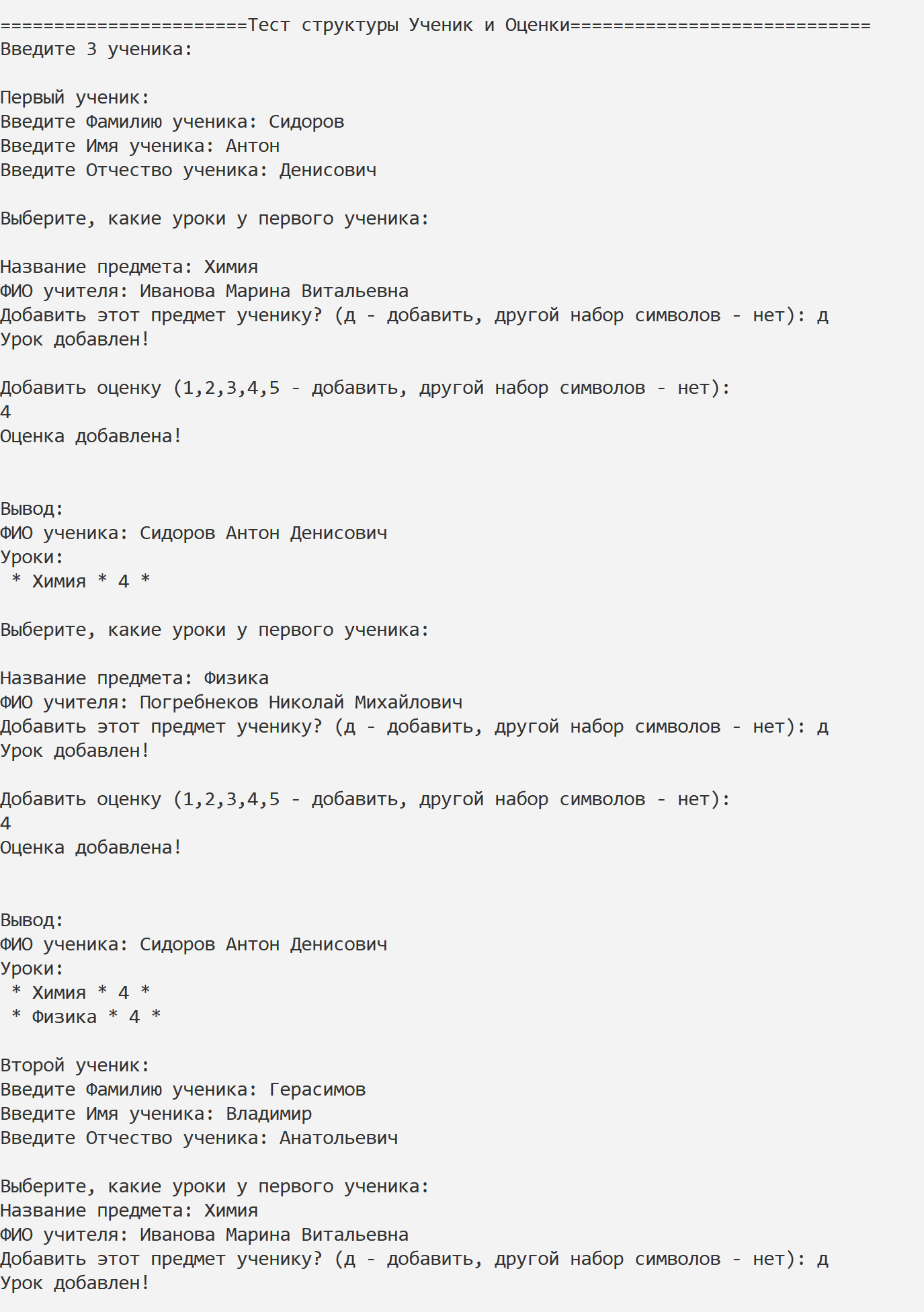
system("Pause");

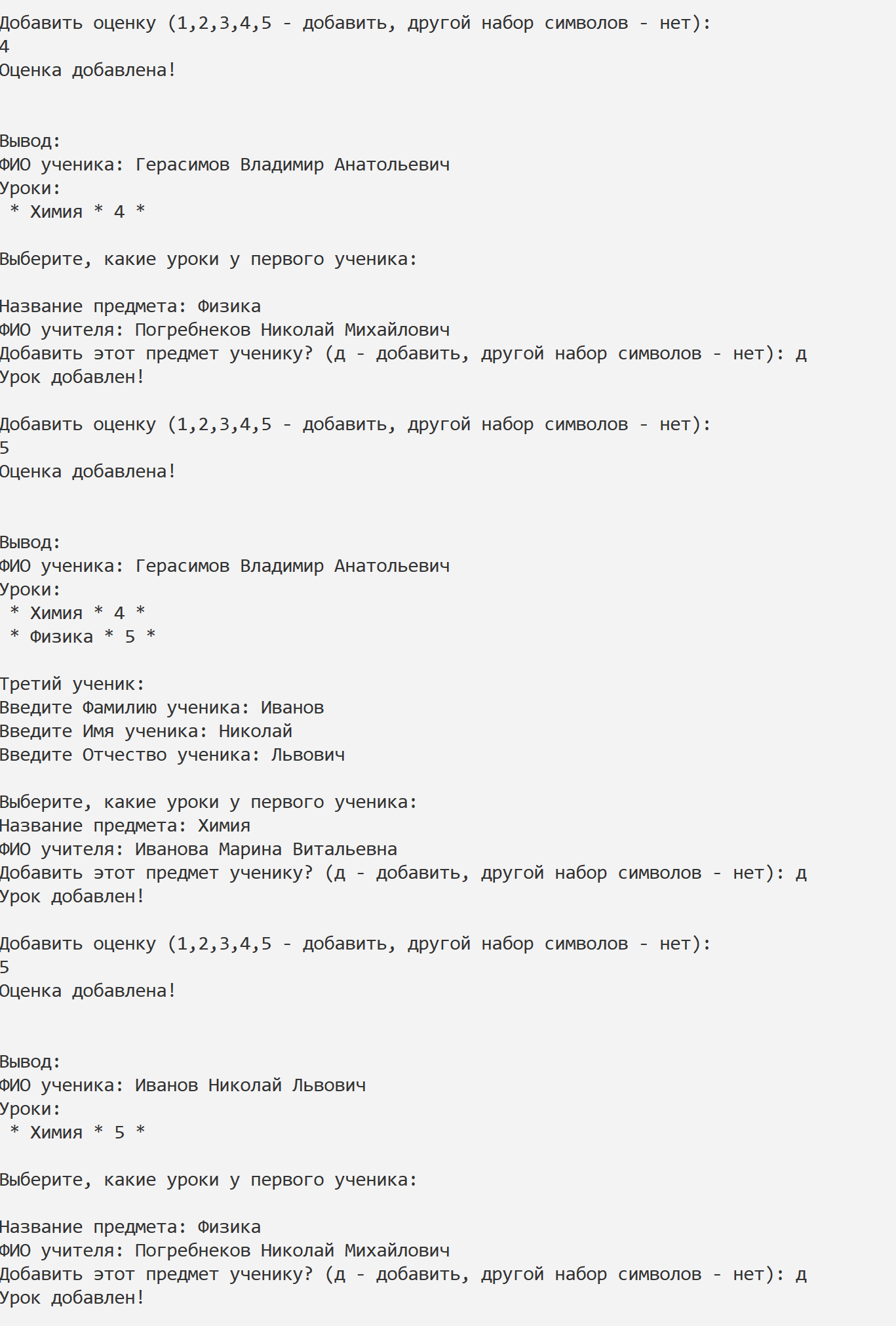
}

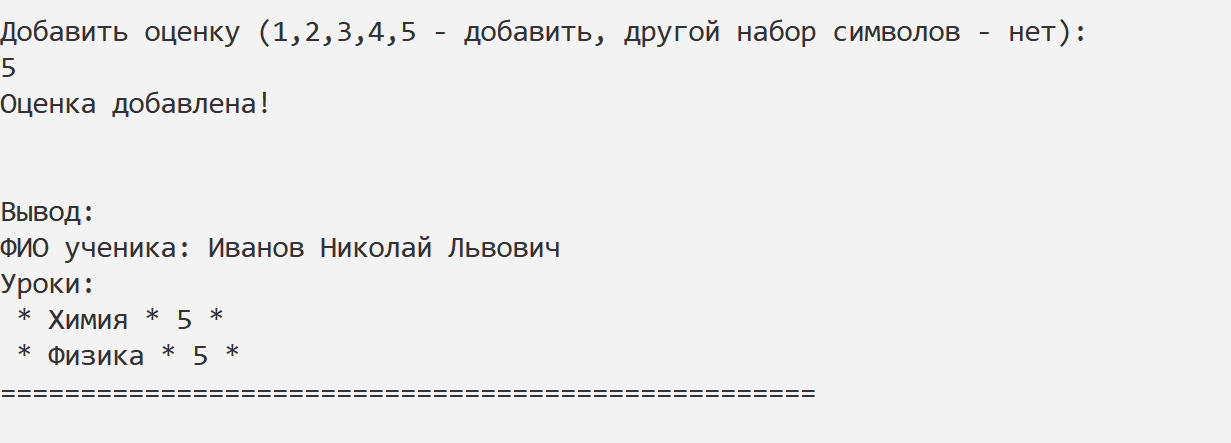
Тесты программы:

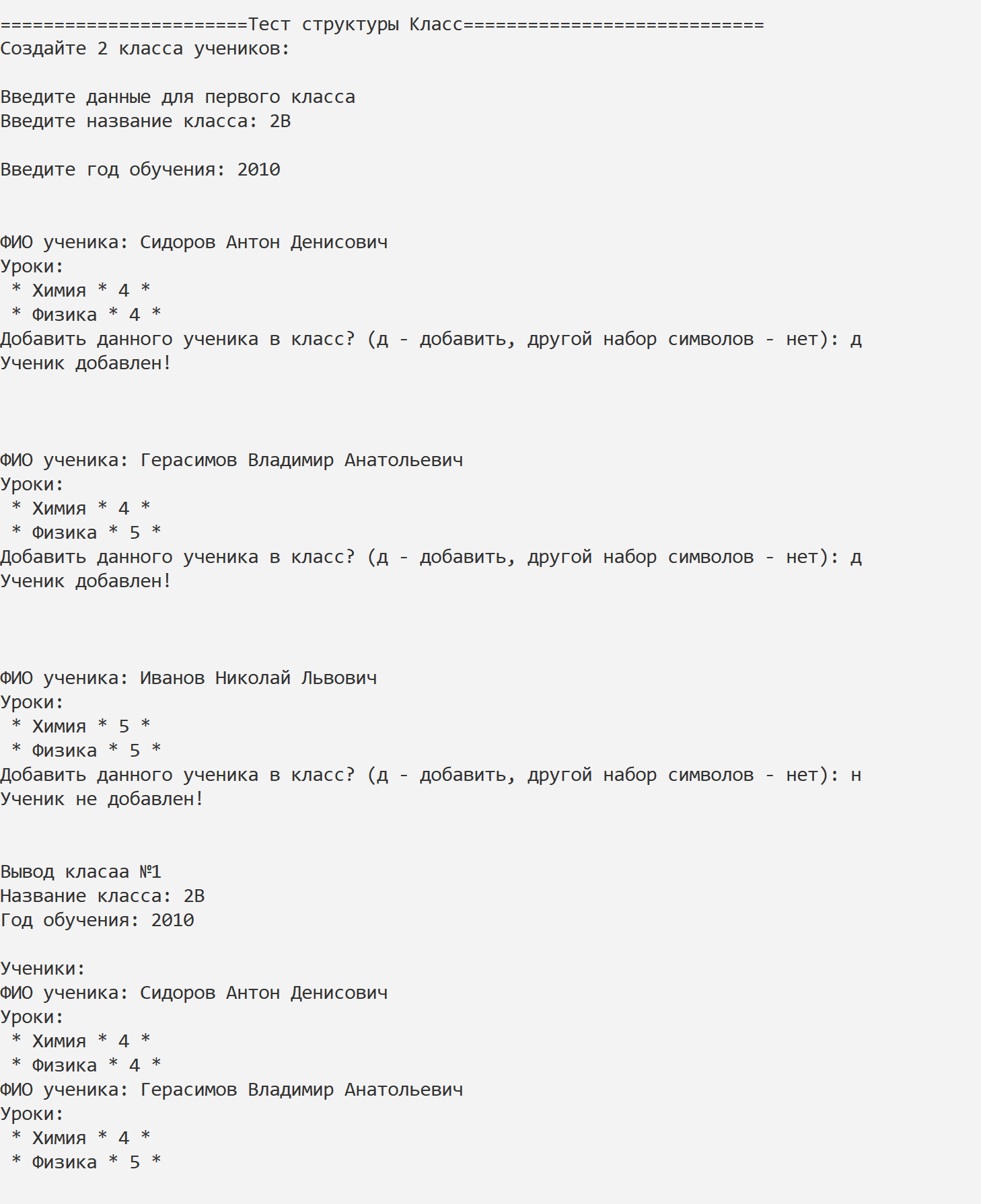


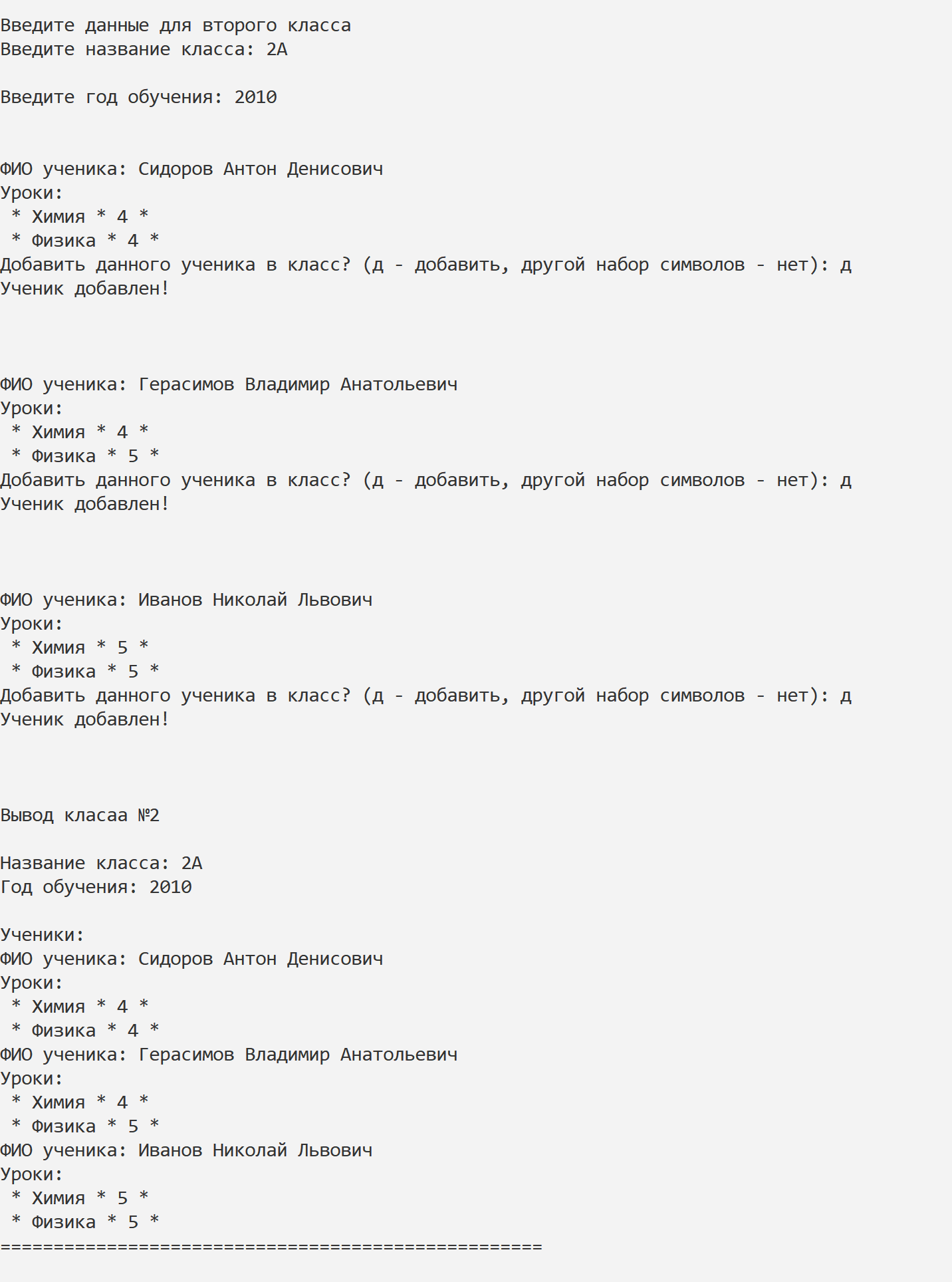


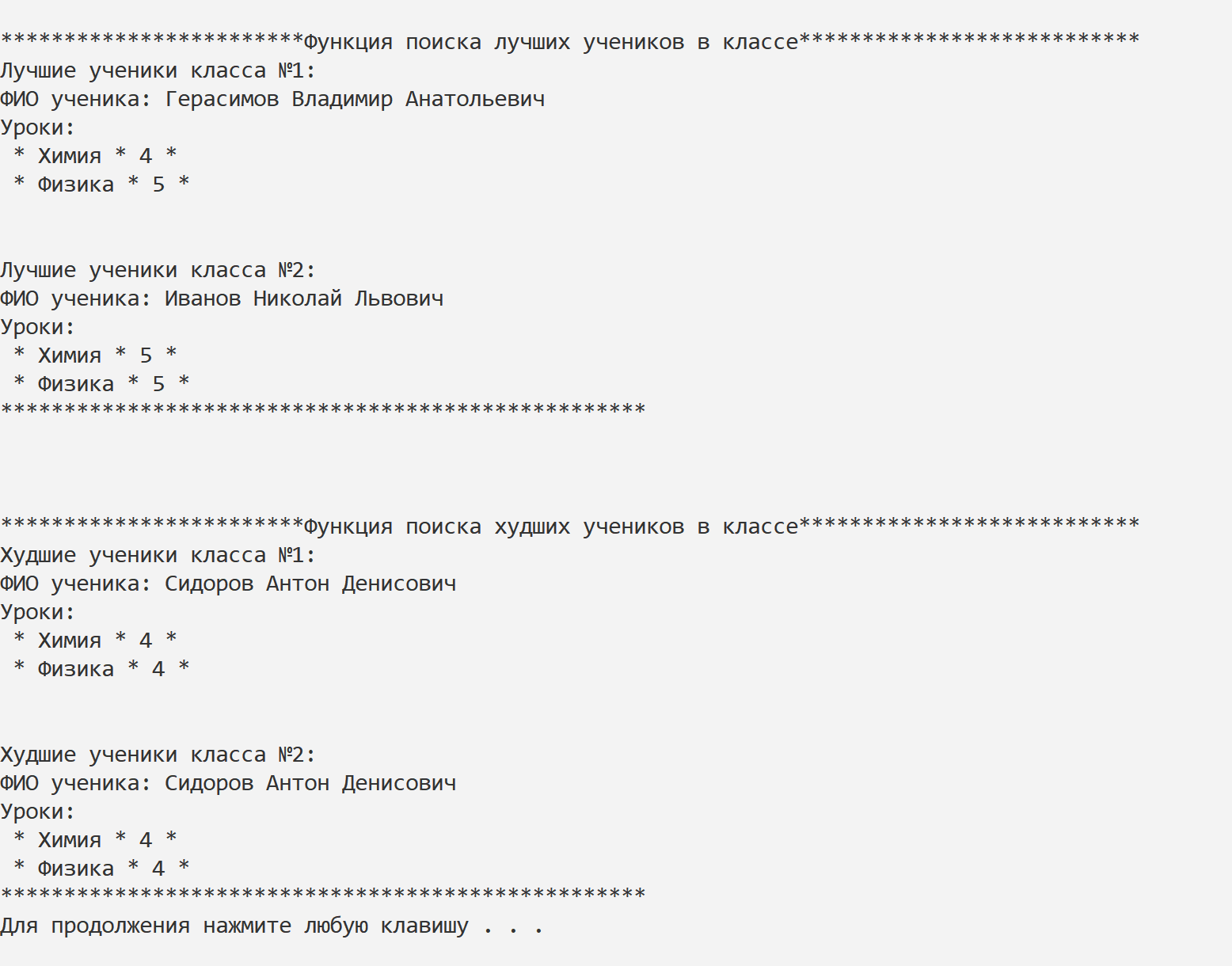












Ссылка на репозиторий на Github: <https://github.com/AlexGr5/lab2_t2.git>