МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

Высшего образования

«чувашский государственный университет имении И. Н. Ульянова»

Факультет информатики и вычислительной техники

Кафедра компьютерных технологий

РАСЧЕТНО-ГРАФИЧЕСКАЯ РАБОТА

По дисциплине «Информатика и программирование»

Выполнила студент гр.КТ-43-21

Казаков Андрей Юрьевич

Проверил: Алюнов Д.Ю

Вариант № 9

Чебоксары

2022

**Задание:**

Вариант 9

Разработать информационно справочную систему для работы с базой данных. Входные данные:

База данных состоит из записей. (Номер спортсмена, ФИО спортсмен, Лучший результат спортсмена в забеге на 100м., возраст спортсмена, в отдельном файле справочная информация о спортсменах)

Обеспечить:

- введение данных в оперативную память в двунаправленный список:

а) из текстового файла;

б) с клавиатуры;

- просмотр содержимого базы данных;

- редактирование базы данных:

а) удаление записи;

б) добавление записи;

- выведение данных:

а) в текстовый файл;

б) на экран;

- поиск данных по номеру спортсмена;

- сортировка данных по лучшему результату;

- получение справочной информации о спортсмене;

**Программный код:**

#include <iostream>

#include <list>

#include <string>

#include <fstream>

#include <windows.h> //ру

using namespace std;

struct Sportsmens

{

    int number;

    string FIO;

    int result;

    int age;

    string spravka;

};

void menu(list<Sportsmens> sportList, list<Sportsmens>::iterator iSp);

void Ending(list<Sportsmens> sportList, list<Sportsmens>::iterator iSp);

void InfoBD(list<Sportsmens> sportList, list<Sportsmens>::iterator iSp)

{

    cout << "\n---------------------Информация о БД---------------------" << endl;

    int x=0;

    for (iSp = sportList.begin(); iSp != sportList.end(); iSp++)

        x++;

    cout << "Всего записей: " << x << endl;

    menu(sportList, iSp);

}

void Add(list<Sportsmens> sportList, list<Sportsmens>::iterator iSp)

{

    cout << "\n---------------------Добавление---------------------" << endl;

    Sportsmens newSportsmen;

    cout << "Введите номер спортсмена: " << endl;

    (cin >> newSportsmen.number).get();

    cout << "Введите ФИО спортсмена: " << endl;

    getline(cin, newSportsmen.FIO);

    cout << "Введите лучший результат спортсмена: " << endl;

    cin >> newSportsmen.result;

    cout << "Введите возраст спортсмена: " << endl;

    (cin >> newSportsmen.age).get();

    cout << "Введите справку о нём: " << endl;

    getline(cin, newSportsmen.spravka);

    sportList.push\_back(newSportsmen);

    menu(sportList, iSp);

}

void Del(list<Sportsmens> sportList, list<Sportsmens>::iterator iSp)

{

    cout << "\n---------------------Удаление---------------------" << endl;

    cout << "Номера спортсменов: ";

    for (iSp = sportList.begin(); iSp != sportList.end(); iSp++)

        cout << iSp->number << " ";

    cout << "\nВведите номер спортсмена для удаления" << endl;

    int x, y; cin >> x;

    for (iSp = sportList.begin(); iSp != sportList.end(); iSp++)

    {

        if (iSp->number == x)

        {

            sportList.erase(iSp);

            cout << "удалено" << endl;

            break;

        }

    }

    menu(sportList, iSp);

}

void Print(list<Sportsmens> sportList, list<Sportsmens>::iterator iSp)

{

    cout << "\n---------------------Вывод---------------------" << endl;

    cout << "Номер \tФИО\t\t\t\tРез-тат\tВозраст" << endl;

    for (iSp = sportList.begin(); iSp != sportList.end(); iSp++) {

        cout << iSp->number << "\t" << iSp->FIO << "\t" << iSp->result << "\t" << iSp->age<< endl;

    }

    menu(sportList, iSp);

}

//сохранение

void SaveList(list<Sportsmens> sportList, list<Sportsmens>::iterator iSp, int end) //end необходим при выходе из программы

{

    ofstream o\_sport("sportsmens.txt");

    ofstream o\_spravka("spravka.txt");

    for (iSp = sportList.begin(); iSp != sportList.end(); iSp++) {

        o\_sport << iSp->number << ";" << iSp->FIO << ";" << iSp->result << ";" << iSp->age << ";" << endl;

        o\_spravka << iSp->spravka << endl;

    }

    cout << "\nСохранено" << endl;

    if (end == 1) exit(2);

    menu(sportList, iSp);

}

void Search(list<Sportsmens> sportList, list<Sportsmens>::iterator iSp)

{

    cout << "\n---------------------Поиск по номеру---------------------" << endl;

    cout << "Введите номер спортсмена: ";

    int x; cin >> x;

    for (iSp = sportList.begin(); iSp != sportList.end(); iSp++)

        if (iSp->number == x)

            cout << iSp->number << " " << iSp->FIO << " " << iSp->result << " " << iSp->age << endl;

    menu(sportList, iSp);

}

//компараторы для сортировки

bool bigger(const Sportsmens& a, const Sportsmens& b)

{

    return a.result > b.result;

}

bool smaller(const Sportsmens& a, const Sportsmens& b)

{

    return a.result < b.result;

}

void SortResult(list<Sportsmens> sportList, list<Sportsmens>::iterator iSp) {

    cout << "\n-----------------Сортировка по результату-----------------" << endl;

    if (sportList.size() == 1)

        cout << "невозможно отсоритровать (размер БД = 1)" << endl;

    else

    {

        cout << "по наибольшему 1\nпо наименьшему 0" << endl;

        int x; cin >> x;

        if (x == 1)

        {

            iSp = sportList.begin();

            sportList.sort(bigger);

        }

        else

        {

            sportList.sort(smaller);

        }

    }

    menu(sportList, iSp);

}

void GetSpravka(list<Sportsmens> sportList, list<Sportsmens>::iterator iSp)

{

    cout << "\n--------------Получение справки о спортсмене--------------" << endl;

    cout << "Номера спортсменов: ";

    for (iSp = sportList.begin(); iSp != sportList.end(); iSp++)

        cout << iSp->number << " ";

    cout << "\nВведите номер спортсмена: ";

    int x; cin >> x;

    for (iSp = sportList.begin(); iSp != sportList.end(); iSp++)

        if (iSp->number == x) {

            cout << "ФИО: " << iSp->FIO << "\nЛучший результат " << iSp->result << "\nВозраст " << iSp->age << endl;

            cout << "Справка: " << iSp->spravka << endl;

            break;

        }

    menu(sportList, iSp);

}

void Ending(list<Sportsmens> sportList, list<Sportsmens>::iterator iSp)

{

    cout << "\nЖелаете сохранить перед выходом? (1/0) " << endl;

    int x; cin >> x;

    if (x==1) SaveList(sportList, iSp, 1);

    exit(2);

}

void menu(list<Sportsmens> sportList, list<Sportsmens>::iterator iSp) {

    cout << "\n---------------------Меню---------------------" << endl;

    cout << "Вывести информацию о БД(0)" << endl

        << "Редактировать:" << endl

        << "\tДобавить запись(1)" << endl

        << "\tУдалить запись(2)" << endl

        << "Вывести данные :" << endl

        << "\tсохранить в текстовый файл(3)" << endl

        << "\tна экран(4)" << endl

        << "Поиск данных спортсмена по номеру(5)" << endl

        << "Сортировка по лучшему результату(6)" << endl

        << "Получение справочной информации о спортсмене(7)" << endl

        << "Выйти из программы(8)" << endl;

    int x; cin >> x;

    switch (x)

    {

    case 0: InfoBD(sportList, iSp);

    case 1: Add(sportList, iSp);

    case 2: Del(sportList, iSp);

    case 3: SaveList(sportList, iSp, 0);

    case 4: Print(sportList, iSp);

    case 5: Search(sportList, iSp);

    case 6: SortResult(sportList, iSp);

    case 7: GetSpravka(sportList, iSp);

    case 8: Ending(sportList, iSp);

    default: cout << "нет такой команды" << endl;

    };

};

int main()

{

    setlocale(LC\_ALL, "RUS");

    SetConsoleCP(1251);

    SetConsoleOutputCP(1251);

    list <Sportsmens> sportList;

    list <Sportsmens>::iterator iSp;

    ifstream f\_sport("sportsmens.txt");

    ifstream f\_spravka("spravka.txt");

    if (f\_sport and f\_spravka)

    {

        cout << "\* чтение из файла осуществляется во время запуска" << endl;

        cout << "sportsmens.txt и spravka.txt прочитаны" << endl;

        string line;

        Sportsmens human;

        while (getline(f\_sport, line, ';'))

        {

                human.number = stoi(line);

            getline(f\_sport, line, ';');

                human.FIO = line;

            getline(f\_sport, line, ';');

                human.result = stoi(line);

            getline(f\_sport, line, ';');

                human.age = stoi(line);

            if (getline(f\_spravka, line))

                human.spravka = line;

            else

                human.spravka = "None";

            sportList.push\_back(human);

            f\_sport.ignore(1, '\n');

        }

    }

    else

    {

        return 1;

        cout << "Не удалось открыть" << endl;

    }

    menu(sportList, iSp);

    f\_sport.close();

    f\_spravka.close();

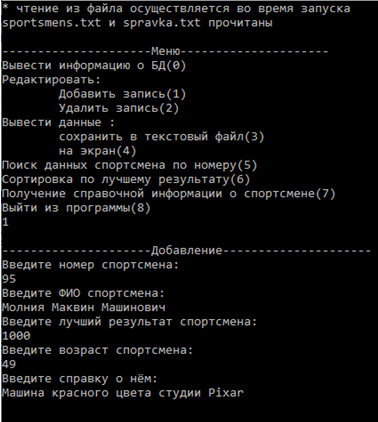
    return 0;

}

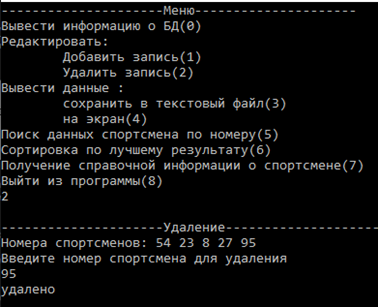
**Скрины**

Редактирование базы данных:

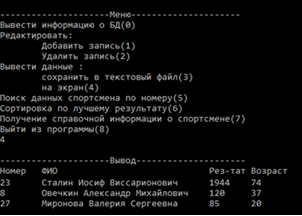
1. Добавление записи:



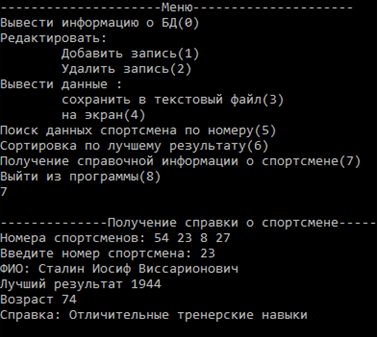
1. Удаление записи:



1. Вывод данных:

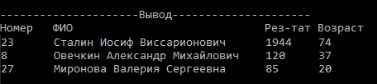


1. Поиск данных по коду спортсмена:

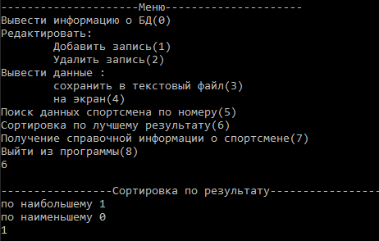


1. Сортировка данных по результату:

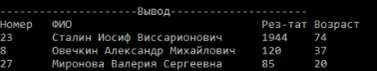
Было:



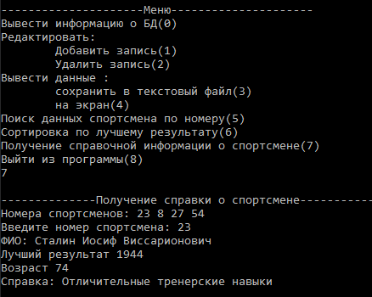
Ввод команды:



Результат:



1. Получение справочной информации о спортсмене:



1. Выход:

