МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«УФИМСКИЙ ГОСУДАРСТВЕННЫЙ АВИАЦИОННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра вычислительной математики и кибернетики

Отчет по лабораторной работе №5

«Функциональное и логическое программирование»

по теме

«Основные механизмы работы Пролога, списки»

Выполнил: студент гр. МО-321

Шемануев А.Е.

Проверил: Профессор, д/н, доцент

Ризванов Д.А.

Уфа – 2022

# **Цель работы**

1. Знакомство со средой Visual Prolog.

* Настройка системы, интерфейс
* Структура программы: domains, database, predicates, goal, clauses

1. Изучение основных механизмов работы Пролога: объекты и отношения между ними.

* Управление перебором. Отсечение (cut, !) и откат (fail). Сопоставление аргументов. Определение отношения “различны”.
* Арифметические операции:

сравнения чисел, факториал, максимальное/минимальное/среднее двух и более чисел.

* Родственные отношения:
* Ввести в программу факты следующего вида:

человек, родитель;

* На базе приведенных выше фактов определить отношения:
* мужчина, женщина;
* отец, мать, сын, дочь;
* брат, сестра, братья, сестры;
* дедушка, бабушка;
* дядя, тетя, племянник, племянница;
* предок, отпрыск, люди одного поколения.
* Определить отношения: ребенок, взрослый.

1. Сформулировать все возможные вопросы, касающиеся отношений, определенных в программе.
2. Выполнить индивидуальное задание к лабораторной работе №1 в соответствии со своим вариантом.

# **Ход работы**

# **Родственные отношения:**

domains

name=symbol

database

parent(name,name)

man(name)

woman(name)

predicates

nondeterm different(name,name).

nondeterm mother(name,name).

nondeterm dad(name,name).

nondeterm son(name,name).

nondeterm daughter(name,name).

nondeterm brother(name,name).

nondeterm sister(name,name).

nondeterm sisters(name,name).

nondeterm granny(name,name).

nondeterm granddad(name,name).

nondeterm gender(name,name).

nondeterm ancestor(name,name).

nondeterm descendant(name,name).

nondeterm generation(name,name).

nondeterm adult(name).

nondeterm child(name).

nondeterm genUp(name,name).

nondeterm genDown(name,name).

clauses

parent("сергей","кирилл").

parent("сергей","никита").

parent("сергей","вика").

parent("вика","денис").

parent("денис","саша").

parent("денис","макс").

parent("маша","саша").

man("сергей").

man("кирилл").

man("никита").

man("денис").

man("саша").

man("макс").

woman("вика").

different(X,Y):-X<>Y.

ancestor(X,Y):-parent(X,Y).

ancestor(X,Y):-parent(X,P),ancestor(P,Y).

descendant(X,Y):-ancestor(Y,X).

gender(X,Y):-Y=муж,assertz(man(X)).

gender(X,Y):-Y=жен,assertz(woman(X)).

mother(X,Y):-parent(X,Y),woman(X).

dad(X,Y):-parent(X,Y),man(X).

son(X,Y):-parent(Y,X),man(X).

daughter(X,Y):-parent(Y,X),woman(X).

brother(X,Y):-parent(Z,X),parent(Z,Y),man(X),different(X,Y).

sister(X,Y):-parent(Z,X),parent(Z,Y),woman(X),different(X,Y).

sisters(X,Y):-parent(Z,X),parent(Z,Y),woman(X),woman(Y),different(X,Y).

granny(X,Y):-parent(X,P),parent(P,Y),woman(X).

granddad(X,Y):-parent(X,P),parent(P,Y),man(X).

generation(X,Y) :- genUp(X,Y);genDown(X,Y).

genUp(X,Y) :- X=Y, !.

genUp(X,Y):- parent(PX,X),parent(PY,Y), genUp(PX,PY).

genDown(X,Y) :- X=Y, !.

genDown(X,Y):- parent(X,PX),parent(Y,PY), genDown(PX,PY).

adult(X):- parent(X,\_).

child(X):- not(parent(X,\_)).

goal

%brother("кирилл", X).

generation("маша", "сергей").

%adult("вика").

**Общие задания:**

**ЗАДАЧА 1.**

Дана программа.

domains

x = symbol

y, z = integer

predicates

nondeterm p(x, y, z)

nondeterm p1(x, y)

nondeterm p2(x, z)

nondeterm p3(x, y)

clauses

p1(a, 1).

p2(a, 2).

p3(a, 3).

p(X, Y, Z) :- p1(X, Y), p2(X, Z).

p(X, Y, Z) :- p2(X, Z), p3(X, Y).

% 1.1. Исполните согласование внешней цели p(a,Y,Z).

% 1.2. В комментариях запишите и прокомментируйте результат

% согласования цели.

Изображение выглядит как текст

Автоматически созданное описание

%Поиск решения начинается, и переходит к 1 определению p(X,Y,Z), где находит факт p1(X,Y), затем p2(X,Z).

Так как в конце 1 определения нет !, то осуществляется переход ко 2 определению p(X,Y,Z), где аналогичным образом находит факты p2(X,Z) и p3(X,Y)

1.3. Укажите системе вывода, чтобы выполнила согласование только

% первого определения предиката p(X,Y,Z).

% 1.4. Исполните согласование внешней цели p(a,Y,Z).

% 1.5. В комментариях запишите и прокомментируйте результат

% согласования цели.

Изображение выглядит как текст

Автоматически созданное описание

% Здесь будет согласован только 1 определение предиката p(X,Y,Z), т.к. в конце 1 определения есть cut(!) которая предотвращает переход ко второму определению p(X,Y,Z).

% 1.6. Удалите из текста программы все внесенные Вами изменения.

% 1.7. Добавьте факт p1(a,11). в базу данных.

% 1.8. Исполните согласование внешней цели p(a,Y,Z).

% 1.9. В комментариях запишите и прокомментируйте результат

% согласования цели.

Изображение выглядит как текст

Автоматически созданное описание

%Здесь 1 определение будет согласовано 2 раза, т.к. мы добавили еще один факт p1(a,11).

% 1.10. Внесите в программу изменения, соответствующие выполнению

% требования 1.3.

% 1.11. Исполните согласование внешней цели p(a,Y,Z).

% 1.12. В комментариях запишите и прокомментируйте результат

% согласования цели.

% 1.13. Поместите в комментарий текст программы и перейдите к

% выполнению требований к работе с программой задачи 2. Изображение выглядит как текст

Автоматически созданное описание

%Здесь 1 определение будет согласовано 2 раза. Так как в начале стоит !(cut), то есть программа предотвратит переход ко 2-у определению.

**ЗАДАЧА 2.**

Дана программа.

domains

x = integer

predicates

nondeterm r(x)

nondeterm r1(x)

nondeterm r2(x)

clauses

r1(1).

r2(2).

r(X) :- r1(X), write("согласовано первое определение").

r(X) :- r2(X), write("согласовано второе определение").

% 1.1. Исполните согласование внешней цели r(X).

% 1.2. В комментариях запишите и прокомментируйте результат

% согласования цели.

Изображение выглядит как текст

Автоматически созданное описание

%Здесь происходит согласование двух определений предиката r

% 1.3. Исполните согласование внешней цели r(1).

% 1.4. В комментариях запишите и прокомментируйте результат

% согласования цели.

Изображение выглядит как текст

Автоматически созданное описание

%Согласуется только первое определение предиката r, т.к. в нем согласуется факт r1.

% 1.5. Исполните согласование внешней цели r(2).

% 1.6. В комментариях запишите и прокомментируйте результат

% согласования цели.

Изображение выглядит как текст

Автоматически созданное описание

%Согласуется только второе определение предиката r, т.к. в нем согласуется факт r2.

% 1.7. Укажите системе вывода, чтобы выполнила согласование только

% первого определения предиката r(X) или никакого.

% 1.8. Исполните согласование внешней цели r(X).

% 1.9. В комментариях запишите и прокомментируйте результат

% согласования цели.

Изображение выглядит как текст

Автоматически созданное описание

%Из-за отсечения ! согласуются только все определения первого предиката r, в данном случае программа выдаст только одно решение.

% 1.10. Исполните согласование внешней цели r(1).

% 1.11. В комментариях запишите и прокомментируйте результат

% согласования цели.

Изображение выглядит как текст

Автоматически созданное описание

%Согласуется только первое определение предиката

% 1.12. Исполните согласование внешней цели r(2).

% 1.13. В комментариях запишите и прокомментируйте результат

% согласования цели.

% 1.14. Поместите в комментарий текст программы и перейдите к

% выполнению требований к работе с программой задачи 3.

Изображение выглядит как текст

Автоматически созданное описание

%Второе определение предиката не согласуется из-за отсечения первого, поэтому решений нет

**ЗАДАЧА 3**

Дана программа.

domains

l = integer\*

predicates

nondeterm append2(l, l, l)

clauses

append2([], L, L) :- !.

append2([H|L1], L2, [H|L3]) :- append2(L1, L2, L3).

% 1.1. Какие цели можно сформулировать к базе знаний,

% представленной в данной программе?

% 1.2. Исполните согласование целей.

% 1.3. В комментариях запишите и прокомментируйте результат

% согласования целей.

% 1.4. Какую задачу выполняет отсечение?

% 1.5. Поместите в комментарий текст программы и перейдите к

% выполнению требований к работе с программой задачи 4.

Изображение выглядит как текст

Автоматически созданное описание

%Отсечение делает так, что программа не выдаст больше одного решения. Могут быть случаи, когда ответов несколько, например append2(X,Y,[1,2,3]) без отсечения вернет 4 решения вместо одного.  
%Можно сформулировать следующие цели: Какой будет результат конкатенации двух списков? Известен первый(второй) список и результат конкатенации, какой должен быть второй(первый) список, чтобы получилось верная конкатенация? И др.

**ЗАДАЧА 4.**

Дана программа.

domains

x = integer

predicates

p(x)

p1(x)

p2(x)

p4(x)

clauses

p1(1).

p1(11).

p2(1).

p2(11).

      p4(X) :- X = 10 + 10.

p(X) :- p1(X), write("p1 ").

p(Y) :- p2(Y), write("p2 ").

p(Z) :- p4(Z), write("p4 ").

% Удалите символы /\* и \*/, ограничивающие текст программы.

% Укажите системе, чтобы программа возвратила

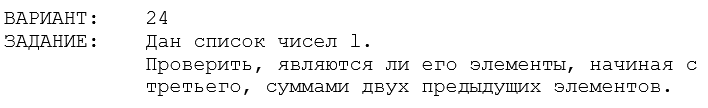
% только первые решения для каждого определения

% пpедиката p(X).

Изображение выглядит как стол

Автоматически созданное описание

**Индивидуальное задание**



# 

# **Листинг программы:**

**Задача 1**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**1.1-1.2**

domains

x = symbol

y, z = integer

predicates

nondeterm p(x, y, z)

nondeterm p1(x, y)

nondeterm p2(x, z)

nondeterm p3(x, y)

clauses

p1(a, 1).

p2(a, 2).

p3(a, 3).

p(X, Y, Z) :- p1(X, Y), p2(X, Z).

p(X, Y, Z) :- p2(X, Z), p3(X, Y).

goal

p(a,Y,Z).

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**1.3-1.5**

domains

x = symbol

y, z = integer

predicates

nondeterm p(x, y, z)

nondeterm p1(x, y)

nondeterm p2(x, z)

nondeterm p3(x, y)

clauses

p1(a, 1).

p2(a, 2).

p3(a, 3).

p(X, Y, Z) :- p1(X, Y), p2(X, Z),!.

p(X, Y, Z) :- p2(X, Z), p3(X, Y).

goal

p(a,Y,Z).

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**1.6-1.9**

domains

x = symbol

y, z = integer

predicates

nondeterm p(x, y, z)

nondeterm p1(x, y)

nondeterm p2(x, z)

nondeterm p3(x, y)

clauses

p1(a, 1).

p1(a,11).

p2(a, 2).

p3(a, 3).

p(X, Y, Z) :- p1(X, Y), p2(X, Z).

p(X, Y, Z) :- p2(X, Z), p3(X, Y).

goal

p(a,Y,Z).

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**1.10-1.13**

domains

x = symbol

y, z = integer

predicates

nondeterm p(x, y, z)

nondeterm p1(x, y)

nondeterm p2(x, z)

nondeterm p3(x, y)

clauses

p1(a, 1).

p1(a,11).

p2(a, 2).

p3(a, 3).

p(X, Y, Z) :- !,p1(X, Y), p2(X, Z).

p(X, Y, Z) :- p2(X, Z), p3(X, Y).

goal

p(a,Y,Z).

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Задача 2**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

1.1–1.2

domains

x = integer

predicates

nondeterm r(x)

nondeterm r1(x)

nondeterm r2(x)

clauses

r1(1).

r2(2).

r(X) :- r1(X), write("согласовано первое определение").

r(X) :- r2(X), write("согласовано второе определение").

goal

r(X).

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1.3–1.4

domains

x = integer

predicates

nondeterm r(x)

nondeterm r1(x)

nondeterm r2(x)

clauses

r1(1).

r2(2).

r(X) :- r1(X), write("согласовано первое определение ").

r(X) :- r2(X), write("согласовано второе определение ").

goal

r(1).

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1.5–1.6

domains

x = integer

predicates

nondeterm r(x)

nondeterm r1(x)

nondeterm r2(x)

clauses

r1(1).

r2(2).

r(X) :- r1(X), write("согласовано первое определение ").

r(X) :- r2(X), write("согласовано второе определение ").

goal

r(2).

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1.7–1.9

domains

x = integer

predicates

nondeterm r(x)

nondeterm r1(x)

nondeterm r2(x)

clauses

r1(1).

r2(2).

r(X) :- !,r1(X), write("согласовано первое определение ").

r(X) :- r2(X), write("согласовано второе определение ").

goal

r(X).

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1.10–1.11

domains

x = integer

predicates

nondeterm r(x)

nondeterm r1(x)

nondeterm r2(x)

clauses

r1(1).

r2(2).

r(X) :- !,r1(X), write("согласовано первое определение ").

r(X) :- r2(X), write("согласовано второе определение ").

goal

r(1).

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1.12-1.14

domains

x = integer

predicates

nondeterm r(x)

nondeterm r1(x)

nondeterm r2(x)

clauses

r1(1).

r2(2).

r(X) :- !,r1(X), write("согласовано первое определение ").

r(X) :- r2(X), write("согласовано второе определение ").

goal

r(2).

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Задача 3.**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

domains

l = integer\*

predicates

nondeterm append2(l, l, l)

clauses

append2([], L, L) :- !.

append2([H|L1], L2, [H|L3]) :- append2(L1, L2, L3).

goal

append2([1,2,3,4],[74,23,12],X).

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Задача 4.**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

domains

x = integer

predicates

nondeterm p(x)

nondeterm p1(x)

nondeterm p2(x)

nondeterm p4(x)

nondeterm p\_1(x)

nondeterm p\_2(x)

nondeterm p\_4(x)

clauses

p1(1).

p1(11).

p2(1).

p2(11).

p4(X) :- X = 10 + 10.

p(X) :- p\_1(X); p\_2(X) ; p\_4(X).

p\_1(X) :- p1(X), write(" p1 "), !.

p\_2(Y) :- p2(Y), write(" p2 "), !.

p\_4(Z) :- p4(Z), write(" p4 "), !.

goal

p(X).

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Индивидуальное задание 19**

domains

i = integer

lii = i\*

li = lii\*

l = li\*

predicates

nondeterm app(l, lii)

nondeterm sum(li, integer)

nondeterm list\_sum(lii, integer)

nondeterm list\_list\_sum(li, integer)

clauses

list\_sum([], 0).

list\_sum([H|T], Sum) :- list\_sum(T,Sum1), Sum = H + Sum1.

list\_list\_sum([], 0).

list\_list\_sum([H|T], Sum) :- list\_sum(H, Sum1), list\_list\_sum(T, Sum2), Sum = Sum1+Sum2.

sum(L, H1) :- list\_list\_sum(L, H1).

app([H|T],[H1|T1]) :- sum(H, H1), app(T, T1), !.

app(L,[]).

goal

app([[[1,2,3],[4,5,6]],[[7,8,9]], [[], [1,2]]], X).

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Родственные отношения**

database

par(string, string)

wom(string)

man(string)

hum(string)

domains

str = string

i = integer

predicates

%nondeterm par(str, str).

nondeterm print1.

nondeterm print2.

nondeterm print3.

nondeterm print4.

nondeterm father(str, str).

nondeterm mother(str, str).

nondeterm son(str, str).

nondeterm daughter(str, str).

nondeterm brother(str, str).

nondeterm sister(str, str).

nondeterm grandfather(str, str).

nondeterm grandmother(str, str).

nondeterm uncle(str, str).

nondeterm aunt(str, str).

nondeterm nephew(str, str). %Племяник

nondeterm niece(str, str). %Племяница

nondeterm ancestor(str, str). %предок

nondeterm offspring(str, str) %Отпрыск

nondeterm sameGeneration(str, str). %предок

nondeterm sameGenerationU(str, str). %предок

nondeterm sameGenerationD(str, str). %предок

nondeterm child(str).

nondeterm adult(str).

nondeterm menu().

nondeterm input().

nondeterm action(str).

nondeterm addI().

nondeterm removeСonnectionI().

nondeterm removePersonI().

nondeterm renameI().

nondeterm changeGenderI().

nondeterm rename(str, str).

nondeterm renameGender(str, str).

nondeterm renamePar(str, str).

nondeterm renameChil(str, str).

nondeterm removePerson(str).

nondeterm changeGender(str, str).

clauses

%ft("Пам", "Боб").

print1:- write("4"), nl, par(A,B), write(A," - ",B),nl, fail.

print1.

print2:- write("4"), nl, wom(A), write("wom - ", A),nl, fail.

print2.

print3:- write("4"), nl, man(A), write("man - ", A),nl, fail.

print3.

print4:- write("4"), nl, hum(A),write("hum - ", A),nl, fail.

print4.

father(A,B) :- par(A,B), man(A).

mother(A,B) :- par(A,B), wom(A).

son(A,B) :- par(B,A), man(A).

daughter(A,B) :- par(B,A), wom(A).

brother(A, B) :- man(A), par(C, A), par(C, B), not(A=B).

sister(A, B) :- wom(A), par(C, A), par(C, B), not(A=B).

grandfather(A, B) :- man(A), par(A,C), par(C, B).

grandmother(A, B) :- wom(A), par(A,C), par(C, B).

uncle(A, B) :- man(A), brother(A,C), par(C, B).

uncle(A, B) :- man(A), sister(A,C), par(C, B).

aunt(A, B) :- wom(A), brother(A,C), par(C, B).

aunt(A, B) :- wom(A), sister(A,C), par(C, B).

nephew(A, B) :- man(A), par(C, A), brother(C,B).

nephew(A, B) :- man(A), par(C, A), sister(C,B).

niece(A, B) :- wom(A), par(C, A), brother(C,B).

niece(A, B) :- wom(A), par(C, A), sister(C,B).

ancestor(A,B) :- par(A,B).

ancestor(A,B) :- par(C, B), ancestor(A,C).

offspring(A,B) :- ancestor(B,A).

sameGeneration(A,B) :- sameGenerationU(A,B); sameGenerationD(A,B) .

sameGenerationU(A,B) :- A=B.%; sister(A,B) ; brother(A,B).

sameGenerationU(A,B) :- par(C,A), par(D,B), sameGenerationU(C,D).

sameGenerationD(A,B) :- A=B.%; sister(A,B) ; brother(A,B).

sameGenerationD(A,B) :- par(A,C), par(B,D), sameGenerationD(C,D).

child(A) :- par(A,B), !,fail.

child(A).

adult(A) :- not(child(A)).

%Вывод

%%Вывод

%Добавить

%Удалить

%Переименовать

%Сохранить

%Save("C:/Path")

%Consult("C:/Path")

rename(Old, New):- hum(Old), assert(hum(New)), renameGender(Old, New), renamePar(Old, New), renameChil(Old, New), retract(hum(Old)).

renameGender(Old, New):- man(Old), assert(man(New)), retract(man(Old)).

renameGender(Old, New):- wom(Old), assert(wom(New)), retract(wom(Old)).

renamePar(Old, New):- par(Old, A), assert(par(New, A)), retract(par(Old, A)), renamePar(Old, New).

renamePar(\_, \_).

renameChil(Old, New):- par(A, Old), assert(par(A, New)), retract(par(A, Old)), renameChil(Old, New).

renameChil(\_, \_).

removePerson(Per):- retractall(par(Per, \_)), fail.

removePerson(Per):- retractall(par(\_, Per)), fail.

removePerson(Per):- retract(man(Per)), fail.

removePerson(Per):- retract(wom(Per)), fail.

removePerson(Per):- retract(hum(Per)).

changeGender(Per, G):- not(G="Ж"); not(G="М"), fail, !.

changeGender(Per, G):- retract(man(Per)), fail.

changeGender(Per, G):- retract(wom(Per)), fail.

changeGender(Per, G):- G = "М", assert(man(Per)).

changeGender(Per, G):- G = "Ж", assert(wom(Per)).

addI():- write("ВВедите родителя: "), readln(Par), nl,

write("ВВедите ребёнка : "), readln(Chi), nl,

assert(par(Par, Chi)).

addI():- write("Ошибка").

removeСonnectionI():- write("Введите родителя: "), readln(Par), nl,

write("Введите ребёнка : "), readln(Chi), nl,

retract(par(Par, Chi)).

removeСonnectionI():- write("Ошибка").

removePersonI():-write("кого убрать: "), readln(Per), nl, removePerson(Per).

removePersonI():-write("Ошибка").

renameI():- write("Введите кого переименовать: "), readln(Old), nl,

write("Введите какое имя дать? : "), readln(New), nl,

rename(Old, New).

renameI():- write("Ошибка").

changeGenderI():-write("Введите кого поменять: "), readln(Per), nl,

write("Пол[М\Ж]: "), readln(G), changeGender(Per, G).

menu():-nl,

write("[0] - Вывод"),nl,

write("[1] - Добавить"),nl,

write("[2] - Удалить связь"),nl,

write("[3] - Удалить персону"),nl,

write("[4] - Переименовать"),nl,

write("[5] - Поменять\добавить пол"),nl,

write("[6] - Сохранить "),nl,

input().

input():-readln(I), action(I).

action("0"):-print1, print2, print3, print4, nl, menu().

action("1"):-addI(), menu().

action("2"):-removeСonnectionI(), menu().

action("3"):-removePersonI(), menu().

action("4"):-renameI(), menu().

action("5"):-changeGenderI(), menu().

action("6"):-save("C:/Users/Orkii/Desktop/Prolog/Labs/my.txt"), menu().

goal

consult("C:/Users/Orkii/Desktop/Prolog/Labs/my.txt"),

print1, print2, print3, print4, write(" "),

menu().