Министерство образования Республики Беларусь

Учреждение образования

Брестский государственный технический университет

Кафедра ИИТ

Лабораторная работа №12

За 2 семестр

По дисциплине: ОАиП

Тема: “Бинарные и текстовые файлы”

Подготовил:

Студент 1 курса

Группы ПО-4(2)

Левоцкий Н. Д.

Проверил:

Хацкевич М. В.

Брест 2020

Цель работы: Изучить принципы программирования с использованием

бинарных файлов. Ознакомиться с основными функциями в Си для работы с

бинарными файлами.

Задание:

В программу разработанную в прошлой лабораторной работе добавить чтение и сохранение данных массива структур при помощи бинарных файлов следующим образом:

1. При первом запуске программы должен создаваться бинарный или текстовый файл на выбор пользователя для хранения данных

из массива структур.

2.При добавлении новой записи в массив структур в файл должна дописываться новая запись, без изменения остальных записей.

3. При повторном запуске программы, если файл уже существует, то информация в массив структур должна читаться из этого файла. Если файл отсутствует, то он должен создаваться (см. Пункт 1).

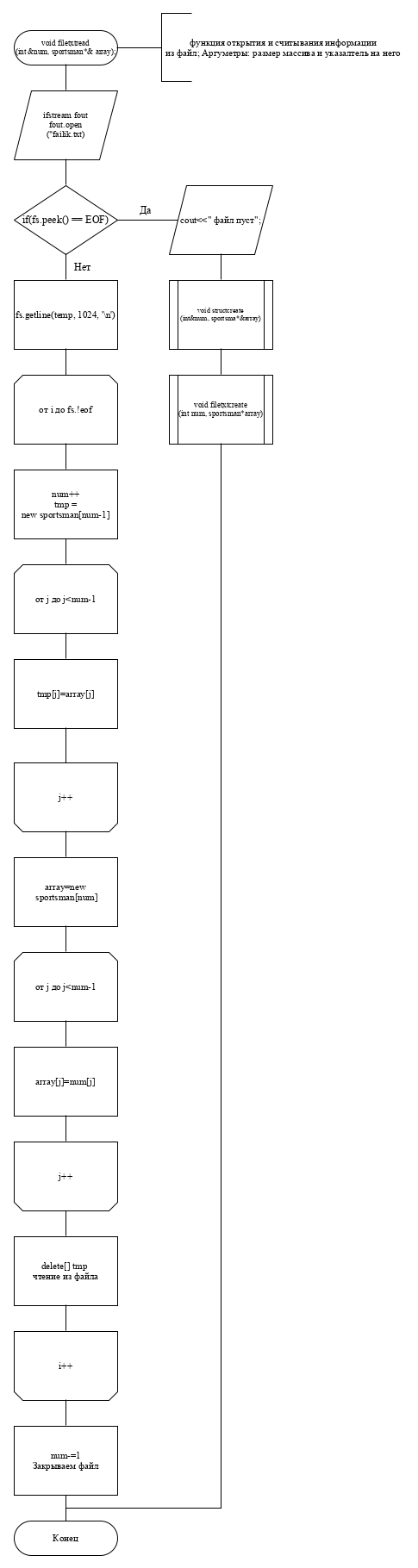
4. Все изменения (сортировка, изменения полей записи, удаление записи) – сохраняются в файле при помощи полной перезаписи содержимого.

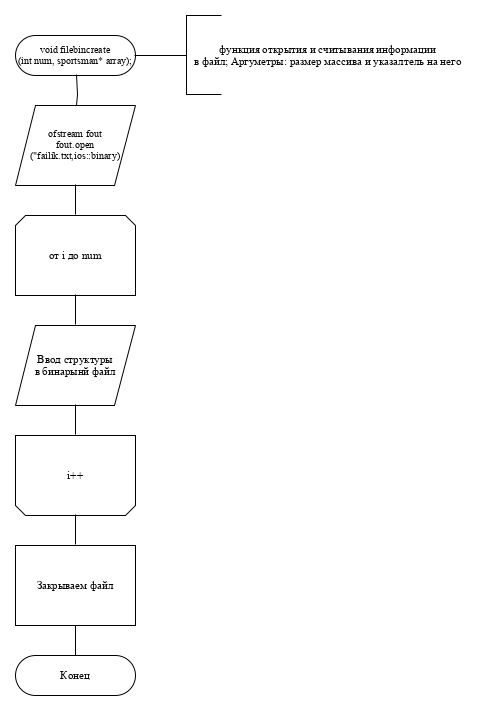
5. Сделать вывод о том, какие преимущества использования конкретного типа файлов (бинарные или текстовые) в решаемой вами задаче.

**Вариант 7**

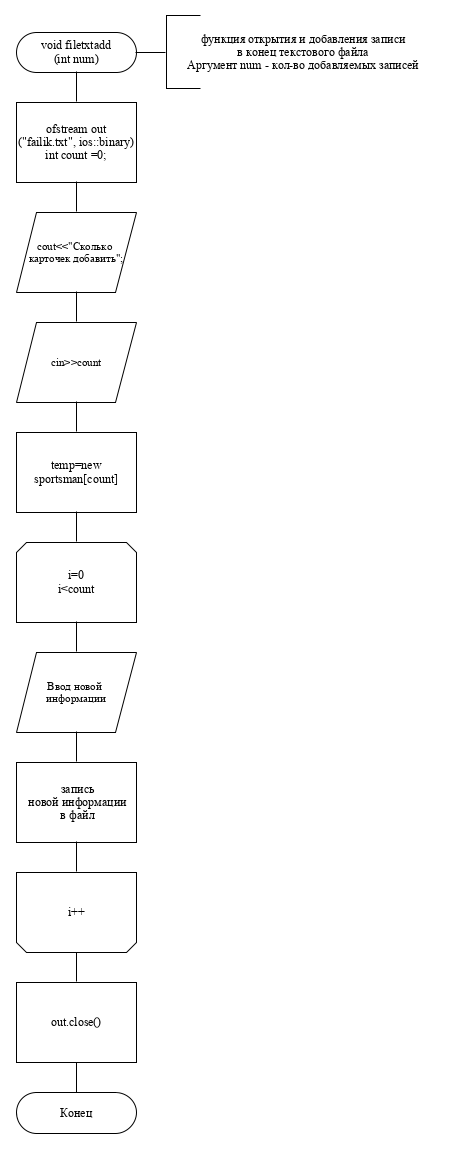
Блок-схемы:

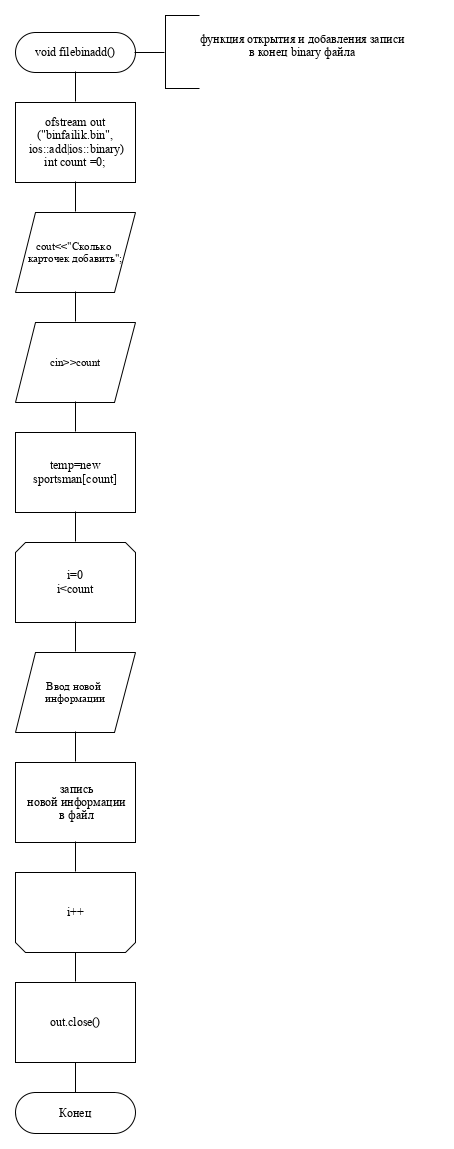












**Код программы:**

#include<iostream>

#include<stdio.h>

#include<cstdlib>

#include <string>

#include<iomanip>

#include<fstream>

using namespace std;

struct sportsman

{

enum league

{

LaLIGA = 1,

Bundesliga,

Seria\_A,

League\_1,

PremierFL

};

union countryteam

{

int avail;

char result;

} inf;

char country[30];

char team[30];

char name[30];

char surname[30];

char secondname[30];

int number;

int weight;

int height;

int age;

char liga[30];

unsigned int goal : 7;

};

int menu();

void struct\_create(int& num, sportsman\*& array);

void struct\_output(int num, sportsman\* array);

void struct\_sort(int num, sportsman\* array,int type);

void struct\_delete(int& num, sportsman\* array,int type);

void struct\_OlderThan20(int num, sportsman\* array);

void filetxtcreate(int num, sportsman\* array);

void filebincreate(int num, sportsman\* array);

void filetxtread(int& num, sportsman\*& array);

void filebinread(int& num, sportsman\*& array);

void filetxtadd(int num);

void filebinadd();

int main()

{

system("color F0");

setlocale(NULL, "RUS");

menu();

return 0;

}

int menu()

{

int num = 0;

sportsman\* card = new sportsman[num];

int choice;

int filechoice;

cout << "Работа с текстовым(1)/бинарным(2) файлом: ";

cin >> filechoice;

do

{

cout<< "Выберите пункт: " << endl;

cout << "1. Ввод данных в массив/чтение из файла(если файл уже создан) " << endl;

cout << "2. Вывод содержимого массива " << endl;

cout << "3. Сортировка фамилий спортсменов по алфавитному порядку " << endl;

cout << "4. Удаления записи по заданной фамилии спортсмена " << endl;

cout << "5. Вывод фамилий спортсменов, возраст которых больше 20 лет " << endl;

cout << "6. Добавить запись " << endl;

cout << "Для того чтобы завершить работу нажмите 0 " << endl;

cin >> choice;

switch (choice)

{

case 1:

if (filechoice == 1)

{

filetxtread(num, card);

}

if (filechoice == 2)

{

filebinread(num, card);

}break;

case 2:struct\_output(num, card); break;

case 3:struct\_sort(num, card,filechoice); break;

case 4:struct\_delete(num, card,filechoice); break;

case 5:struct\_OlderThan20(num, card); break;

case 6:

if (filechoice == 1)

{

filetxtadd(num);

}

if (filechoice == 2)

{

filebinadd();

}

break;

};

}

while (choice != 0);

delete[] card;

return 0;

}

void struct\_create(int& num, sportsman\*& array)

{

int size;

cout << "Количество спортсменов: ";

cin >> size;

num = num + size;

sportsman\* temp = new sportsman[num];

for (int i = 0; i < num - size; i++) {

temp[i] = array[i];

}

array = new sportsman[num];

for (int i = 0; i <num - size; i++) {

array[i] = temp[i];

}

delete[] temp;

int goals;

int picker;

cout << endl;

int counter = 1;

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

{

cout << counter << " спортсмен" << endl;

cout << "Страна: ";

cin.ignore();

cin.getline(array[i].country, 30);

cout << "Команда: ";

cin.getline(array[i].team, 30);

cout << "Игровой номер: ";

cin >> array[i].number;

cin.ignore();

cout << "Фамилия: ";

cin.getline(array[i].surname, 30);

cout << "Имя: ";

cin.getline(array[i].name, 30);

cout << "Отчество: ";

cin.getline(array[i].secondname, 30);

cout << "Возраст: ";

cin >> array[i].age;

cout << "Рост(см): ";

cin >> array[i].height;

cout << "Вес(кг): ";

cin >> array[i].weight;

cout << "Трофеи с командой страны (1 - eсть, 2 - нет): ";

cin >> array[i].inf.avail;

if (array[i].inf.avail == 1)

array[i].inf.result = '+';

if (array[i].inf.avail == 2)

array[i].inf.result = '-';

cout << "В какой лиге выступает спортсмен (1 - LaLIGA" << endl << " 2 - Bundesliga 3 - Seria A" << endl << " 4 - League 1 5 - PremierFL): ";

cin >> picker;

switch (picker)

{

case (array[i].LaLIGA):

strcpy\_s(array[i].liga, "LaLIGA");

break;

case (array[i].Bundesliga):

strcpy\_s(array[i].liga, "Bundesliga");

break;

case (array[i].Seria\_A):

strcpy\_s(array[i].liga, "Seria A");

break;

case (array[i].League\_1):

strcpy\_s(array[i].liga, "League 1");

break;

case (array[i].PremierFL):

strcpy\_s(array[i].liga, "Premier FL");

break;

}

cout << "Количество голов: ";

cin >> goals;

array[i].goal = goals;

counter++;

cout << endl << endl;

}

cout << endl;

}

void struct\_output(int num, sportsman\* array)

{

cout << setw(5) << left << "#" << setw(10) << left << "Cтрана:" << setw(10) << left << "Команда:" << setw(10) << left << "Номер:" << setw(10) << left << "Фамилия:" << setw(10) << left << "Имя:" << setw(10) << left << "Отчество:" << setw(10) << left << "Возраст:" << setw(10) << left << "Рост:" << setw(10) << left << "Вес:" << setw(10) << left << "Трофеи:" << setw(18) << left << "Чемпионат:" << setw(20) << left << "Голы:" << endl;

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

cout << setw(5) << left << i + 1;

cout << setw(10) << left << array[i].country;

cout << setw(10) << left << array[i].team;

cout << setw(10) << left << array[i].number;

cout << setw(10) << left << array[i].surname;

cout << setw(10) << left << array[i].name;

cout << setw(10) << left << array[i].secondname;

cout << setw(10) << left << array[i].age;

cout << setw(10) << left << array[i].height;

cout << setw(10) << left << array[i].weight;

cout << setw(10) << left << array[i].inf.result;

cout << setw(18) << left << array[i].liga;

cout << setw(20) << left << array[i].goal;

cout << endl;

}

}

void struct\_sort(int num, sportsman\* array,int type)

{

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

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

if (strcmp(array[j].surname, array[i].surname) < 0)

{

swap(array[i], array[j]);

}

if (type == 1)

{

filetxtcreate(num, array);

}

if (type == 2)

{

filebincreate(num, array);

}

}

void struct\_OlderThan20(int num, sportsman\* array)

{

cout << endl;

cout << "Спортсмены старше 20 лет: " << endl;

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

{

if (array[i].age > 20)

cout << array[i].surname << " " << endl;

}

cout << endl;

}

void struct\_delete(int& num, sportsman\* array,int type)

{

string Surname;

cout << "Введите Фамилию спортсмена, запись о котором нужно удалить: ";

cin >> Surname;

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

{

if (array[i].surname == Surname)

{

num--;

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

array[j] = array[j + 1];

i--;

}

}

sportsman\* newarray = new sportsman[num];

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

newarray[i] = array[i];

array = new sportsman[num];

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

array[i] = newarray[i];

delete[] newarray;

if (type == 1)

filetxtcreate(num, array);

}

void filetxtcreate(int num, sportsman\* array)

{

ofstream fout;

fout.open("failik.txt");

if (fout.is\_open())

{

fout << setw(5) << left << "#" << setw(10) << left << "Cтрана:" << setw(10) << left << "Команда:" << setw(10) << left << "Номер:" << setw(10) << left << "Фамилия:" << setw(10) << left << "Имя:" << setw(10) << left << "Отчество:" << setw(10) << left << "Возраст:" << setw(10) << left << "Рост:" << setw(10) << left << "Вес:" << setw(10) << left << "Трофеи:" << setw(18) << left << "Чемпионат:" << setw(20) << left << "Голы:" << endl;

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

{

fout << setw(5) << left << i + 1;

fout << setw(10) << left << array[i].country;

fout << setw(10) << left << array[i].team;

fout << setw(8) << left << array[i].number;

fout << setw(12) << left << array[i].surname;

fout << setw(10) << left << array[i].name;

fout << setw(10) << left << array[i].secondname;

fout << setw(10) << left << array[i].age;

fout << setw(10) << left << array[i].height;

fout << setw(10) << left << array[i].weight;

fout << setw(15) << left << array[i].inf.result;

fout << setw(18) << left << array[i].liga;

fout << setw(18) << left << array[i].goal;

fout << endl;

}

fout.close();

}

}

void filetxtread(int& num, sportsman\*& array)

{

num = 0;

ifstream fs("failik.txt");

if (fs.peek() == EOF)

{

cout << "Файл пуст" << endl;

struct\_create(num, array);

filetxtcreate(num, array);

}

else

{

int goals;

char temp[1024];

int count;

fs.getline(temp, 1024,'\n');

for (int i = 0; !fs.eof(); i++)

{

num++;

sportsman\* tmp = new sportsman[num - 1];

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

tmp[j] = array[j];

array = new sportsman[num];

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

array[j] = tmp[j];

delete[] tmp;

fs >> count >> array[i].country;

fs >> array[i].team >> array[i].number >> array[i].surname;

fs >> array[i].name >> array[i].secondname;

fs >> array[i].age >> array[i].height >> array[i].weight;

fs >> array[i].inf.result;

fs >> array[i].liga;

fs >>goals;

array[i].goal = goals;

}

num -= 1;

cin.ignore();

cout << "data is eated";

fs.close();

}

}

void filetxtadd(int num)

{

ofstream out("failik.txt", ios\_base::app);

int count, goals, leaga;

cout << "Сколько карточек добавить? ";

cin >> count;

sportsman\* temp = new sportsman[count];

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

{

cout << "Страна: ";

cin.ignore();

cin.getline(temp[i].country, 30);

cout << "Команда: ";

cin.getline(temp[i].team, 30);

cout << "Игровой номер: ";

cin >> temp[i].number;

cin.ignore();

cout << "Фамилия: ";

cin.getline(temp[i].surname, 30);

cout << "Имя: ";

cin.getline(temp[i].name, 30);

cout << "Отчество: ";

cin.getline(temp[i].secondname, 30);

cout << "Возраст: ";

cin >> temp[i].age;

cout << "Рост(см): ";

cin >> temp[i].height;

cout << "Вес(кг): ";

cin >> temp[i].weight;

cout << "Трофеи с командой страны (1 - eсть, 2 - нет): ";

cin >> temp[i].inf.avail;

if (temp[i].inf.avail == 1)

temp[i].inf.result = '+';

if (temp[i].inf.avail == 2)

temp[i].inf.result = '-';

cout << "В какой лиге выступает спортсмен (1 - LaLIGA" << endl << " 2 - Bundesliga 3 - Seria A" << endl << " 4 - League 1 5 - PremierFL): ";

cin >> leaga;

switch (leaga)

{

case (temp[i].LaLIGA):

strcpy\_s(temp[i].liga, "LaLIGA");

break;

case (temp[i].Bundesliga):

strcpy\_s(temp[i].liga, "Bundesliga");

break;

case (temp[i].Seria\_A):

strcpy\_s(temp[i].liga, "Seria A");

break;

case (temp[i].League\_1):

strcpy\_s(temp[i].liga, "League 1");

break;

case (temp[i].PremierFL):

strcpy\_s(temp[i].liga, "Premier FL");

break;

}

cout << "Количество голов: ";

cin >> goals;

temp[i].goal = goals;

out << setw(5) << left << i + 1+num;

out << setw(10) << left << temp[i].country;

out << setw(10) << left << temp[i].team;

out << setw(8) << left << temp[i].number;

out << setw(12) << left << temp[i].surname;

out << setw(10) << left << temp[i].name;

out << setw(10) << left << temp[i].secondname;

out << setw(10) << left << temp[i].age;

out << setw(10) << left << temp[i].height;

out << setw(10) << left << temp[i].weight;

out << setw(15) << left << temp[i].inf.result;

out << setw(18) << left << temp[i].liga;

out << setw(18) << left << temp[i].goal;

out << endl;

}

out.close();

}

void filebincreate(int num, sportsman\* array)

{

ofstream out("binfailik.bin", ios::binary);

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

{

out.write((char\*) & array[i], sizeof(array[i]));

}

out.close();

}

void filebinread(int& num, sportsman\*& array)

{

ifstream in("binfailik.bin", ios::binary);

num = 0;

int goals;

if (in.peek() == EOF)

{

cout << "Данный файл пуст" << endl;

struct\_create(num,array);

filebincreate(num, array);

}

else

{

cout << "Data is in a file" << endl;

for (int i = 0; !in.eof(); i++)

{

num++;

sportsman\* tmp = new sportsman[num - 1];

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

tmp[j] =array[j];

array = new sportsman[num];

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

array[j] = tmp[j];

delete[] tmp;

in.read((char\*) & array[i], sizeof(array[i]));

}

num -= 1;

in.close();

}

}

void filebinadd()

{

ofstream out("binfailik.bin", ios\_base::app | ios::binary);

int count, goals, leaga;

cout << "Сколько карточек добавить? ";

cin >> count;

sportsman\* temp = new sportsman[count];

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

{

cout << "Страна: ";

cin.ignore();

cin.getline(temp[i].country, 30);

cout << "Команда: ";

cin.getline(temp[i].team, 30);

cout << "Игровой номер: ";

cin >> temp[i].number;

cin.ignore();

cout << "Фамилия: ";

cin.getline(temp[i].surname, 30);

cout << "Имя: ";

cin.getline(temp[i].name, 30);

cout << "Отчество: ";

cin.getline(temp[i].secondname, 30);

cout << "Возраст: ";

cin >> temp[i].age;

cout << "Рост(см): ";

cin >> temp[i].height;

cout << "Вес(кг): ";

cin >> temp[i].weight;

cout << "Трофеи с командой страны (1 - eсть, 2 - нет): ";

cin >> temp[i].inf.avail;

if (temp[i].inf.avail == 1)

temp[i].inf.result = '+';

if (temp[i].inf.avail == 2)

temp[i].inf.result = '-';

cout << "В какой лиге выступает спортсмен (1 - LaLIGA" << endl << " 2 - Bundesliga 3 - Seria A" << endl << " 4 - League 1 5 - PremierFL): ";

cin >> leaga;

switch (leaga)

{

case (temp[i].LaLIGA):

strcpy\_s(temp[i].liga, "LaLIGA");

break;

case (temp[i].Bundesliga):

strcpy\_s(temp[i].liga, "Bundesliga");

break;

case (temp[i].Seria\_A):

strcpy\_s(temp[i].liga, "Seria A");

break;

case (temp[i].League\_1):

strcpy\_s(temp[i].liga, "League 1");

break;

case (temp[i].PremierFL):

strcpy\_s(temp[i].liga, "Premier FL");

break;

}

cout << "Количество голов: ";

cin >> goals;

temp[i].goal = goals;

out.write((char\*)&temp[i], sizeof(temp[i]));

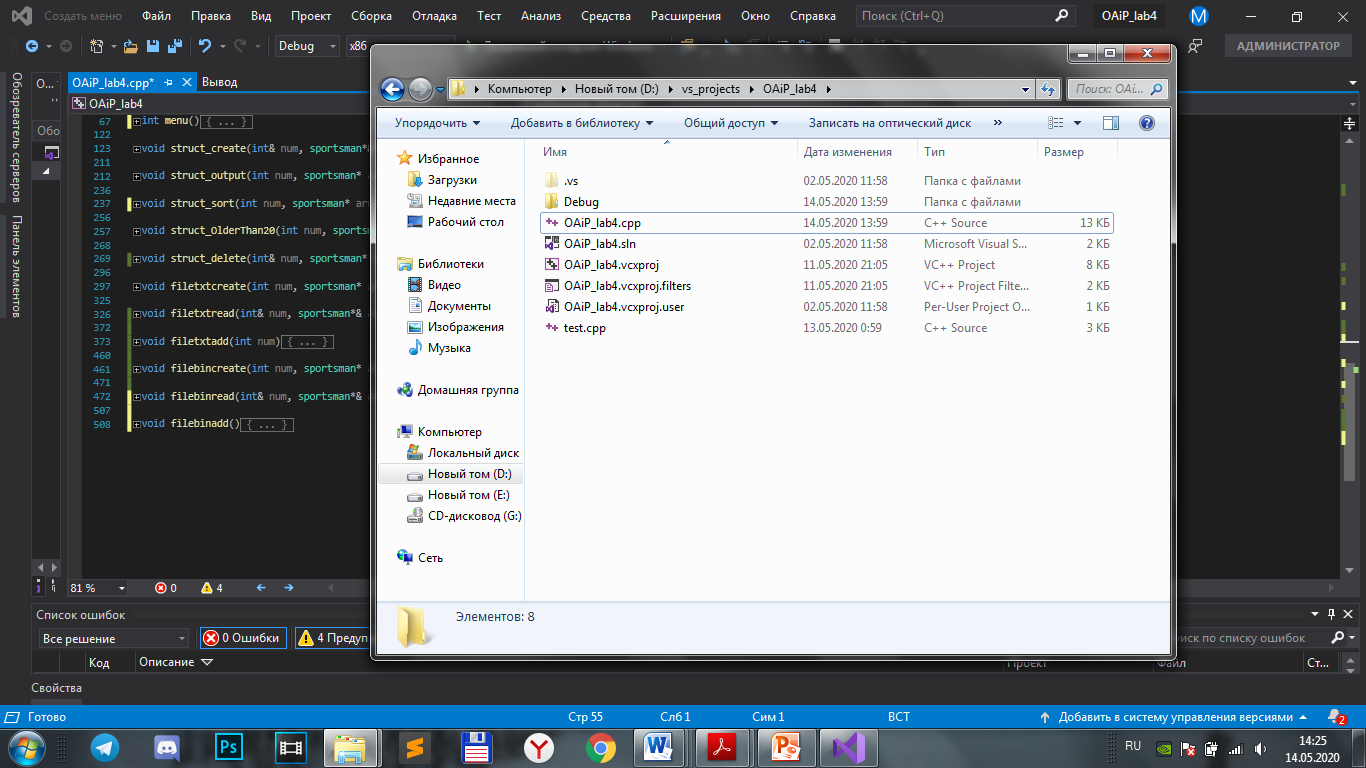
}

out.close();

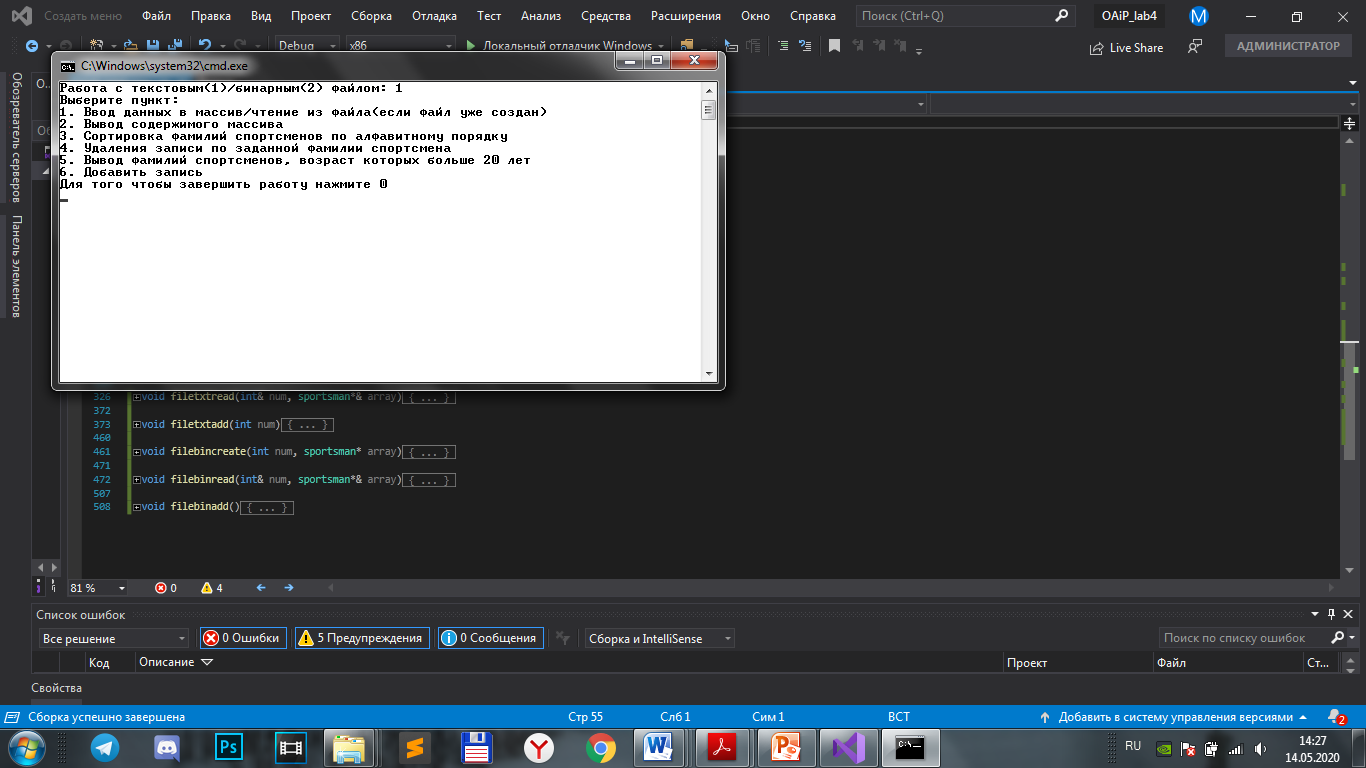
}

До первого запуска:

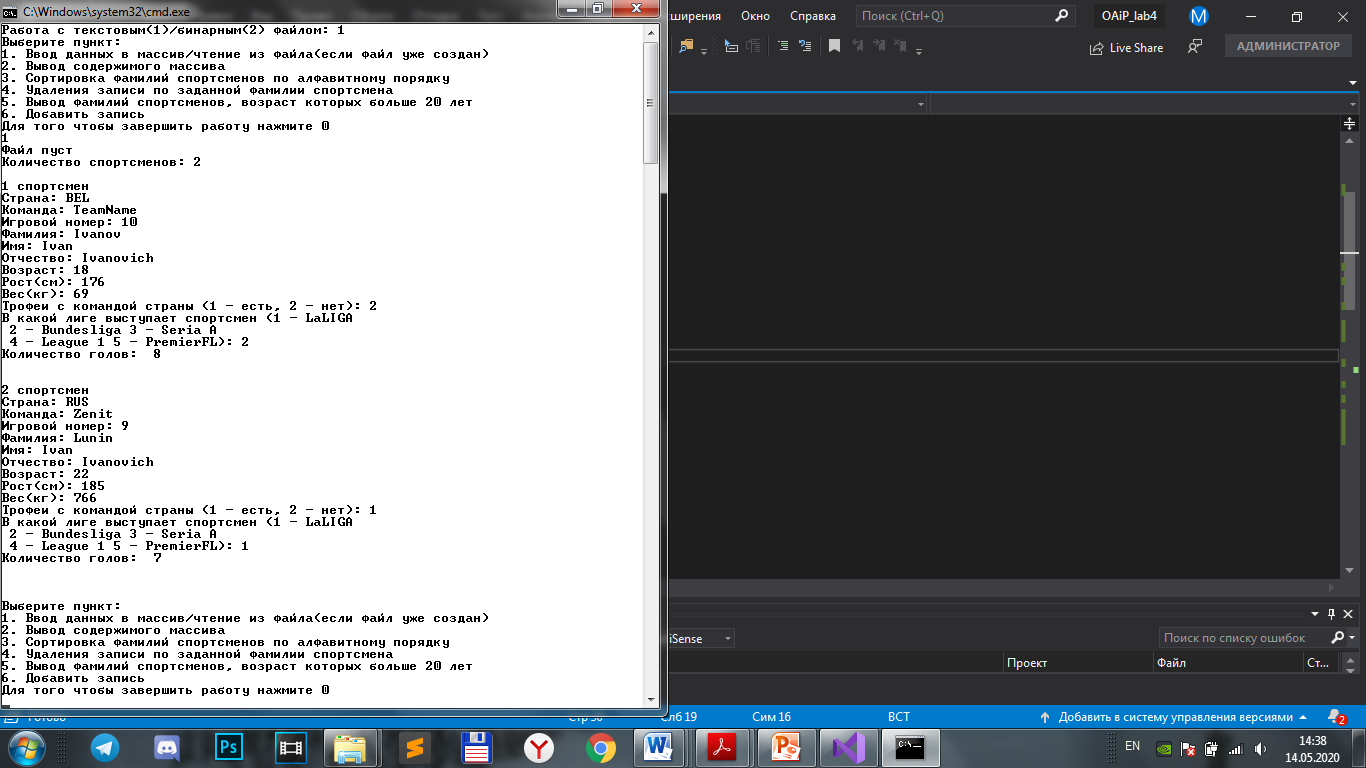
Папка с проектом



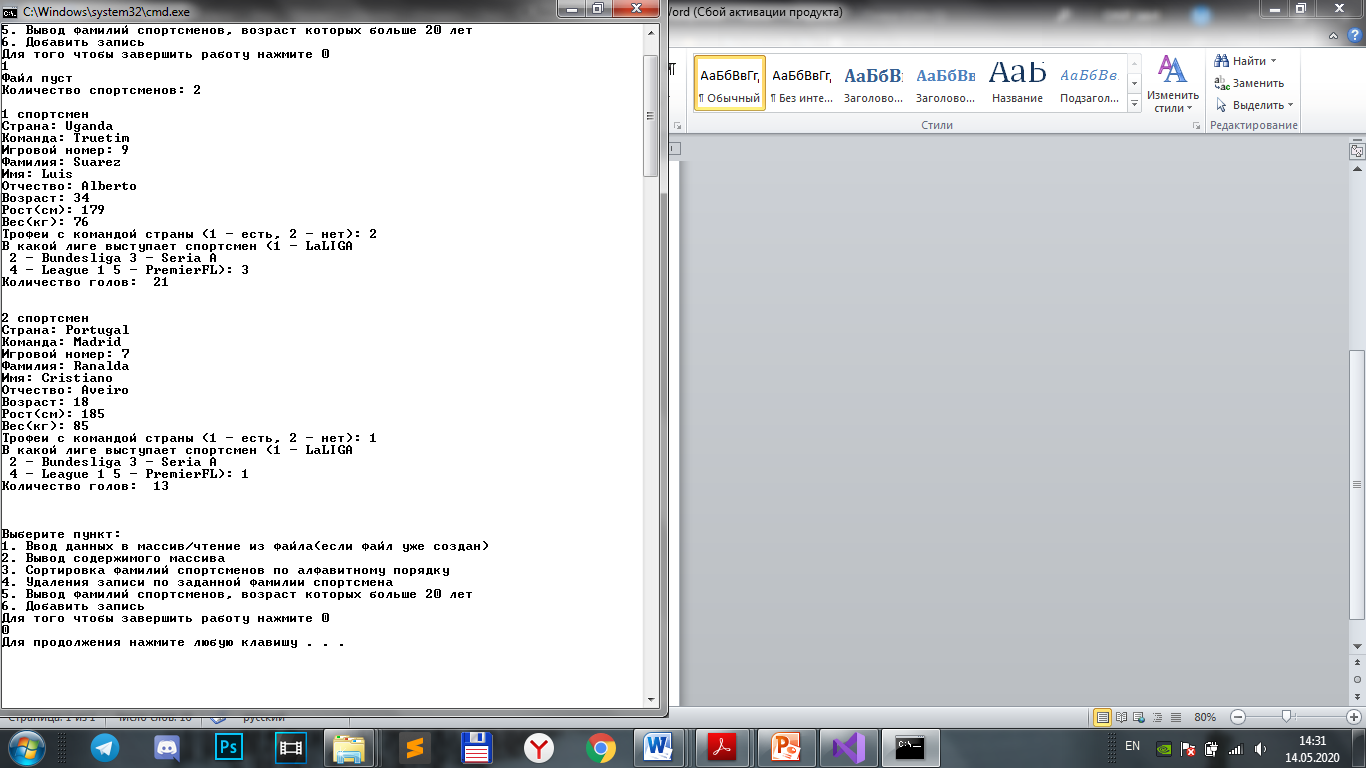
Выбираем работать с текстовым файлом



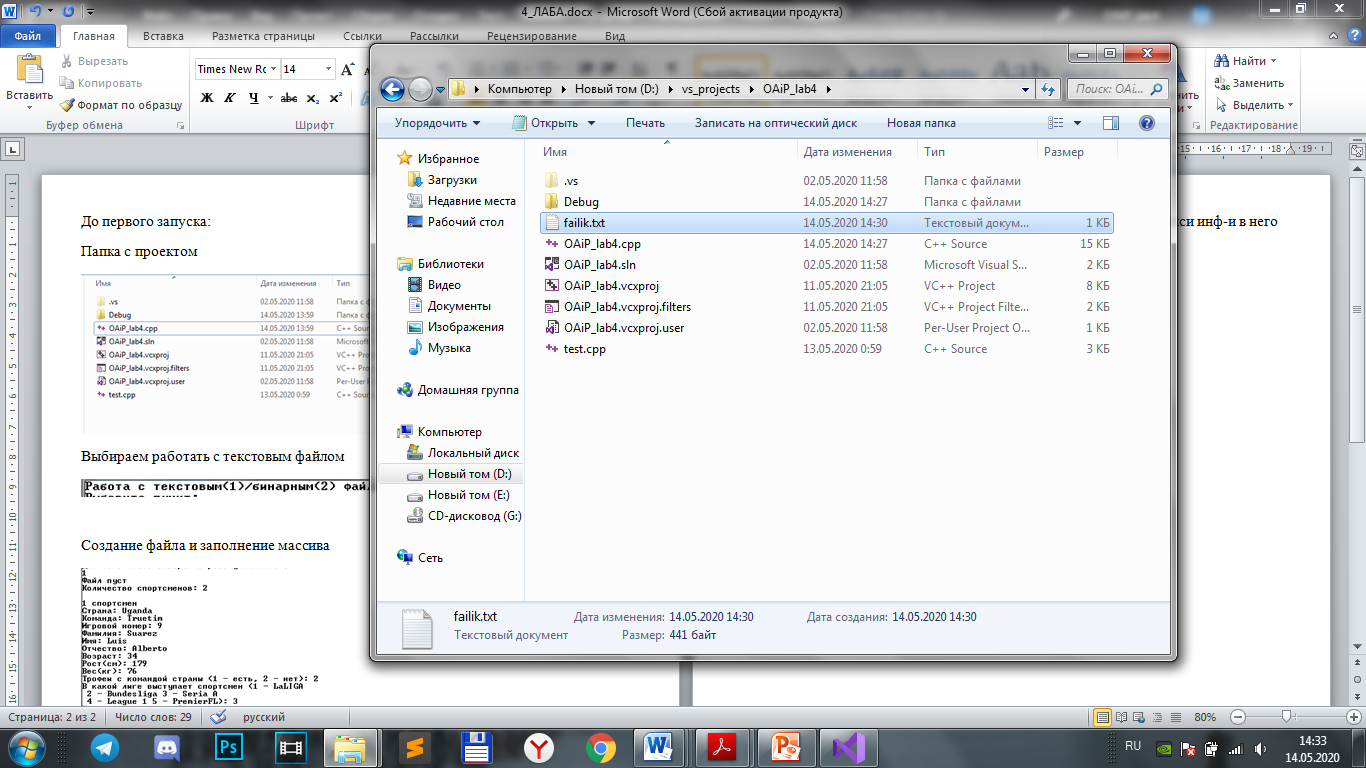
Создание файла и заполнение массива



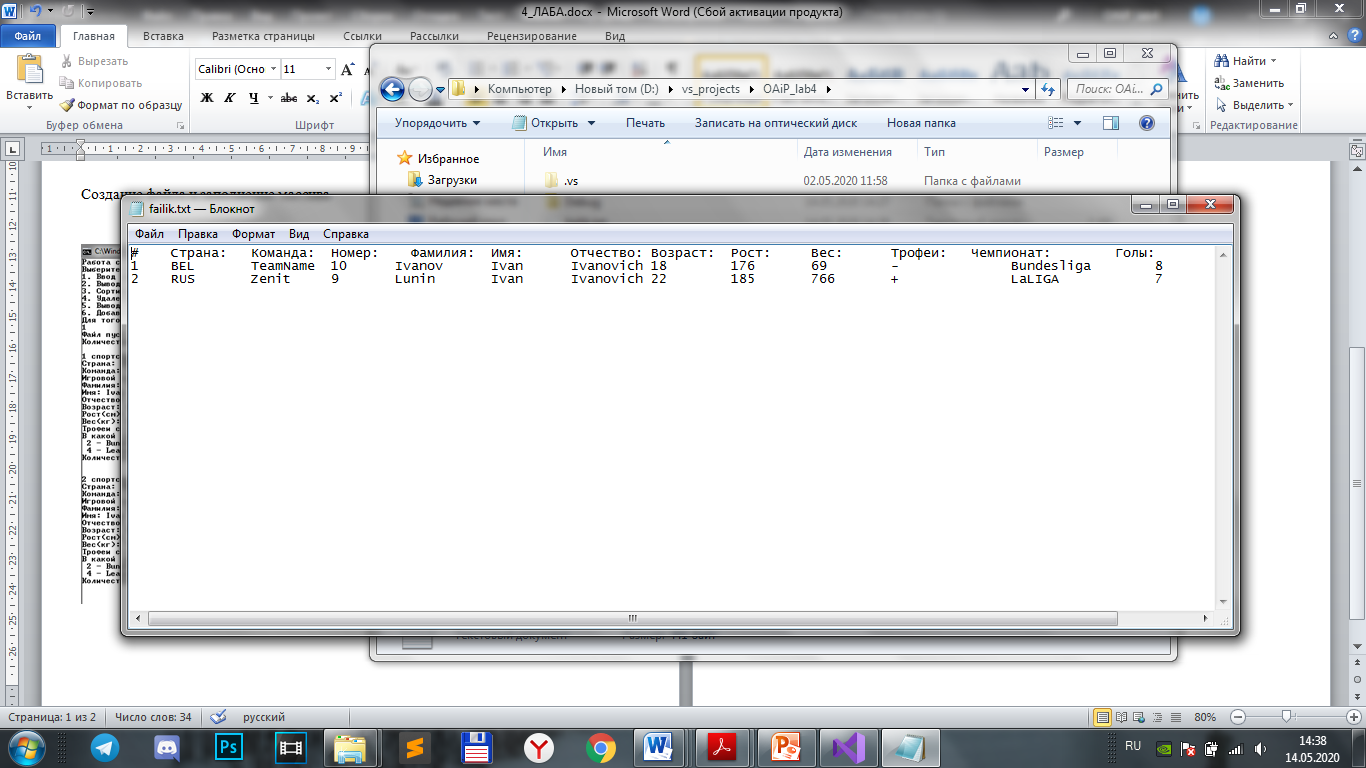
Завершаем работу с программой для проверки создания файла и записи инф-и в него(файл)

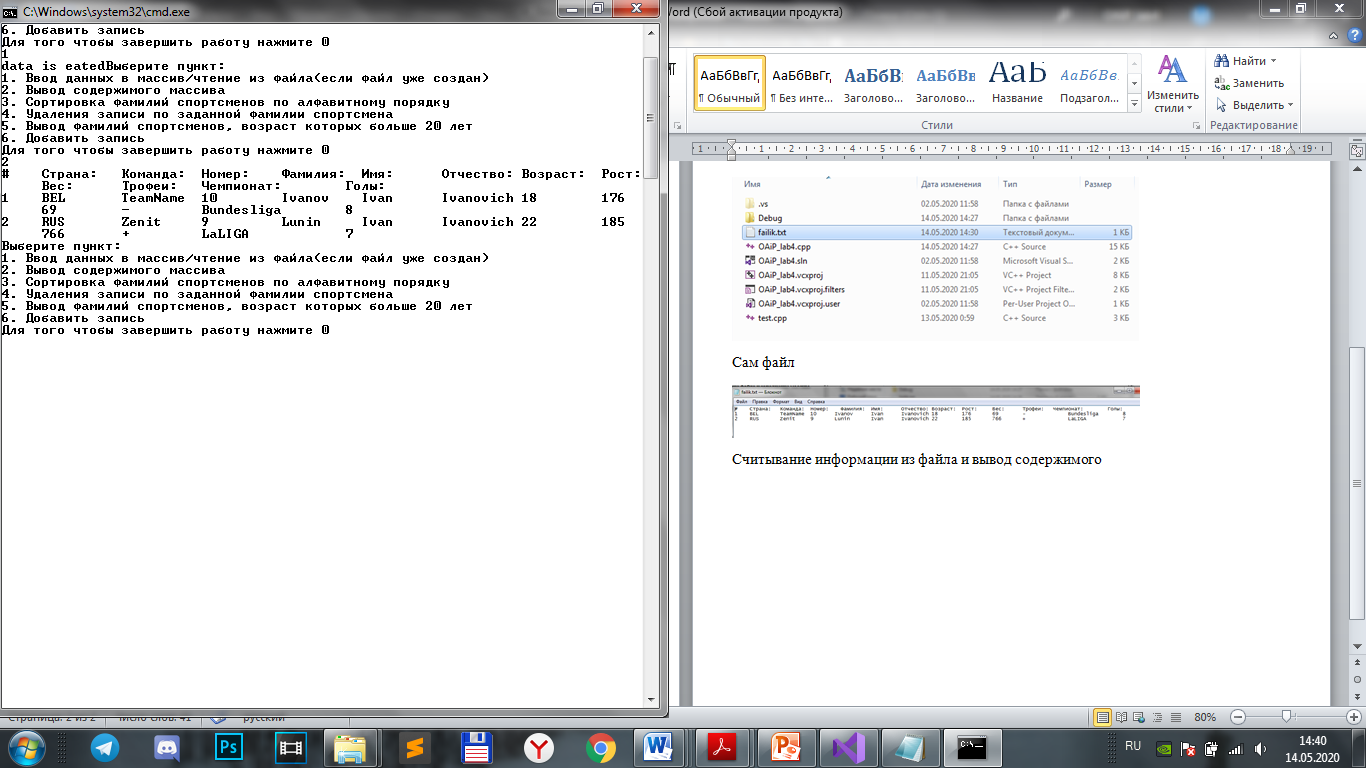


Папка с проектом

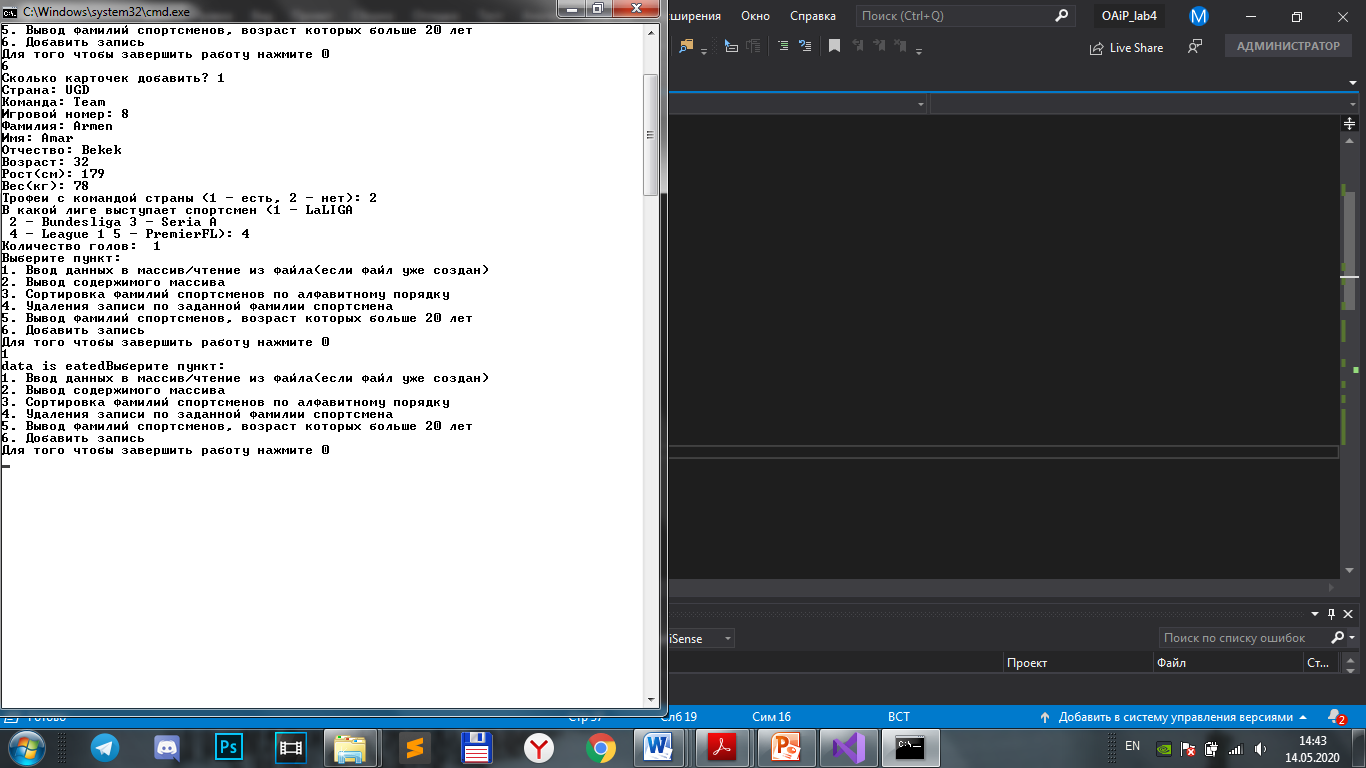


Сам файл

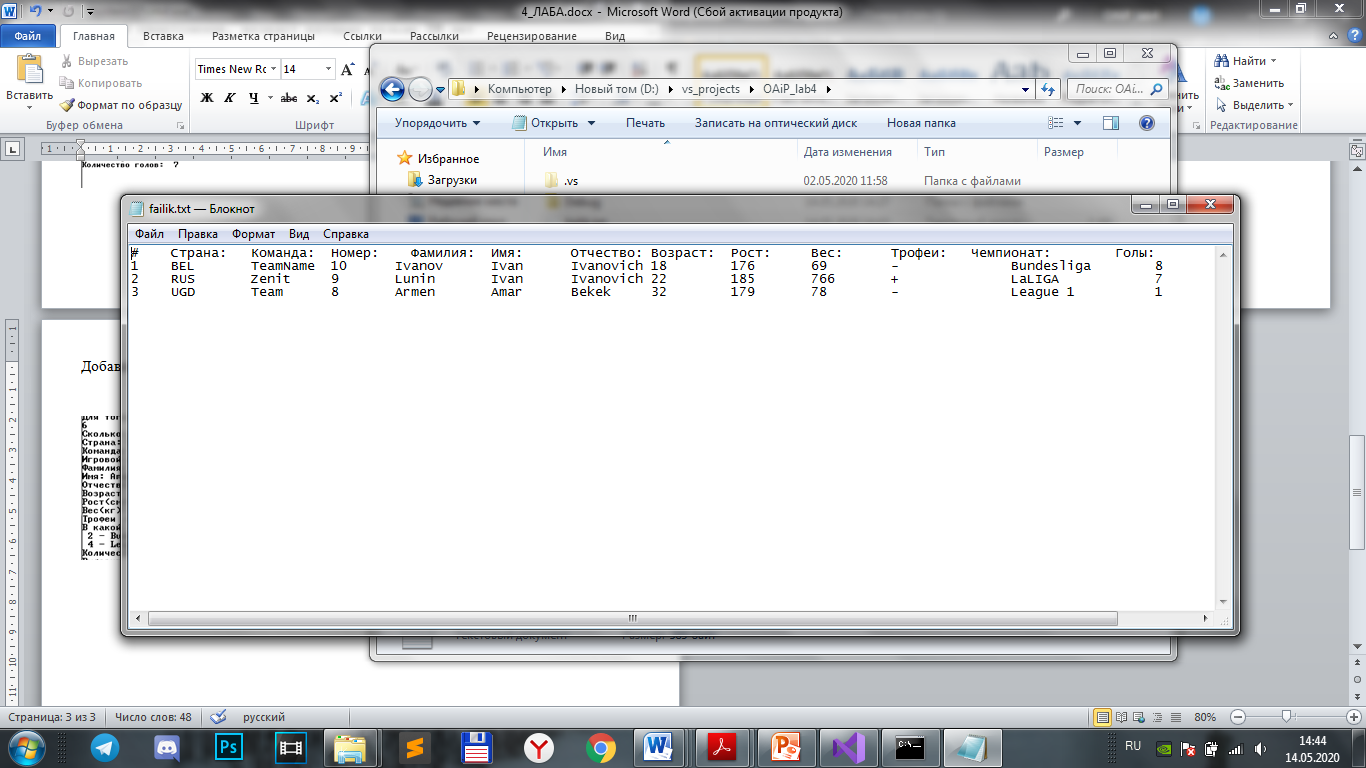


Считывание информации из файла и вывод содержимого после второго запуска

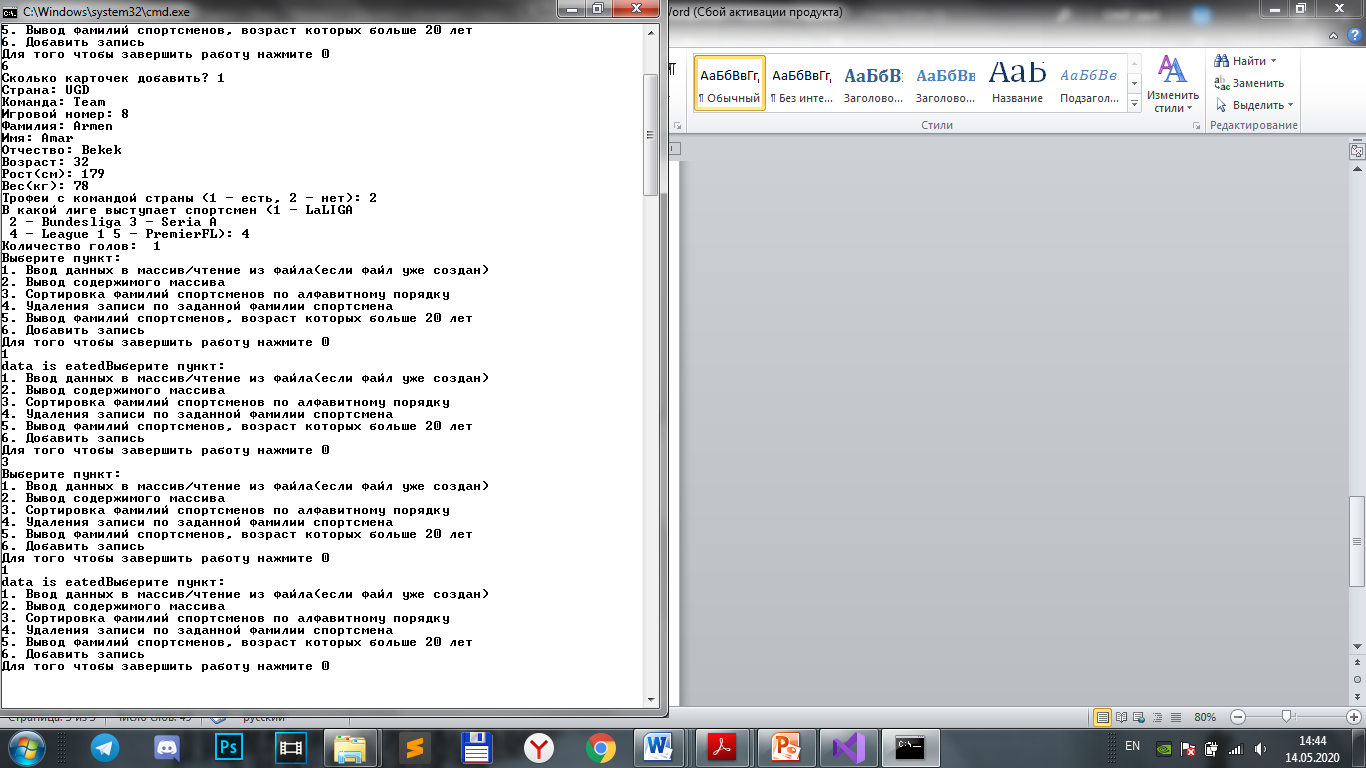
Добавляем запись и считываем ее в файл



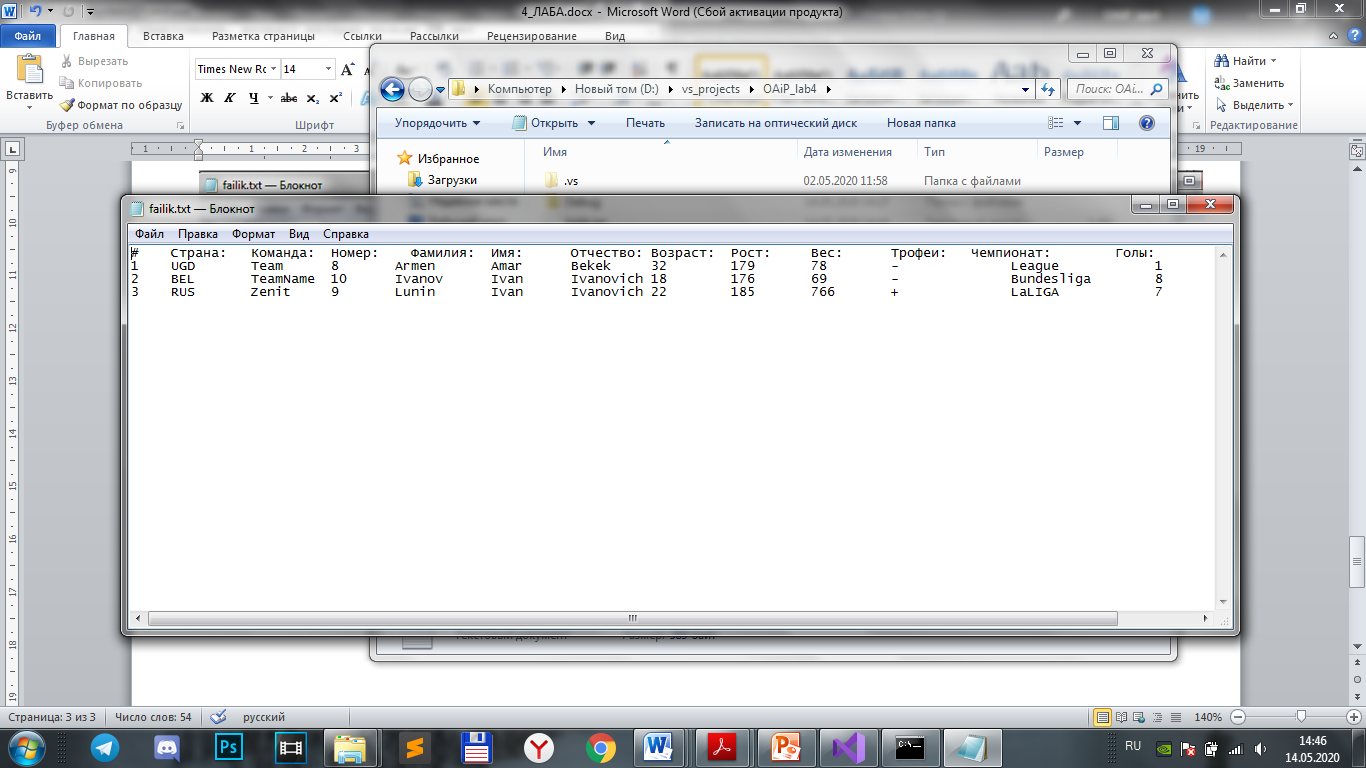
Файл



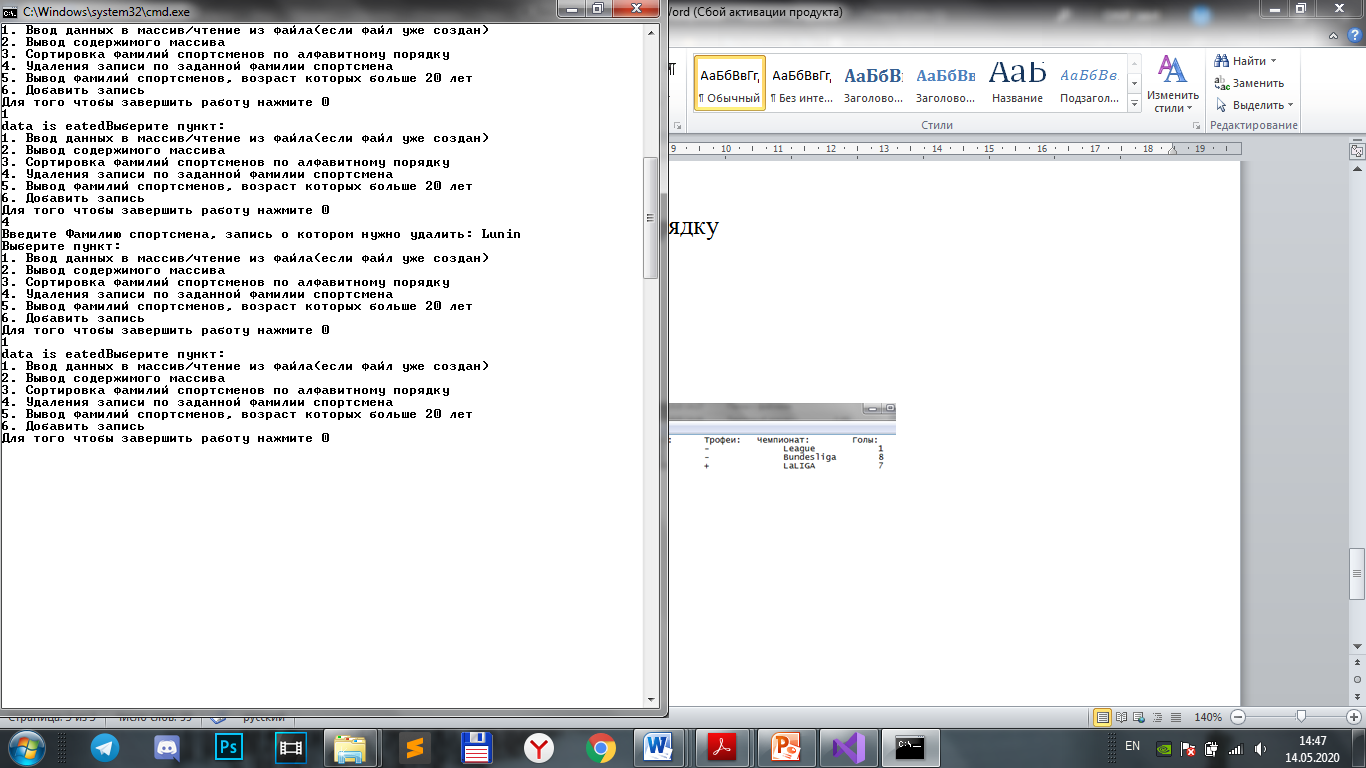
Сортируем фамилии по алфавитному порядку



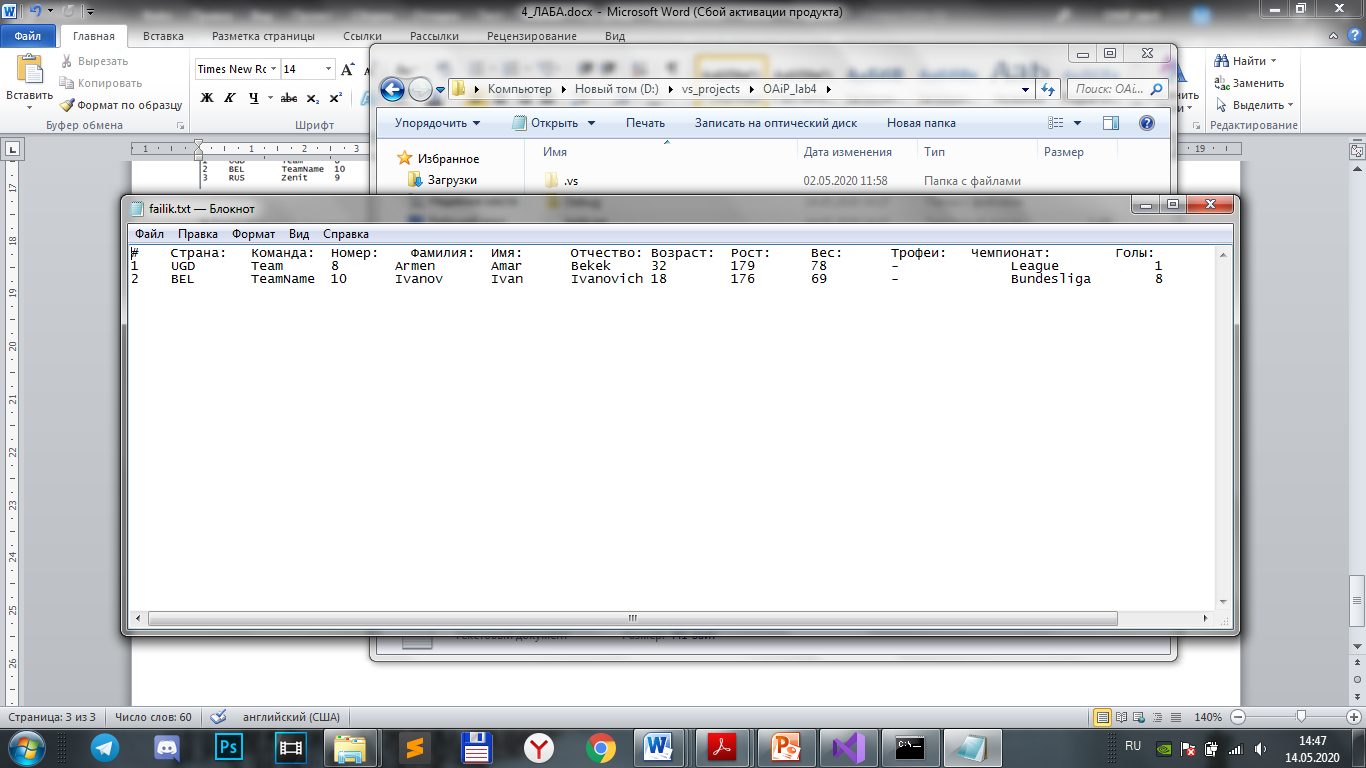
Файл



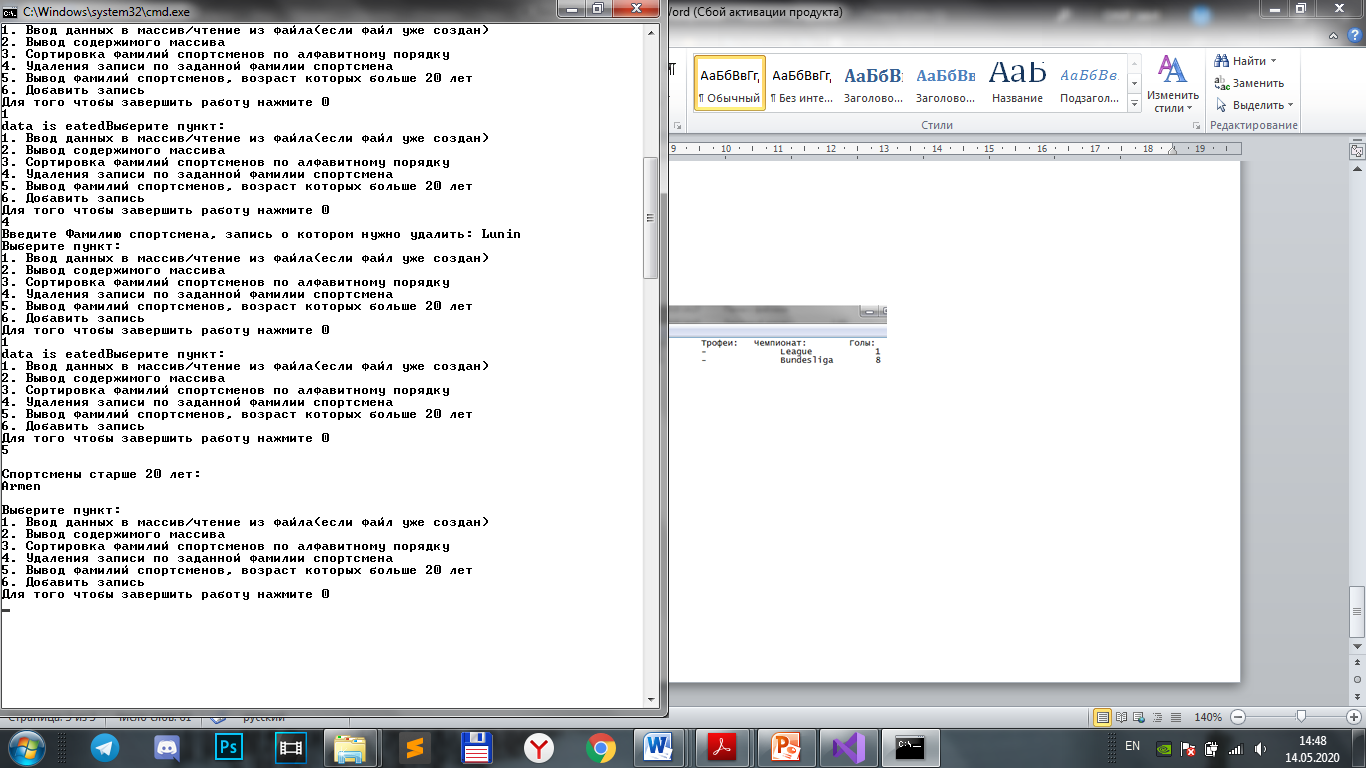
Удаляем запись о спортсмене Lunin



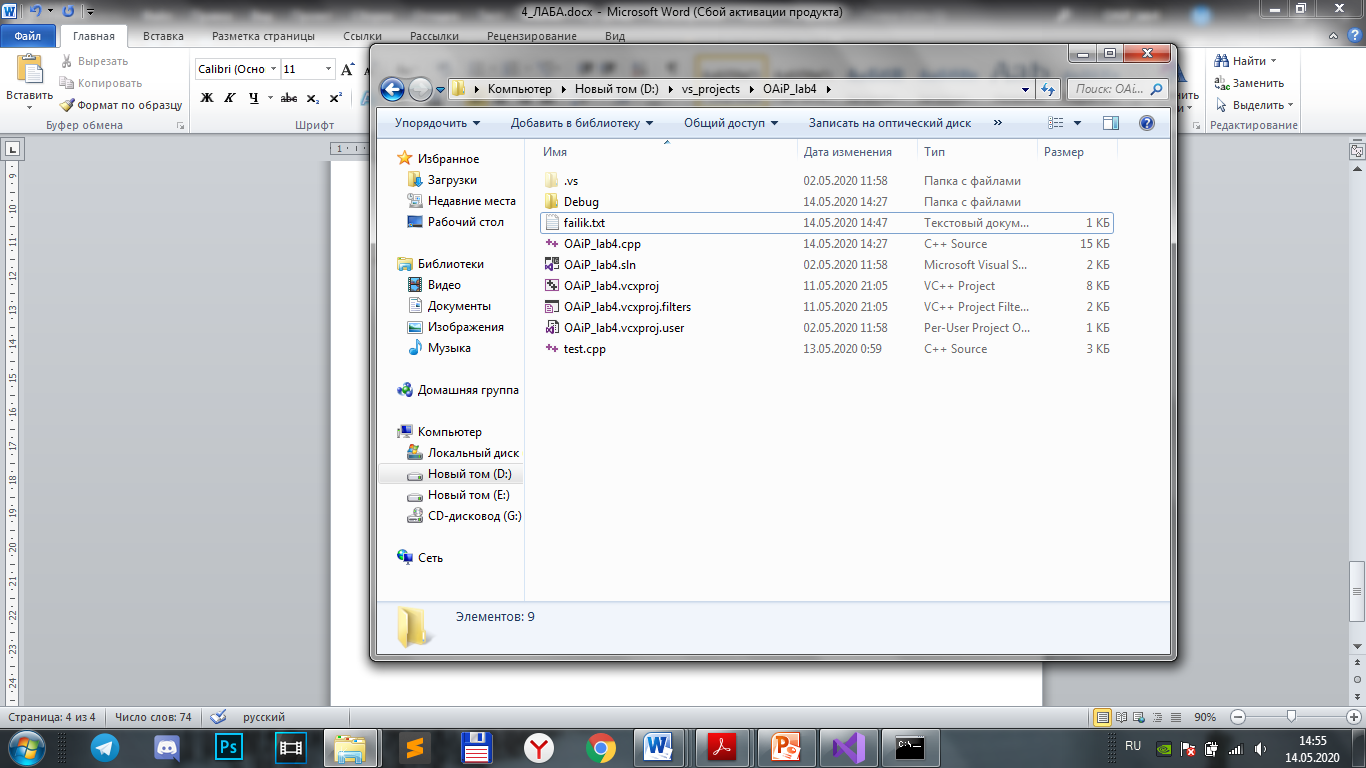
Файл



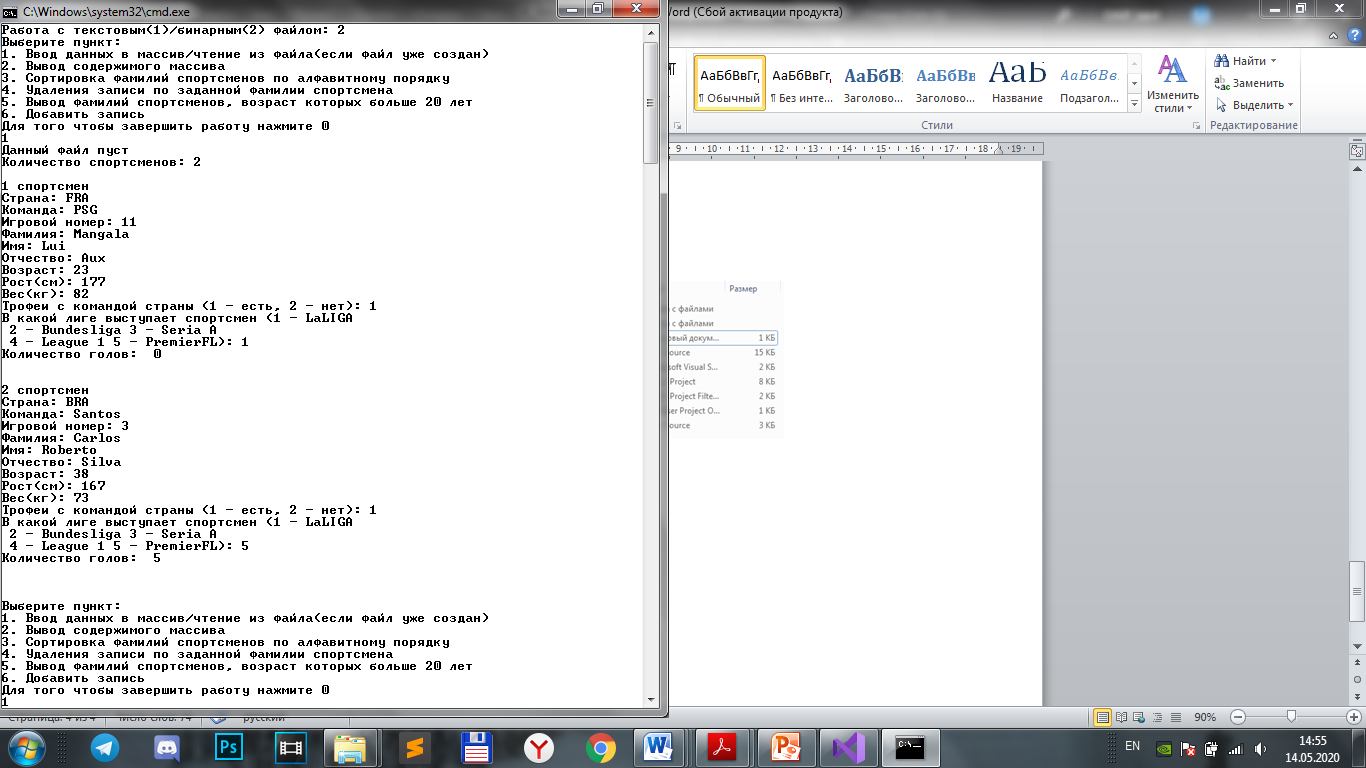
Работоспособность функции вывода спортсменов старше 20 лет



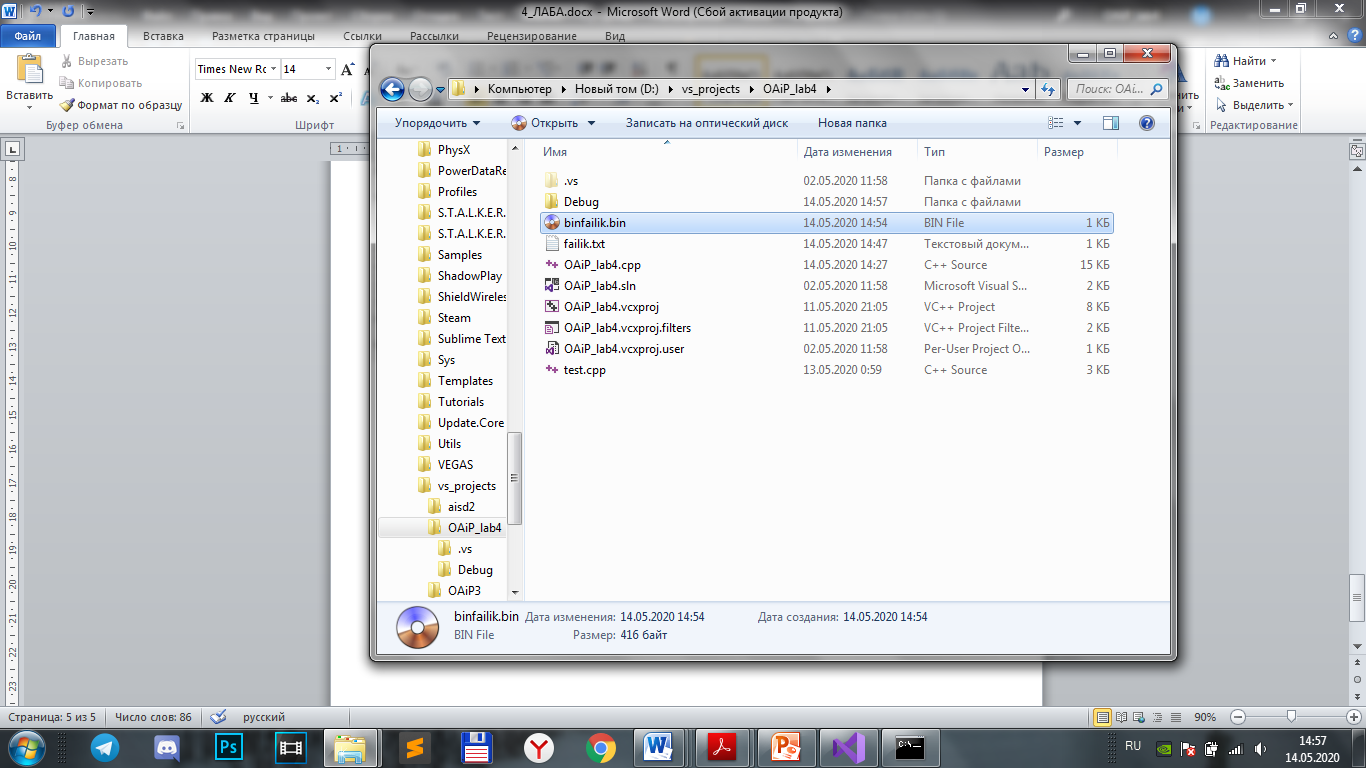
До первого запуска



Выбираем работу с бинарным файлом и создаем массив, записываем его в файл

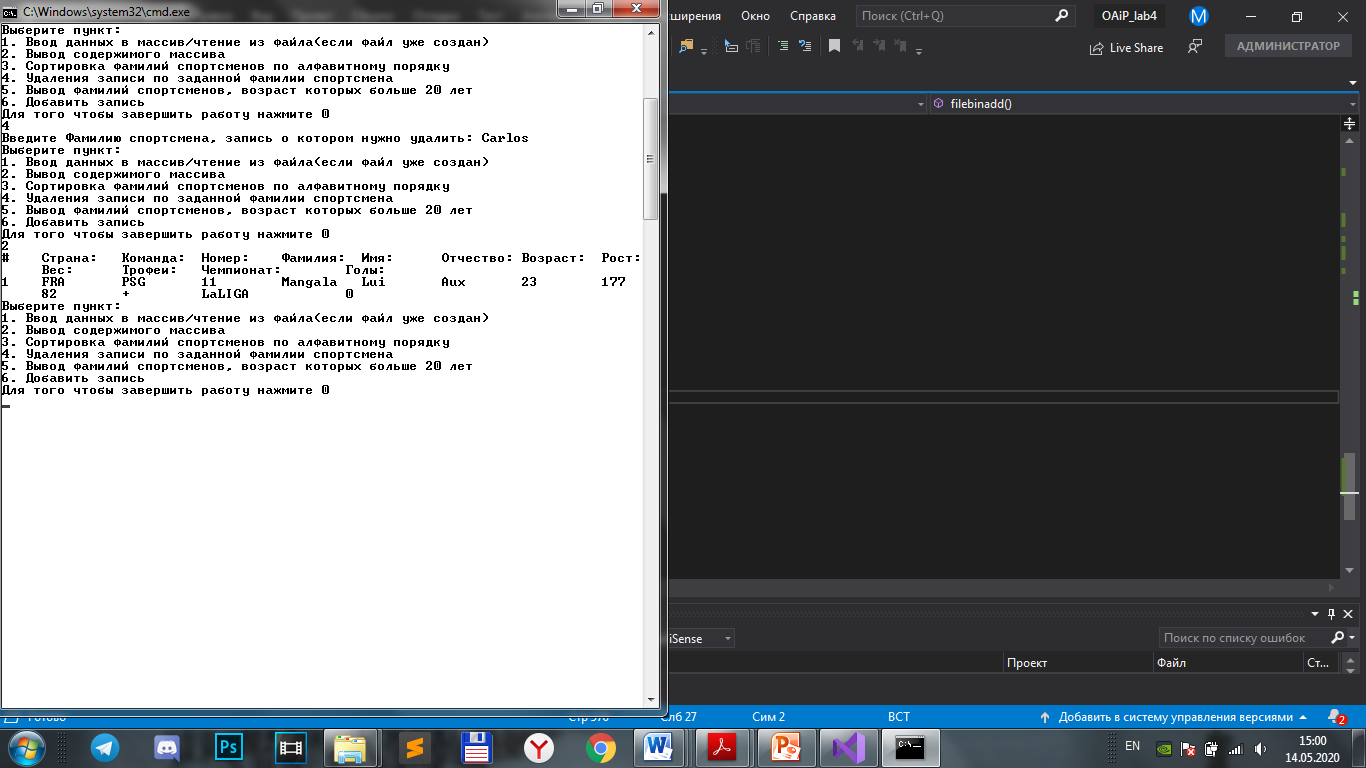


Файл в папке



После второго запуска считываем инфу из файла и записываем в консоль

Удаляем запись о спортсмене Carlos и выводим содержимое файла



**Вывод**: В данном случае не было никакого смысла использовать бинарный тип файлов для хранения информации, т к в этой лабораторной работе я работал с текстовым типом данных и для хранения этих данных идеально подходил текстовый файл.