Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЁТ**

**по лабораторной работе №8**

Дисциплина: «Информатика»

Тема: Структуры данных

Вариант 23

Выполнил:

Студент группы РИС-20-1б

Кузнецов Михаил Сергеевиич

Проверила:

Доцент кафедры ИТАС

Полякова О. А.

**Цель работы**

Цель – написать программу с использованием структур данных и выполнение различных операций с ними.

Для достижения поставленной цели необходимо решить следующие задачи:

* Найти как создаются и объявляются структуры данных.
* Найти как работают структуры данных.
* Научиться считывать информацию из файла и записывать её в структуру.
* Разработать программу по изученному материалу.

**Постановка задачи**

1) Необходимо разработать программу, которая считывает информацию из файла и записывает её в структуру «stadion».

2) Затем необходимо ввести номер элемента, перед которым вставить два дополнительных (заранее заготовленных) элемента структуры. Далее ввести год и удалить те элементы, годы постройки которых меньше заданного.

3) В конце программы обработанная база данных записывается в файл.

**Анализ задачи**

1. Определить какие действия необходимо выполнить для решения задачи.

* Удаление элементов, год постройки которых меньше заданного.
* Добавление двух элементов, перед элементом с заданным номером.
* Поля структуры: название стадиона, год постройки, количество площадок, виды спорта.
* Название стадиона и виды спорта представлены в виде строк, год и количество площадок целыми числами.
* Ввод и вывод осуществляются считыванием и записью данных в файл.
* Все поставленные задачи решаются через циклы for и while, и операторами if/else.

2) Информация из файла хранится в массиве из 5 элементов, информация для записи в файл хранится в массиве из 7 элементов:

stadion stadions[5];

stadion stadionsOut[7];

3) Считывание данных производится через цикл while:

while (!file.eof())

{

getline(file, currentLine);

stadions[stadionInd].name = currentLine;

getline(file, currentLine);

stadions[stadionInd].year = stoi(currentLine);

getline(file, currentLine);

stadions[stadionInd].amountGrounds = stoi(currentLine);

getline(file, currentLine);

stadions[stadionInd].typeSports = currentLine;

getline(file, currentLine);

stadionInd++;

}

4) Вставка элементов через цикл for:

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

{

arrOut[i] = arr[i];

}

arrOut[num - 1].name = "Молния";

arrOut[num - 1].year = 2000;

arrOut[num - 1].amountGrounds = 3;

arrOut[num - 1].typeSports = "Бег Гимнастика Лыжи";

arrOut[num].name = "Локомотив";

arrOut[num].year = 2002;

arrOut[num].amountGrounds = 6;

arrOut[num].typeSports = "Футбол Воллейбол Баскетбол Хоккей";

for (int i = num + 1; i < 7; i++)

{

arrOut[i] = arr[i - 2];

}

5)Так как необходимо удалить элементы, год постройки которых меньше заданного, то такие элементы не будут записаны в файл:

void write\_in\_file(stadion arr[], ofstream &file, int year)

{

for (int i = 0; i < 7; i++)

{

if (arr[i].year >= year)

{

file << "Stadion name: " << arr[i].name << endl;

file << "Year: " << arr[i].year << endl;

file << "Amount Grounds: " << arr[i].amountGrounds << endl;

file << "Type Sports: " << arr[i].typeSports << endl << endl;

}

}

}

**Код**

#include <iostream>

#include <string>

#include <fstream>

using namespace std;

struct stadion

{

string name;

int year = 0;

int amountGrounds = 0;

string typeSports;

};

void read\_file(stadion stadions[], fstream &file)

{

string currentLine;

int stadionInd = 0;

while (!file.eof())

{

getline(file, currentLine);

stadions[stadionInd].name = currentLine;

getline(file, currentLine);

stadions[stadionInd].year = stoi(currentLine);

getline(file, currentLine);

stadions[stadionInd].amountGrounds = stoi(currentLine);

getline(file, currentLine);

stadions[stadionInd].typeSports = currentLine;

getline(file, currentLine);

stadionInd++;

}

}

void print\_struct(stadion stadions[], int num\_elem)

{

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

{

cout << "Element №" << i + 1 << endl;

cout << "Stadion name: " << stadions[i].name << endl;

cout << "Year: " << stadions[i].year << endl;

cout << "Amount Grounds: " << stadions[i].amountGrounds << endl;

cout << "Type Sports: " << stadions[i].typeSports << endl << endl;

}

}

void insert\_elements(stadion arr[], int num, stadion arrOut[])

{

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

{

arrOut[i] = arr[i];

}

arrOut[num - 1].name = "Молния";

arrOut[num - 1].year = 2000;

arrOut[num - 1].amountGrounds = 3;

arrOut[num - 1].typeSports = "Бег Гимнастика Лыжи";

arrOut[num].name = "Локомотив";

arrOut[num].year = 2002;

arrOut[num].amountGrounds = 6;

arrOut[num].typeSports = "Футбол Воллейбол Баскетбол Хоккей";

for (int i = num + 1; i < 7; i++)

{

arrOut[i] = arr[i - 2];

}

}

void write\_in\_file(stadion arr[], ofstream &file, int year)

{

for (int i = 0; i < 7; i++)

{

if (arr[i].year >= year)

{

file << "Stadion name: " << arr[i].name << endl;

file << "Year: " << arr[i].year << endl;

file << "Amount Grounds: " << arr[i].amountGrounds << endl;

file << "Type Sports: " << arr[i].typeSports << endl << endl;

}

}

}

int main()

{

//system("chcp 65001>null");

system("chcp 1251>null");

fstream file ("lab8.txt");

stadion stadions[5];

stadion stadionsOut[7];

int key\_year = 0;

int numDel = 0;

if (file.is\_open())

{

read\_file(stadions, file);

print\_struct(stadions, 5);

do

{

cout << "Введите номер элемента, перед которым необходимо вставить два элемента:" << endl;

cin >> numDel;

cout << endl;

} while (numDel < 1 || numDel > 5);

insert\_elements(stadions, numDel, stadionsOut);

print\_struct(stadionsOut, 7);

do

{

cout << "Введите год, ниже которого все элементы будут удалены" << endl;

cin >> key\_year;

cout << endl;

} while (key\_year < 1);

file.close();

remove("lab8.txt");

ofstream fileOut("lab8.txt");

write\_in\_file(stadionsOut, fileOut, key\_year);

}

else

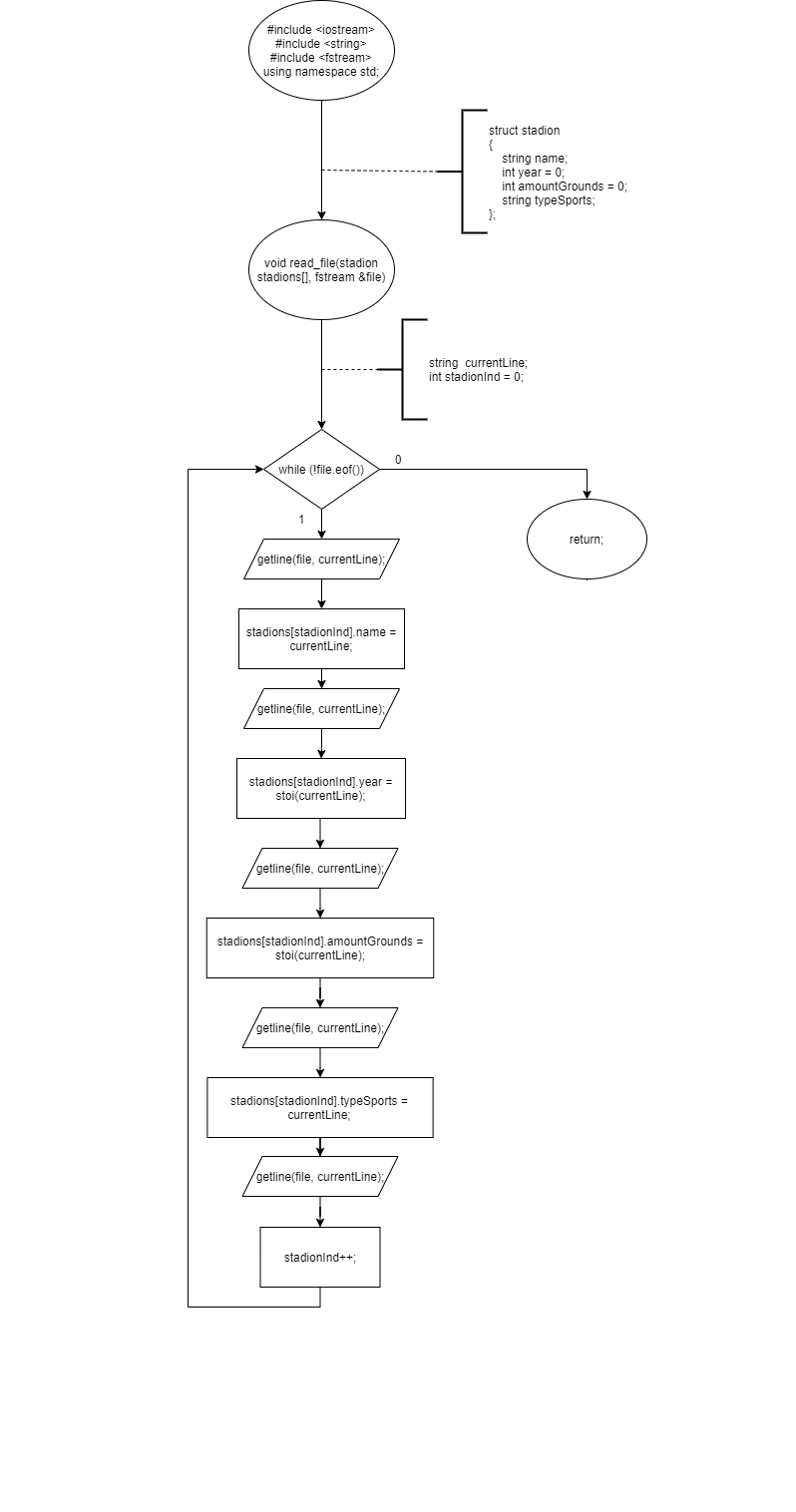
{

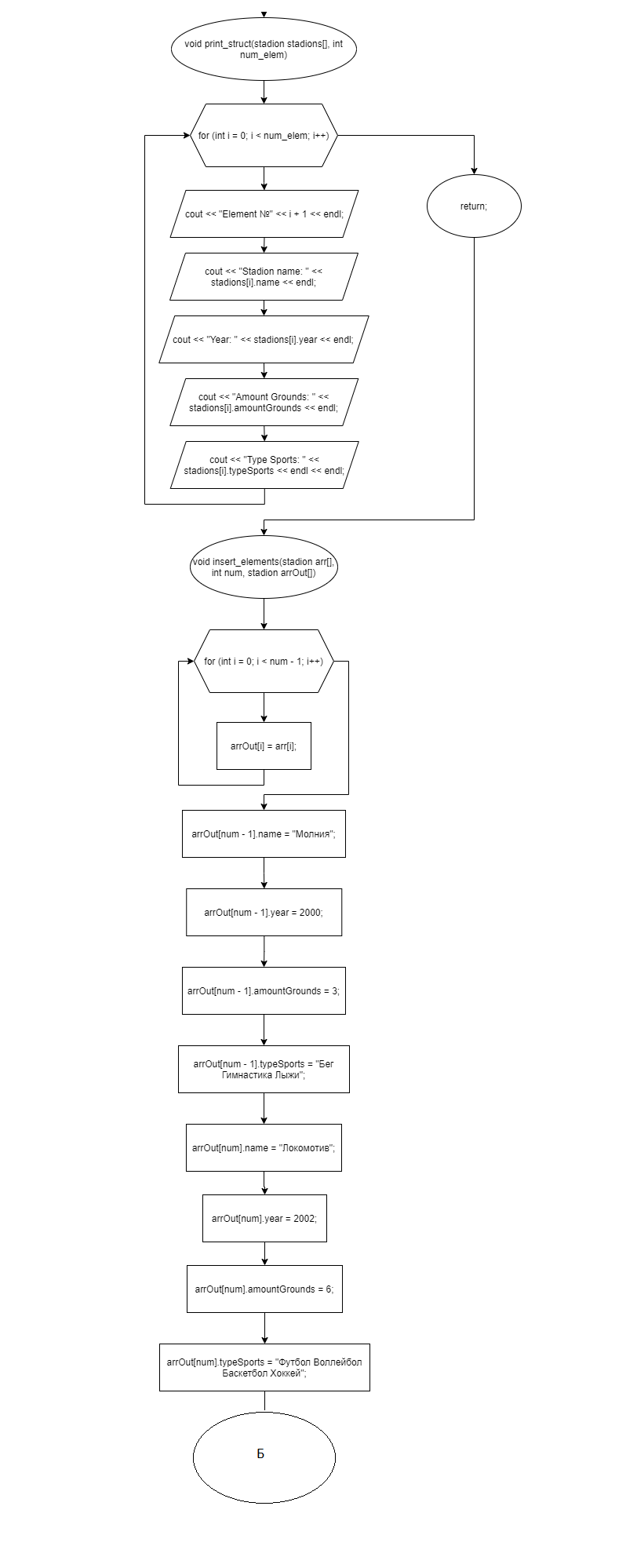
cout << "Файл не найден";

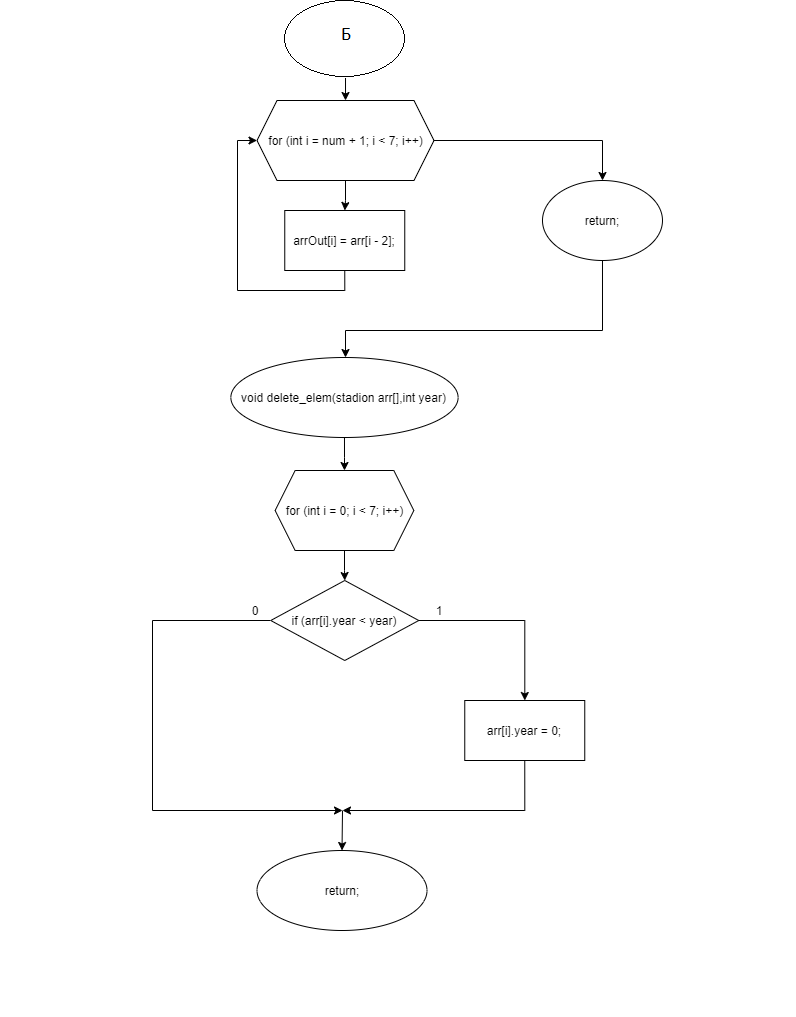
}

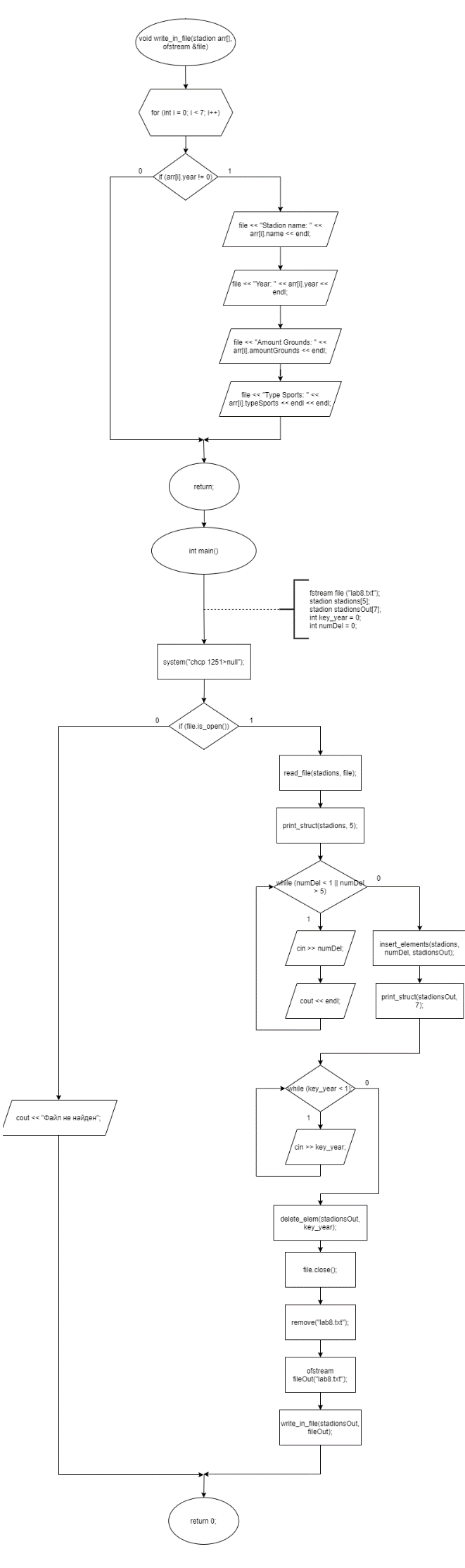
}

**Блок-схема**

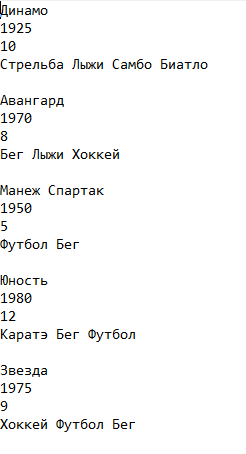
****

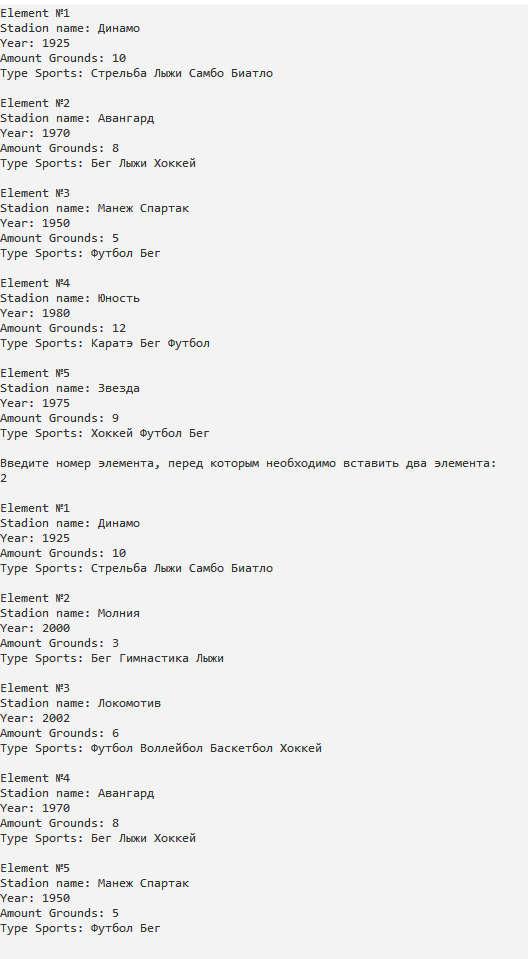
****

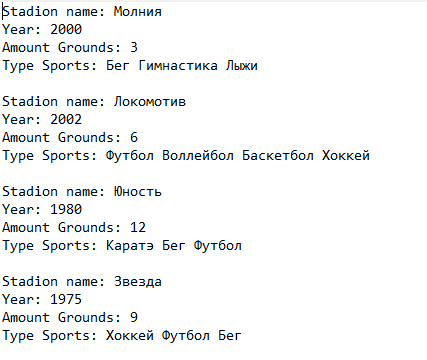
****

****

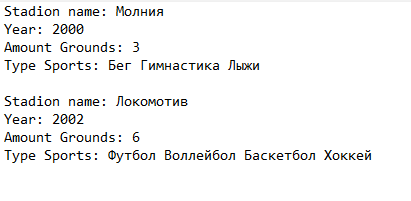
**Скриншоты тестов**

****

****

****

****

****