МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования «Южно-Уральский государственный университет»

(национальный исследовательский университет)

Высшая школа электроники и компьютерных наук

Кафедра системного программирования

Создание динамических баз данных

(название практической работы)

ОТЧЕТ

по практической работе № 8

по дисциплине «Функциональное и логическое программирование»

Выполнил:

студент группы КЭ–401

/ Д.А. Иванов /

(подпись)

« » 2024 г.

Проверил:

преподаватель кафедры СП

/ М.А. Ческидова /

(подпись)

« » 2024 г.

Оглавление

[1. Задание 1 3](#_Toc179839618)

[2. Задание 2 5](#_Toc179839619)

[3. ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ 7](#_Toc179839620)

# Задание 1

Составьте программу "Футбольная база данных" (имена игроков, названия их команд, номера, рост, вес, количество сыгранных в NFL сезонов). Программа должна осуществлять занесение в БД новой информации об игроках, удаление информации об игроках из БД, вывод информации по результату поиска сведений о футболисте на экран, окончание работы с программой (в виде меню).

На листинге 1 представлен код программы, удовлетворяющей требованиям.

Листинг 1 ­– База данных футболистов

:- dynamic football\_player/6.

football\_player('john', 'Denver Broncos', 11, 190, 90, 5).

football\_player('jane', 'Denver Broncos', 1, 196, 100, 5).

football\_player('bob', 'Pittsburgh Steelers', 23, 188, 80, 2).

% Меню

menu :-

write('Футбольная база данных:'), nl,

write('1. Добавить нового игрока'), nl,

write('2. Удалить игрока'), nl,

write('3. Вывести информацию об игроке'), nl,

write('4. Выход'), nl,

write('Выберите пункт меню: '), read(Option), nl,

execute(Option).

% Выполнение выбранного пункта меню

execute(1) :-

write('Введите имя игрока: '), read(Name),

write('Введите название команды: '), read(Team),

write('Введите номер игрока: '), read(Number),

write('Введите рост игрока: '), read(Height),

write('Введите вес игрока: '), read(Weight),

write('Введите количество сезонов в NFL: '), read(Seasons),

asserta(football\_player(Name, Team, Number, Height, Weight, Seasons)), nl,

write('Игрок добавлен в базу данных.'), nl, menu.

execute(2) :-

write('Введите имя игрока для удаления: '), read(Name),

retract(football\_player(Name, \_, \_, \_, \_, \_)), nl,

write('Игрок удален из базы данных.'), nl, menu.

execute(3) :-

write('Введите имя игрока для вывода информации: '), read(Name),

( football\_player(Name, Team, Number, Height, Weight, Seasons)

-> write('Имя: '), write(Name), nl,

write('Команда: '), write(Team), nl,

write('Номер: '), write(Number), nl,

write('Рост: '), write(Height), nl,

write('Вес: '), write(Weight), nl,

write('Количество сезонов в NFL: '), write(Seasons), nl

; write('Игрок не найден.'), nl

),

menu.

execute(4) :-

write('Работа с программой завершена.'), nl.

execute(\_) :-

write('Неверный пункт меню, попробуйте еще раз.'), nl, menu.

На рисунке 1 представлены результаты выполнения этой программы.

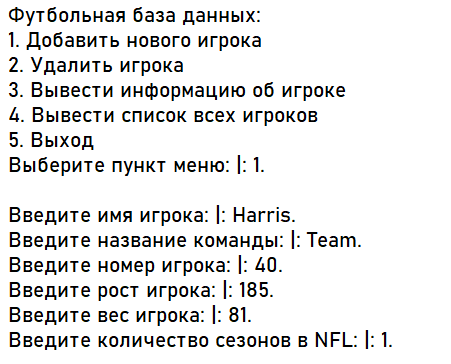


Рисунок 1 – Результат выполнения задания 1

# Задание 2

Добавьте в программу "Футбольная база данных" еще вывод списка всех игроков из БД.

На листинге 2 представлен код программы удовлетворяющией требованиям.

Листинг 2 – Улучшенная БД

:- dynamic football\_player/6.

football\_player('john', 'Denver Broncos', 11, 190, 90, 5).

football\_player('jane', 'Denver Broncos', 1, 196, 100, 5).

football\_player('bob', 'Pittsburgh Steelers', 23, 188, 80, 2).

% Меню

menu :-

write('Футбольная база данных:'), nl,

write('1. Добавить нового игрока'), nl,

write('2. Удалить игрока'), nl,

write('3. Вывести информацию об игроке'), nl,

write('4. Вывести список всех игроков'), nl,

write('5. Выход'), nl,

write('Выберите пункт меню: '), read(Option), nl,

m(Option).

% Выполнение выбранного пункта меню

m(1) :-

write('Введите имя игрока: '), read(Name),

write('Введите название команды: '), read(Team),

write('Введите номер игрока: '), read(Number),

write('Введите рост игрока: '), read(Height),

write('Введите вес игрока: '), read(Weight),

write('Введите количество сезонов в NFL: '), read(Seasons),

asserta(football\_player(Name, Team, Number, Height, Weight, Seasons)), nl,

write('Игрок добавлен в базу данных.'), nl, menu.

m(2) :-

write('Введите имя игрока для удаления: '), read(Name),

retract(football\_player(Name, \_, \_, \_, \_, \_)), nl,

write('Игрок удален из базы данных.'), nl, menu.

m(3) :-

write('Введите имя игрока для вывода информации: '), read(Name),

( football\_player(Name, Team, Number, Height, Weight, Seasons)

-> write('Имя: '), write(Name), nl,

write('Команда: '), write(Team), nl,

write('Номер: '), write(Number), nl,

write('Рост: '), write(Height), nl,

write('Вес: '), write(Weight), nl,

write('Количество сезонов в NFL: '), write(Seasons), nl

; write('Игрок не найден.'), nl

),

menu.

m(4) :-

write('Список всех игроков:'), nl,

list\_all\_players,

menu.

m(5) :-

write('Работа с программой завершена.'), nl.

m(\_) :-

write('Неверный пункт меню, попробуйте еще раз.'), nl, menu.

% Вывод списка всех игроков

list\_all\_players :-

findall(Player, football\_player(Player, \_, \_, \_, \_, \_), Players),

( Players = []

-> write('База данных пуста.'), nl

; write('Игрок'), write(' | '), write('Команда'), write(' | '), write('Номер'), write(' | '), write('Рост'), write(' | '), write('Вес'), write(' | '), write('Сезоны в NFL'), nl,

print\_players(Players)

),

nl.

print\_players([]).

print\_players([Player|Rest]) :-

football\_player(Player, Team, Number, Height, Weight, Seasons),

write(Player), write(' | '), write(Team), write(' | '), write(Number), write(' | '), write(Height), write(' | '), write(Weight), write(' | '), write(Seasons), nl,

print\_players(Rest).

Результат представлен на рисунке 2.

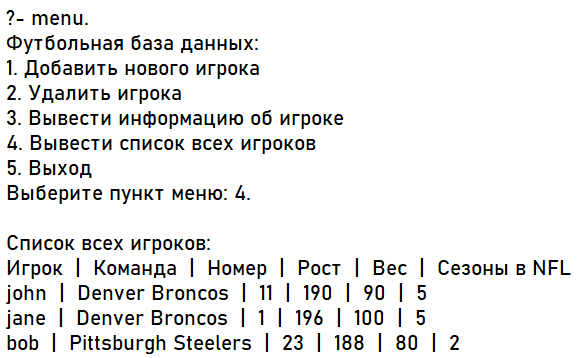


Рисунок 2 – Результат выполнения задания 2

# ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

Создать базу данных о группе студентов: фамилия, имя, группа. Выяснить, имеются ли в группе однофамильцы.

На листинге 3 представлен код программы для выявления однофамильцев.

Листинг 3 ­– Выявление однофамильцев

distinct(List) :-

sort(List, Sorted),

length(List, Length),

length(Sorted, Length).

:- dynamic student/3.

student('Дмитрий', 'Иванов', 401).

student('Ислам', 'Хасанов', 401).

student('Олег', 'Иванов', 401).

same\_surname :-

findall(X, student(\_, X, \_), List),

(

writeln(List),

distinct(List)

-> write('Нет студентов с одинаковой фамилией.'), nl

; write('Есть студенты с одинаковой фамилией.'), nl

).

На рисунке 3 представлен результат выполнения этой программы.

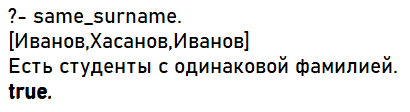


Рисунок 3 – Результат выполнения индивидуального задания