



МИНОБРНАУКИ РОССИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технологический университет «СТАНКИН»
(ФГБОУ ВО МГТУ «СТАНКИН»)

Институт
информационных технологий

Кафедра
информационных систем

Отчет по лабораторной работе №4

по дисциплине **«Интеллектуальные и экспертные системы»**
на тему: «Работа с рекурсией в среде Prolog»

Студент
группа ИДБ–22–06

Мустафаева П.М.

подпись

Руководитель
старший преподаватель

Быстрикова В. А.

подпись

ВВЕДЕНИЕ

Целью работы является изучение возможностей представлений и обработки данных в программах на языке Пролог с использованием рекурсии.

ХОД РАБОТЫ

ГЕНЕАЛОГИЧЕСКОЕ ДЕРЕВО

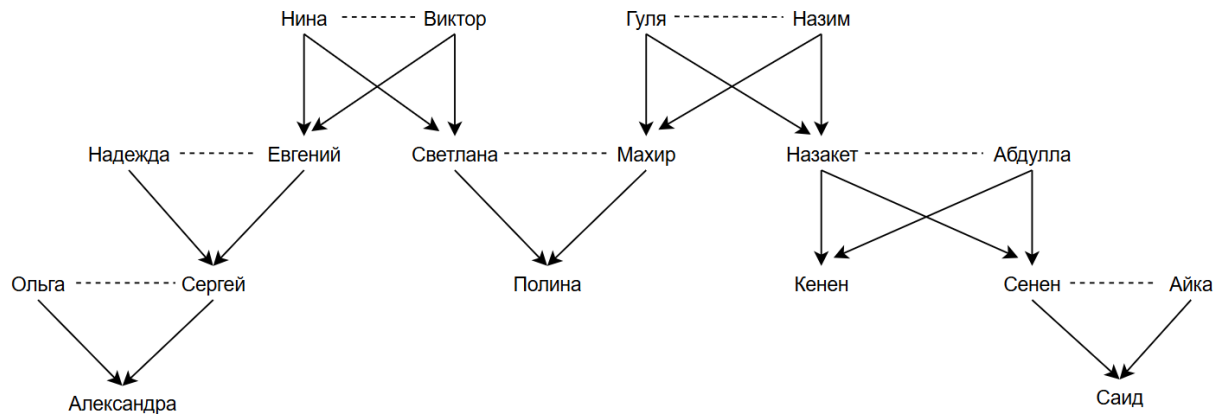


Рис. 1 Генеалогическое дерево до 4 поколения

ЗАДАНИЕ 1

Дополните генеалогическое дерево, построенное в рамках выполнения лабораторной работы 3, до 4-5 поколений (в любом направлении).

Вариант 2.

а) С помощью рекурсии определить является ли один человек потомком другого человека. Предикат должен возвращать значение true при выполнении этого условия.

б) Модифицировать предыдущий предикат (правило) или написать новый для определения количества поколений, разделяющих двух людей, один из которых является потомком другого.

в) С помощью рекурсии определить всех потомков указанного человека любого поколения. Для вывода их имен используйте встроенный предикат `writeln`.

г) Модифицировать предыдущий предикат (правило) или написать новый для определения номера поколения найденного потомка.

Листинг 1 – Программный код

```
мужчина('Виктор').  
мужчина('Назим').
```

мужчина('Евгений').
мужчина('Махир').
мужчина('Абдулла').
мужчина('Сергей').
мужчина('Сенен').
мужчина('Кенен').
мужчина('Саид').

женщина('Нина').
женщина('Туля').
женщина('Надежда').
женщина('Светлана').
женщина('Назакет').
женщина('Полина').
женщина('Ольга').
женщина('Айка').
женщина('Александра').

женаты('Виктор', 'Нина').
женаты('Назим', 'Туля').
женаты('Евгений', 'Надежда').
женаты('Махир', 'Светлана').
женаты('Абдулла', 'Назакет').
женаты('Сергей', 'Ольга').
женаты('Сенен', 'Айка').
родитель('Виктор', 'Светлана').
родитель('Нина', 'Светлана').
родитель('Виктор', 'Евгений').
родитель('Нина', 'Евгений').
родитель('Назим', 'Махир').
родитель('Туля', 'Махир').
родитель('Назим', 'Назакет').
родитель('Туля', 'Назакет').
родитель('Евгений', 'Сергей').
родитель('Надежда', 'Сергей').
родитель('Махир', 'Полина').
родитель('Светлана', 'Полина').
родитель('Абдулла', 'Кенен').
родитель('Назакет', 'Кенен').
родитель('Абдулла', 'Сенен').
родитель('Назакет', 'Сенен').
родитель('Сергей', 'Александра').
родитель('Ольга', 'Александра').
родитель('Сенен', 'Саид').
родитель('Айка', 'Саид').

% Предикат для определения потомка
потомок(Потомок, Родитель) :-
 родитель(Родитель, Потомок).
потомок(Потомок, Родитель) :-
 родитель(Предок, Потомок),

потомок(Предок, Родитель).

% Предикат для определения количества поколений между двумя людьми
поколения_между(Человек1, Человек2, 1) :-

родитель(Человек2, Человек1).

поколения_между(Человек1, Человек2, Количество) :-

родитель(Предок, Человек1),

поколения_между(Предок, Человек2, МеньшееКоличество),

Количество is МеньшееКоличество + 1.

% Предикат для вывода всех потомков

потомки(Предок) :-

потомок(Потомок, Предок),

writeln(Потомок),

fail.

потомки(_).

% Предикат для вывода всех потомков с указанием поколения

потомки_с_поколением(Предок) :-

поколения_между(Потомок, Предок, НомерПоколения),

write(Потомок), write(' '), writeln(НомерПоколения),

fail.

потомки_с_поколением(_).

Результаты ответа на вопросы представлены на рис. 2-5.

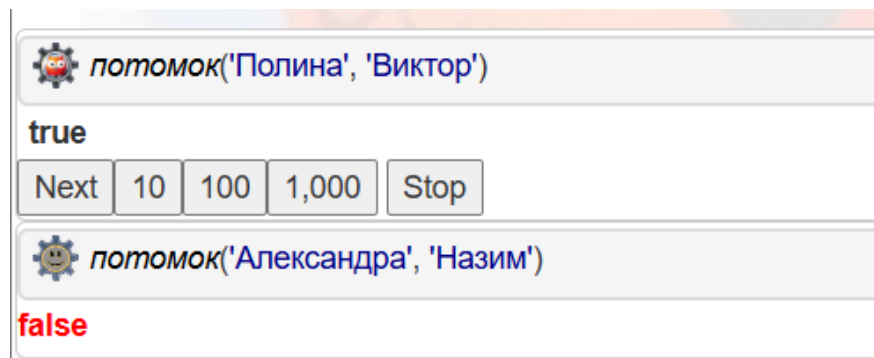


Рис. 2 Запрос «Является ли один человек предком другого?»

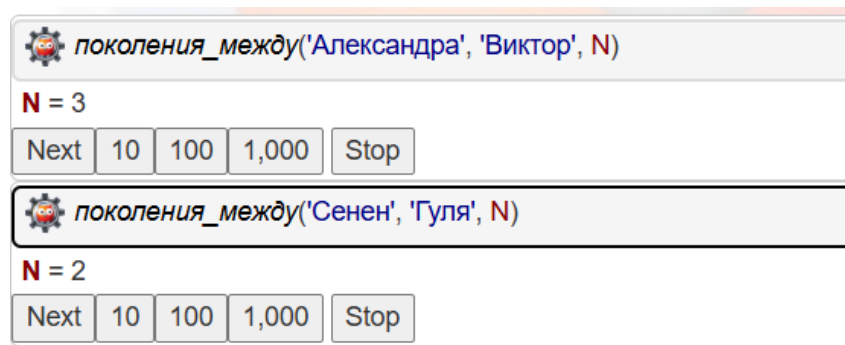


Рис. 3 Запрос «Сколько поколений между двумя людьми?»

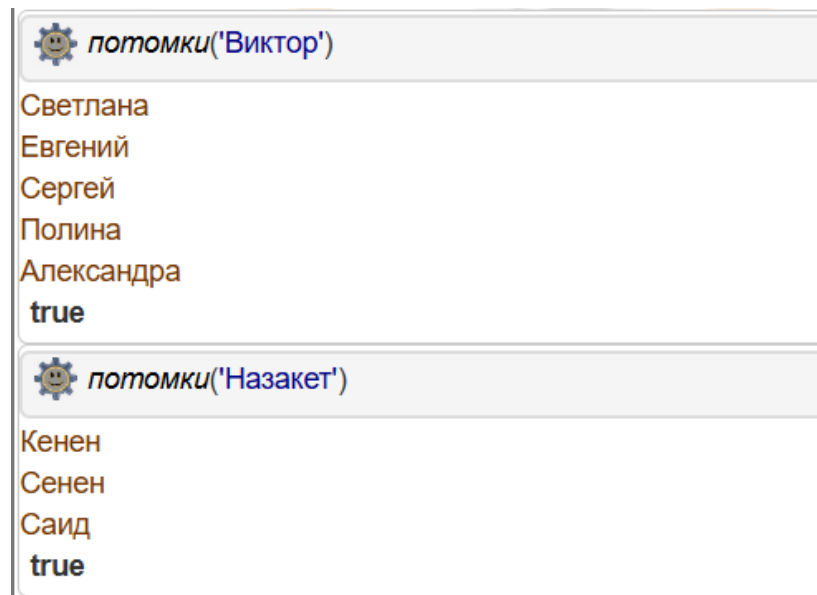


Рис. 4 Запрос «Определить всех потомков указанного человека»

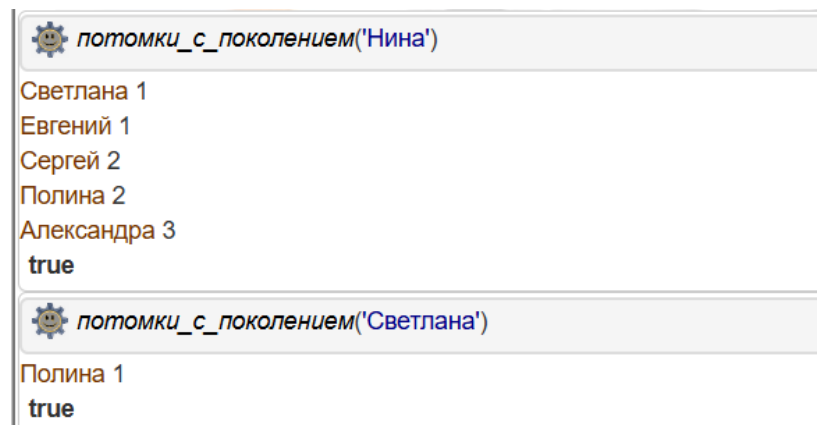


Рис. 5 Запрос «Определить всех потомков указанного человека с номером поколения»

ЗАДАНИЕ 2

Вариант 2.

Нахождение n-го числа последовательности Фибоначчи, равного сумме двух предыдущих членов. Первые два числа последовательности равны 1.

Листинг 2 – Программный код

```
fib(1, 1) :- !.      % Первое число Фибоначчи = 1
fib(2, 1) :- !.      % Второе число Фибоначчи = 1
fib(N, Result) :-    % Рекурсия
    N > 2,            % Проверка, что N > 2
    N1 is N - 1,      % N-1
    N2 is N - 2,      % N-2
    fib(N1, Res1),    % Вычисляем Fib(N-1)
    fib(N2, Res2),    % Вычисляем Fib(N-2)
```

Result is Res1 + Res2. % Fib(N) = Fib(N-1) + Fib(N-2)

Результат работы программы представлен на рис. 6.

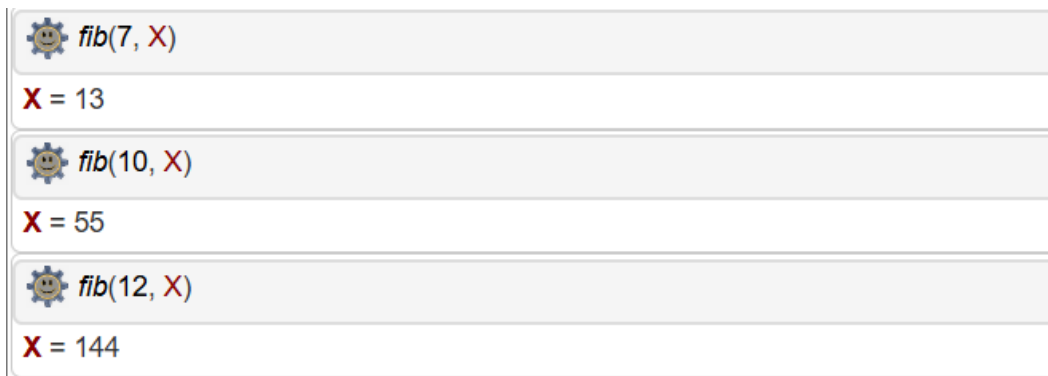


Рис. 6 Результат нахождения n-го числа последовательности Фибоначчи

ЗАДАНИЕ 3

Вариант 2.

Разработать программу расчета функции с использованием рекурсивных правил, отвечающую следующим требованиям.

А) Программа должна запрашивать у пользователя:

- N – количество членов ряда, учитываемых при расчете приближенного значения функции;
- X – значение переменной, меньше 1 (если в формуле есть X).

Б) Выдавать результаты работы на экран:

- приближенное значение функции, рассчитанное с помощью ряда;
- точное значение функции, рассчитанное с помощью встроенных функций.

В) По варианту добавить в программу правила для определения приближенного значения функции с помощью ряда:

$$2. y = \frac{1}{(1+x)^2} = 1 - 2x + 3x^2 - 4x^3 + \dots + (-1)^{n-1}(n+1)x^{n-1}$$

Листинг 3 – Программный код

```
степень(_, 0, 1) :- !.  
степень(X, N, Result) :-  
    N > 0,  
    N1 is N - 1,
```

```

степень(X, N1, R),
Result is X * R.

член_ряда(X, N, ЗначениеЧлена) :-
    K is N - 1,
    Знак is (-1)^K,
    степень(X, K, X_степень),
    Коэффициент is K + 1,
    ЗначениеЧлена is Знак * Коэффициент * X_степень.

сумма_ряда(X, 1, Сумма) :-
    член_ряда(X, 1, Сумма).

сумма_ряда(X, N, Сумма) :-
    N > 1,
    PrevN is N - 1,
    сумма_ряда(X, PrevN, PartialSum),
    член_ряда(X, N, CurrentTerm),
    Сумма is PartialSum + CurrentTerm.

приближенное_значение(X, N, Approx) :- сумма_ряда(X, N, Approx).
точное_значение(X, Exact) :- Exact is 1 / (1 + X)^2.

вычислить_функцию :-
    write('Введите N (количество членов ряда): '), flush_output,
    read(N), integer(N), N > 0,
    write('Введите X (|X| < 1): '), flush_output,
    read(X), number(X), abs(X) < 1,

    приближенное_значение(X, N, Approx),
    точное_значение(X, Exact),

    nl, write('Результаты:'), nl,
    format('Приближенное значение (~d членов): ~10g~n', [N, Approx]),
    format('Точное значение: ~15g~n', [Exact]).

```

Результат работы программы представлен на рис. 7.

 вычислить_функцию

Введите N (количество членов ряда):

5

Введите X ($|X| < 1$):

0.5

Результаты:

Приближенное значение (5 членов): 0.5625

Точное значение: 0.4444444444444444

true


Next

10

100

1,000

Stop

 вычислить_функцию

Введите N (количество членов ряда):

10

Введите X ($|X| < 1$):

0.7

Результаты:

Приближенное значение (10 членов): 0.17008462

Точное значение: 0.346020761245675

true

Next

10

100

1,000

Stop

Рис. 7 Результат расчета приближенного значения функции с помощью ряда

9

ВЫВОД

В ходе лабораторной работы были изучены возможности представления и обработки данных в языке Пролог с использованием рекурсии. На примере генеалогического дерева разработаны рекурсивные предикаты для проверки родственных связей, подсчёта поколений между предком и потомком, а также вывода всех потомков с указанием их поколения. Рекурсия продемонстрировала свою эффективность при работе с иерархическими структурами.

Также реализовано вычисление чисел Фибоначчи и расчёт значения функции через ряд, где рекурсивные правила позволили компактно выразить математические зависимости.