**Министерство образования и науки Кыргызской Республики**

**Кыргызский государственный технический университет им И.Раззакова**

**Факультет информационных технологий**

**Кафедра «Программное обеспечение компьютерных систем»**

**КУРСОВОЙ ПРОЕКТ**

по дисциплине: «**Функционально-ориентированное (структурное) проектирование ПО»**

на тему**: «Справочник Абитуриента »**

Вариант № 2

Выполнил: студентка гр. ПИ(б)-5-19

**Ажиходжоева Каныкей**

Руководитель: доцент **Искаков Р.Т.**

Бишкек - 2020

**ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ**

по дисциплине «Функционально-ориентированное (структурное) проектирование» ПО

Студент Ажиходжоева Каныкей

Группа ПИ-5-19

Дата выдачи 30-03-2020

Дата окончания 15-05-2020

Тема курсового проекта

Справочник абитуриента.

Условие задачи:

База вузов: наименование, адрес, перечень специальностей, конкурс прошлого года по каждой специальности (дневной, вечерней, заочной форм), размер оплаты при договорном обучении. Выбор по разным критериям: все о данном вузе; все о данной специальности, поиск минимального конкурса по данной специальности или вообще.

Ф.И.О. и подпись руководителя: доцент Искаков Р.Т. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ф.И.О. и подпись студента Ажиходжоева Каныкей Алмазовна

**Содержание**

[**Постановка задачи:** 4](#_Toc41048746)

[**Графическое представление алгоритма решения в виде блок-схемы** 5](#_Toc41048747)

[**Словесный алгоритм** 15](#_Toc41048748)

[**Программа на языке С++** 21](#_Toc41048749)

[**Тестовый пример** 30](#_Toc41048750)

[**Список литературы** 34](#_Toc41048751)

# **Постановка задачи:**

В данной курсовой работе требуется создать консольную программу на языке программирования C++, которое обеспечивает формирование, обработку, вывод информации в консоли. Информация храниться в файлах расширения «.txt». Необходимо разработать базу вузов, где для каждого вуза имеются следующие сведения:

1. Наименование.
2. Адрес.
3. Перечень специальностей.
4. Конкурс прошлого года по каждой специальности.
5. Размер оплаты при договорном обучении.

В программе должны быть реализованы следующие возможности:

1. Добавление файлов при помощи ввода в консоль.
2. Вывод файлов в консоль.
3. в файле.
   1. По имени университета.
   2. По номеру специальности.

Программа при выполнении должна сопровождаться диалоговыми сообщениями (создать меню), т.е. пояснять свои действия (иметь понятный для пользователя интерфейс).

Файлы в данной программе реализованы в виде структуры с названием University. Данная структура содержит шесть полей:

1. name – название университета, тип – строчный string;
2. address – адрес, тип – строчный string;
3. specialty[n] – массив сохраняющий номера специальностей, тип – целочисленный int.
4. name\_specialty – название специальности, тип – строчный string;
5. sp[n] - массив сохраняющий проходной балл, тип – целочисленный int.
6. contract[n] - массив, сохраняющий цены за оплату специальностей, тип – целочисленный int.

# **Графическое представление алгоритма решения в виде блок-схемы**

Начало

x

x

return 0

Break;

Операция не найдена.

Break;

Break;

university\_menu()

return 0

Break;

choice\_menu()

case 1

case 2

case 3

university\_menu()

k

k

Конец

Break;

Операция не найдена.

Break;

Break;

university\_show()

Break;

university\_add()

system("cls")

case 1

case 2

case 3

Ошибка

!in.is\_open()

in.open("University.txt")

choice\_university()

ifstream in

e

check=kol=cod\_un=0

нет да

un.name

un.address

!in.eof()

e == n.name

Нет Да

check++

un.specialty[i]

Break;

Break;

Break;

Break;

b\_un[i].name\_specialty = ОФС

b\_un[i].name\_specialty = ИВТ

b\_un[i].name\_specialty = ИБ

b\_un[i].name\_specialty = ПИ

un.specialty[i] un.sp[i]

un.contract[i]

i = 0; i < n; i++

cod\_spec

b\_un[i].name\_specialty

un.sp[i]

un.contract[i]

i == 0

cod\_spec = i +1

нет да

Совпадения не найдены!

check>(kol-1)

kol++

b\_un[i].name\_specialty

un.sp[i]

un.contract[i]

да

нет

Конец

in.close()

university\_show()

cod\_un = 0

Ошибка

in.open("University.txt")

ifstream in

!in.is\_open()

Нет да

Ошибка

in.peek()=EOF

Нет да

un.specialty[i]

b\_un[i].name\_specialty = ПИ

Break;

Break;

Break;

Break;

b\_un[i].name\_specialty = ОФС

b\_un[i].name\_specialty = ИВТ

b\_un[i].name\_specialty = ИБ

un.specialty[i]

un.sp[i]

un.contract[i]

i = 0; i < n; i++

cod\_un++

un.name

un.address

!in.eof()

cod\_un++

cod\_spec

b\_un[i].name\_specialty

un.sp[i]

un.contract[i]

i == 0

b\_un[i].name\_specialty

un.sp[i]

un.contract[i]

in.close()

Конец

Да

l = 0

ofstream out

out.open("University.txt", ios::app)

l == 0

l = 0

university\_add()

out.close()

YesOrNot == 1

YesOrNot

un.specialty[i]

un.sp[i]

un.contract[i]

I=0;i<n;i++

un.name

un.address

нет

Да

Нет

Да

нет

Конец

choice\_menu()

L

L

Case 1

Break;

choice\_university()

Break;

Операция не найдена.

Break;

system("cls")

Break;

choice\_specialty()

Case 2

Case 3

Конец

choice\_specialty()

check=kol=cod\_un=0

!in.eof()

Ошибка

!in.is\_open()

in.open("University.txt")

e

ifstream in

Нет Да

Break;

b\_un[i].name\_specialty = ОФС

b\_un[i].name\_specialty = ИВТ

b\_un[i].name\_specialty = ИБ

b\_un[i].name\_specialty = ПИ

Break;

Break;

Break;

un.specialty[i]

un.specialty[i] un.sp[i]

un.contract[i]

i = 0; i < n; i++

un.name

un.address

да

e==un.specialty[i]

Нет Да

cod\_un++

check++

cod\_spec = i + 1

cod\_un

un.name,

un.address

cod\_spec,

b\_un[i].name\_specialty

un.sp[i],un.contract[i]

kol++

check > (kol - 1)

Да

Ошибка

нет

in.close()

Конец

# **Словесный алгоритм**

Создать константную переменную n типа int и присвоить ей значение 4.

Создать переменную cod\_un типа int и присвоить ей значение 0.

Создать переменную cod\_spec типа int.

Создать структуру University с 6 полями name, address, specialty[n], name\_specialty, sp[n], contract[n].

Создать переменную un типа University.

**Функция main**

**Начало**

1. Вывести все пункты меню.
2. Ввести номер операции.
3. С помощью множественного условного оператора-Switch Case, выполнить функцию в соответствии с веденным номером.
4. Если введена операция 1, то вызывать функцию university\_menu(),потом возвращаться к пункту 1.
5. Если введена операция 2, то вызывать функцию choice\_menu(),потом возвращаться к пункту 1.
6. Если введена операция 3, то переходить к пункту 4.
7. Если введенной операции нет, выводить сообщение об ошибке.
8. Возвращать 0.

**Конец**

**Функция university\_menu()**

**Начало**

1. Вывести все пункты меню.
2. Ввести номер операции.
3. С помощью множественного условного оператора-Switch Case, выполнить функцию в соответствии с веденным номером.
4. Если введена операция 1, то вызывать функцию university\_show(),потом возвращаться к пункту 1.
5. Если введена операция 2, то вызывать функцию university\_add(),потом возвращаться к пункту 1.
6. Если введена операция 3, то возвращаться к функции Main.
7. Если введенной операции нет, выводить сообщение об ошибке и возвращаться к функции Main.

**Конец**

**Функция university\_show()**

**Начало**

1. Открыть файл "University.txt», в режиме чтения.
2. Если нет доступа к файлу, то выдавать ошибку. Если есть файл, но он пустой, то выдавать сообщение о том, что файл пуст.
3. Иначе, работает цикл While, который считывает все данные до конца файла.
4. Выводить на консоль данные Name и Address.
5. Добавить 1 к переменной cod\_un.
6. Начинает работать цикл For от 0 до n, с шагом 1.Он будет идти до пункта 12.
7. Выводить на консоль данные Specialty, sp, contract.
8. С помощью множественного условного оператора-Switch Case, выполнить функцию в соответствии с веденной Specialty.
9. Если введена операция 1, то выводить “Программная Инженерия”, далее выходить из Switch Case.
10. Если введена операция 2, то выводить “Информационная Безопасность”, далее выходить из Switch Case.
11. Если введена операция 3, то выводить “ Информатика и вычислительная техника”, далее выходить из Switch Case.
12. Если введена операция 4, то выводить “ Управление в технических системах”, далее выходить из Switch Case.
13. Добавить 1 к переменной cod\_spec.
14. Если переменная i=0, то выводить на консоль Name,sp, contract. Иначе, вводить в файл spec,name,sp,contract.
15. Выход из цикла For.
16. Выход из цикла While.
17. Закрыть файл.

**Конец**

**Функция university\_add()**

**Начало**

1. Открыть файл "University.txt», в режиме добавления.
2. Начинается цикл Do-While. Он будет идти до пункта 10.
3. Если переменная l=0, то добавить к ней 1.
4. Вводить в консоль и в файл данные Name и Address.
5. Начинает работать цикл For от 0 до n, с шагом 1. Он будет идти до пункта 6.
6. Вводить в консоль и в файл данные Specialty , sp, contract.
7. Конец цикла For.
8. Ввести да или нет.
9. Если введена 1, то продолжать цикл Do-While, иначе выход.
10. Закрыть файл.

**Конец**

**Функция choice\_menu()**

**Начало**

1. Вывести все пункты меню.
2. Ввести номер операции.
3. С помощью множественного условного оператора-Switch Case, выполнить функцию в соответствии с веденным номером.
4. Если введена операция 1, то вызывать функцию choice\_university(),потом возвращаться к пункту 1.
5. Если введена операция 2, то вызывать функцию choice\_specialty(),потом возвращаться к пункту 1.
6. Если введена операция 3, то возвращаться к функции Main.
7. Если введенной операции нет, выводить сообщение об ошибке и возвращаться к функции Main.

**Конец**

**Функция choice\_university()**

**Начало**

1. Открыть файл "University.txt», в режиме чтения.
2. Ввести название университета.
3. Переменную cod\_un прировнять к 0.
4. Если нет доступа к файлу, то выдавать ошибку, иначе, работает цикл While, который считывает все данные до конца файла и работает до пункта 17.
5. Выводить из файла данные Name и Address.
6. Если введенная переменная равна Name, то добавить 1 к переменной cod\_un.Начало условного оператора if,который закончится на 13 пункте.
7. Вывод на консоль cod\_un, un.name, un.address.
8. Начинает работать цикл For от 0 до n, с шагом 1. Он будет идти до пункта 12.
9. С помощью множественного условного оператора-Switch Case, выполнить функцию в соответствии с веденной Specialty.
10. Если введена операция 1, то выводить “Программная Инженерия”, далее выходить из Switch Case.
11. Если введена операция 2, то выводить “Информационная Безопасность”, далее выходить из Switch Case.
12. Если введена операция 3, то выводить “ Информатика и вычислительная техника”, далее выходить из Switch Case.
13. Если введена операция 4, то выводить “ Управление в технических системах”, далее выходить из Switch Case.
14. Переменная cod\_spec равна переменной i с добавлением 1.
15. Если переменная i=0, то выводить на консоль Name,sp, contract. Иначе, вводить в файл spec,name,sp,contract.
16. Конец цикла For.
17. Конец условного оператора if.
18. Иначе, добавить к переменной check 1.
19. Добавить к переменной kol 1.
20. Если переменная check больше чем разность переменной kol и 1, то выводить сообщение о том что нет совпадений.
21. Выход из While.
22. Закрыть файл.

**Конец**

**Функция choice\_specialty()**

**Начало**

1. Открыть файл "University.txt», в режиме чтения.
2. Ввести номер специальности.
3. Если нет доступа к файлу, то выдавать ошибку, иначе, работает цикл While, который считывает все данные до конца файла и работает до пункта 11.
4. Выводить из файла данные Name и Address.
5. Начинает работать цикл For от 0 до n, с шагом 1. Он будет идти до пункта 9.
6. С помощью множественного условного оператора-Switch Case, выполнить функцию в соответствии с веденной Specialty.
7. Если введена операция 1, то выводить “Программная Инженерия”, далее выходить из Switch Case.
8. Если введена операция 2, то выводить “Информационная Безопасность”, далее выходить из Switch Case.
9. Если введена операция 3, то выводить “ Информатика и вычислительная техника”, далее выходить из Switch Case.
10. Если введена операция 4, то выводить “ Управление в технических системах”, далее выходить из Switch Case.
11. Если введенная переменная равна Specialty, то добавить 1 к переменной cod\_un.Начало условного оператора if.Иначе добавить к переменной check 1.
12. Добавить к переменной kol 1.
13. Конец цикла For.
14. Если переменная check больше чем разность переменной kol и 1, то выводить сообщение о том что нет совпадений.
15. Выход из While.
16. Закрыть файл.

**Конец**

# **Программа на языке С++**

#include <iostream>

#include <iomanip>

#include <windows.h>

#include <fstream>

#include <locale.h>

#include <cstring>

#include <string>

using namespace std;

const int n = 4;

struct University //Структура "Universty"

{

string name;

string address;

int specialty[n];

string name\_specialty;

int sp[n];

int contract[n];

};

University un;

int cod\_un = 0;

int cod\_spec;

void university\_menu();

void university\_show();

void university\_add();

void choice\_menu();

void choice\_university();

void choice\_specialty();

int main()//Главная функция Main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int x;

metka:

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

cout << "| Справочник абитуриента |" << endl;

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

cout << "| Главное Меню: |" << endl;

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

cout << "| 1.Список университетов. |" << endl;

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

cout << "| 2.Выбор. |" << endl;

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

cout << "| 3.Выход. |" << endl;

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

cout << endl;

cout << "Введите номер операции:";

cin >> x;

system("cls");

switch (x)

{

case 1:

university\_menu();

goto metka; break;

case 2:

choice\_menu();

goto metka; break;

case 3:

return 0; break;

default:

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

goto metka; break;

}

return 0;

}

void university\_menu()//Меню списка университетов

{

system("cls");

int k;

un:

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

cout << "| Список университетов: |" << endl;

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

cout << "| 1.Показать университеты. |" << endl;

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

cout << "| 2.Добавить университеты. |" << endl;

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

cout << "| 3.Назад. |" << endl;

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

cout << endl;

cout << "Введите номер операции:";

cin >> k;

system("cls");

switch (k)

{

case 1:

university\_show();

goto un; break;

case 2:

university\_add();

goto un; break;

case 3:

system("cls"); break;

default:

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

goto un; break;

}

}

void university\_show()//Функция показывающая все данные об университетах

{

cod\_un = 0;

system("cls");

string name\_un;

cout << endl;

cout << "\*Курс предоставлен Национальным Банком Кыргызкой Республики, на 07.05.2020." << endl;

cout << endl;

ifstream in;

in.open("University.txt");

if (!in.is\_open())

{

cout << "Ошибка открытия файла!" << endl;

}

else if (in.peek() == EOF) {

cout << "Файл пуст!Введите данные!" << endl;

}

else

{

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

cout << "| № | Университет | Адреc | № | Специальность | Проходной балл | Контракт |" << endl;

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

while (!in.eof())

{

in >> un.name >> un.address;

cod\_un++;

cout << "| " << setw(1) << cod\_un << " | " << setw(11) << un.name << " | " << setw(17) << un.address << " | " << 1;

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

{

in >> un.specialty[i];

in >> un.sp[i];

in >> un.contract[i];

switch (un.specialty[i])

{

case 1: name\_un = "Программная Инженерия"; break;

case 2: name\_un = "Информац.Безопасность"; break;

case 3: name\_un = "ИВТ"; break;

case 4: name\_un = "ОФС"; break;

}

cod\_spec = i + 1;

if (i == 0)

{

cout << " | " << setw(21) << name\_un << " | " << setw(14) << un.sp[i] << " | " << setw(8) << un.contract[i] << " | " << endl;

}

else

{

cout << "| " << setw(4) << " | " << setw(14) << " | " << setw(20) << " | " << cod\_spec << " | " << setw(21) << name\_un << " | " << setw(14) << un.sp[i] << " | " << setw(8) << un.contract[i] << " | " << endl;

}

}

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

}

}

in.close();

}

void university\_add()//Функция вводящая новые данные об университетах

{

int l = 0;

int YesOrNot = 1;

ofstream out;

out.open("University.txt", ios::app);

do {

if (l == 0)

{

out << endl;

l++;

}

cout << "Введите название университета который хотите добавить: ";

cin >> un.name;

cout << "Введите адресс университета:";

cin >> un.address;

out << un.name << endl;

out << un.address << endl;

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

cout << "| 1 - Программная\_Инженерия |" << endl;

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

cout << "| 2 - Информац. Безопасность |" << endl;

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

cout << "| 3 - ИВТ |" << endl;

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

cout << "| 4 - ОФС |" << endl;

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

cout << endl;

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

{

cout << "Введите код " << i + 1 << "-вой специальности: ";

cin >> un.specialty[i];

cout << "Введите минимальный балл " << i + 1 << "-вой специальности: ";

cin >> un.sp[i];

cout << "Введите контракт " << i + 1 << "-вой специальности: ";

cin >> un.contract[i];

cout << endl;

out << un.specialty[i] << endl;

out << un.sp[i] << endl;

out << un.contract[i];

if (i != 3)

{

out << endl;

}

}

system("cls");

cout << "Продолжить ввод университета? Да = 1; Нет = 0" << endl;

cin >> YesOrNot;

system("cls");

} while (YesOrNot == 1);

out.close();

}

void choice\_menu()//Меню выбора

{

int l;

n:

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

cout << "| Выбор: |" << endl;

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

cout << "| 1.Университет. |" << endl;

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

cout << "| 2.Специальность. |" << endl;

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

cout << "| 3.Назад. |" << endl;

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

cout << endl;

cout << "Введите номер операции:";

cin >> l;

system("cls");

switch (l)

{

case 1:

choice\_university();

goto n; break;

case 2:

choice\_specialty();

goto n; break;

case 3:

system("cls"); break;

default:

system("cls");

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

goto n; break;

}

}

void choice\_university()//Функция которая по вводу имени ищет название университета

{

ifstream in;

string e;

University b\_un[n];

int check = 0, kol = 0;

cout << "Введите название Университета:";

cin >> e;

cod\_un = 0;

in.open("University.txt");

if (!in.is\_open())

{

cout << "Ошибка открытия файла!" << endl;

}

else

{

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

cout << "| № | Университет | Адреc | № | Специальность | Проходной балл | Контракт |" << endl;

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

while (!in.eof())

{

in >> un.name >> un.address;

if (e == un.name)

{

cod\_un++;

cout << "| " << setw(1) << cod\_un << " | " << setw(11) << un.name << " | " << setw(17) << un.address << " | " << 1;

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

{

in >> un.specialty[i];

in >> un.sp[i];

in >> un.contract[i];

switch (un.specialty[i])

{

case 1: b\_un[i].name\_specialty = "Программная Инженерия"; break;

case 2: b\_un[i].name\_specialty = "Информац.Безопасность"; break;

case 3: b\_un[i].name\_specialty = "ИВТ"; break;

case 4: b\_un[i].name\_specialty = "ОФС"; break;

}

cod\_spec = i + 1;

if (i == 0)

{

cout << " | " << setw(21) << b\_un[i].name\_specialty << " | " << setw(14) << un.sp[i] << " | " << setw(8) << un.contract[i] << " | " << endl;

}

else

{

cout << "| " << setw(4) << " | " << setw(14) << " | " << setw(20) << " | " << cod\_spec << " | " << setw(21) << b\_un[i].name\_specialty << " | " << setw(14) << un.sp[i] << " | " << setw(8) << un.contract[i] << " | " << endl;

}

}

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

}

else

{

check++;

}

kol++;

}

if (check > (kol - 1))

{

cout << "Совпадения не найдены!" << endl;

}

}

in.close();

}

void choice\_specialty()//Функция которая по вводу номера ищет специальность

{

ifstream in;

int e, check = 0, kol = 0;

University b\_un[n];

cod\_un = 0;

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

cout << "| 1 - Программная\_Инженерия |" << endl;

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

cout << "| 2 - Информац. Безопасность |" << endl;

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

cout << "| 3 - ИВТ |" << endl;

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

cout << "| 4 - ОФС |" << endl;

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

cout << "Введите код специальности:";

cin >> e;

system("cls");

in.open("University.txt");

if (!in.is\_open())

{

cout << "Ошибка открытия файла!" << endl;

}

else

{

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

cout << "| № | Университет | Адреc | № | Специальность | Проходной балл | Контракт |" << endl;

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

while (!in.eof())

{

in >> un.name >> un.address;

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

{

in >> un.specialty[i];

in >> un.sp[i];

in >> un.contract[i];

switch (un.specialty[i])

{

case 1: b\_un[i].name\_specialty = "Программная Инженерия"; break;

case 2: b\_un[i].name\_specialty = "Информац.Безопасность"; break;

case 3: b\_un[i].name\_specialty = "ИВТ"; break;

case 4: b\_un[i].name\_specialty = "ОФС"; break;

}

if (e == un.specialty[i])

{

cod\_un++;

cod\_spec = i + 1;

cout << "| " << setw(1) << cod\_un << " | " << setw(11) << un.name << " | " << setw(17) << un.address << " | " << cod\_spec << " | " << setw(21) << b\_un[i].name\_specialty;

cout << " | " << setw(14) << un.sp[i] << " | " << setw(8) << un.contract[i] << " | " << endl;

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

}

else

{

check++;

}

kol++;

}

}

if (check > (kol - 1))

{

cout << "Нет специальности с данным кодом!" << endl;

}

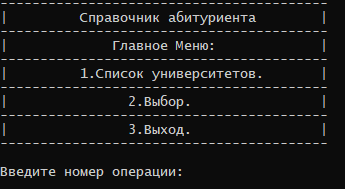
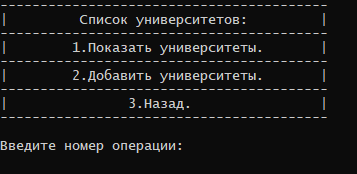
}

in.close();

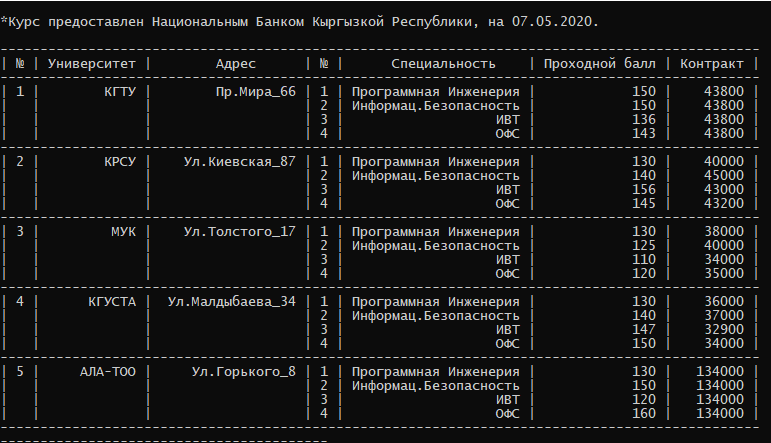
}

# **Тестовый пример**

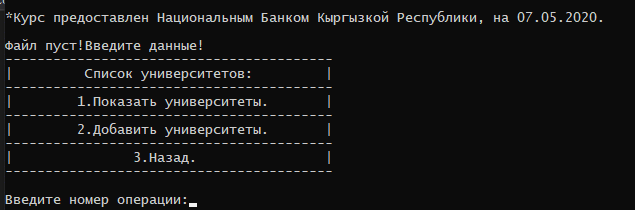
Главное меню: Меню списка университетов:

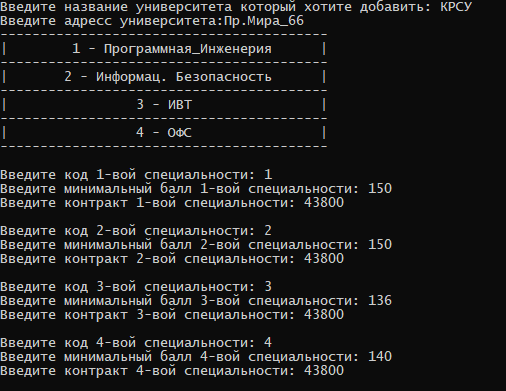
Вывод университетов из файла:



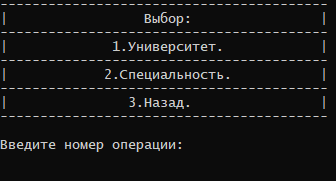
Вывод университетов из файла(Если файл пуст):



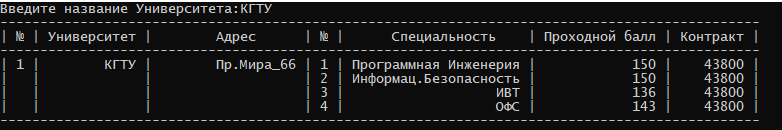
Ввод новых университетов в файл:

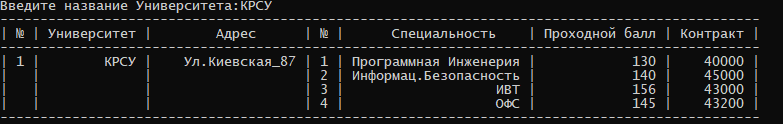


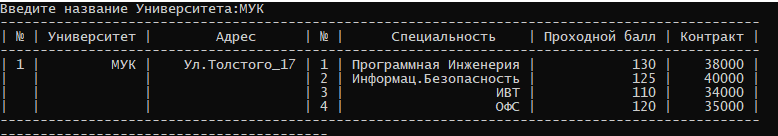
Меню выбора:



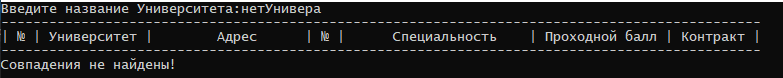
Выбор по университету:



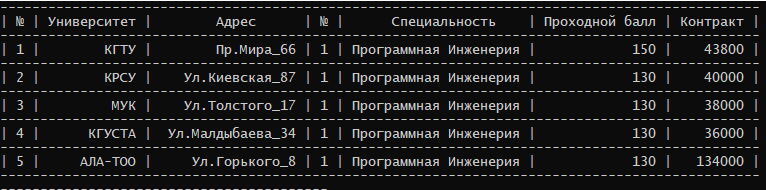


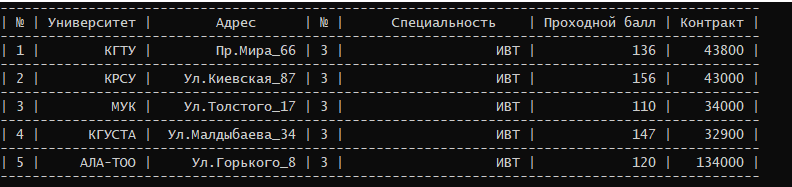


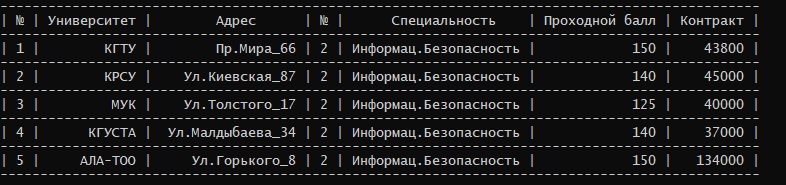
Выбор по университету (Если его нет):



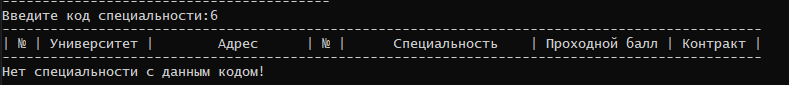
Выбор по специальности:



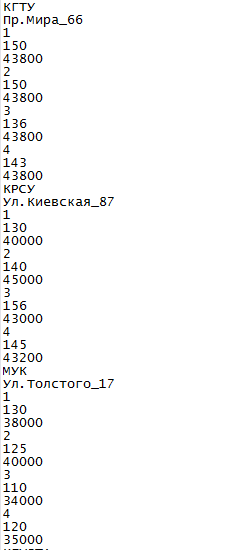




Выбор по специальности (если данного номера нет):



Список университетов в файле:



# **Список литературы**

1. Культин, Н. Microsoft Visual C++ в задачах и примерах / Н. Культин. –Издательство: BHV, 2010. – 274 с.
2. Мейерс, С. Эффективное использование С++. 55 верных способов улучшить структуру и код ваших программ. — М.: ДМК Пресс, 2006. — 300с.
3. Прата, C. Язык программирования C++ (C++11). Лекции и упражнения, 6-е издание — М.: Вильямс, 2012. — 1248 с.
4. Сидорина, Т. Самоучитель Microsoft Visual Studio C++ и MFC / Т.Сидорина. – Издательство: BHV, 2009. – 848 с.
5. Страуструп, Б. Язык программирования С++. Специальное издание. Пер. с англ. — М.: Издательство Бином, 2011 г. — 1136 с.
6. Элджер, Дж. C++. Библиотека программиста: Пер. с англ. — СПб.: Питер, 2000. — 320 с.