|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | МИНОБРНАУКИ РОССИИ | | | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |   Институт информационных технологий  Кафедра математического обеспечения и стандартизации информационных технологий (МОСИТ) |

|  |  |
| --- | --- |
| **ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №4** | |
| **по дисциплине** | |
| **«Функциональное и логическое программирование»**  **по теме**  **«Основные элементы языка Prolog»** | |
|  | |
| Выполнил студент группы ИКБО-07-17 | Руденко В.Р. |
| Принял доцент | Смольянинова В.А. |
|  |  |

|  |  |  |
| --- | --- | --- |
| Лабораторная работа выполнена | «\_\_»\_\_\_\_\_\_\_2019 г. | *(подпись студента)* |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_2019 г. | *(подпись руководителя)* |

Москва 2019

Оглавление

[Задание 3](#_Toc31145528)

[Ход работы 4](#_Toc31145529)

[Заключение 8](#_Toc31145530)

# **Задание**

1. Составьте программу Родственные отношения, которая кроме родственных отношений parent (родитель) и ancestor (предок) программа должна содержать одно или несколько из следующих отношений:

brother (брат); sister (сестра); grand-father (дедушка); grand-mother (бабушка); uncle (дядя).

1. Составьте программу, используя отношения likes ("нравится") и can\_buy ("может купить").
2. Составьте собственную программу, состоящую из фактов и правил. Проверьте ее работу.

# **Ход работы**

1. Составить программу «Родственные отношения».

Для определения отношения родства в программе реализованы предикаты родитель, ребенок, предок, бабушка и дедушка.

Программа 1:

domains

s=symbol

predicates

female(s)

male(s)

mother(s,s)

father(s,s)

child(s,s)

nondeterm parent(s,s)

nondeterm ancestor(s,s)

nondeterm grand\_parent(s,s)

nondeterm grand\_father(s,s)

nondeterm grand\_mother(s,s)

clauses

parent(pam,bob).

parent(tom,bob).

parent(tom,liz).

parent(bob,ann).

parent(bob,pat).

parent(pat,jim).

female(pam).

female(liz).

female(ann).

female(pat).

male(tom).

male(bob).

male(jim).

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

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

father(X,Y):- parent(X,Y),male(X).

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

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

grand\_parent(X, Z) :- parent(X, Y), parent(Y, Z).

grand\_father(X, Z) :- grand\_parent(X, Z), male(X).

grand\_mother(X, Z) :- grand\_parent(X, Z), female(X).

Проверка работы (цели):

goal

grand\_mother(pam, ann),

grand\_father(tom, ann). % yes

% grand\_mother(ann, bob) % no

1. Составьте программу с отношениями «нравится» и «может купить».

В программе определено правило «может купить», условиями которого явяются:

* X является человеком
* Y является продуктом или автомобилем
* Y должен нравится X
* Y должен продаваться

И добавлены соответствующие свойства: человек, продукт, автомобиль, продается и отношение «нравится».

Программа 2:

predicates

nondeterm can\_buy(symbol, symbol)

person(symbol)

food(symbol)

car(symbol)

likes(symbol, symbol)

for\_sale(symbol)

clauses

can\_buy(X, Y) :-

person(X),

food(Y),

likes(X, Y),

for\_sale(Y).

can\_buy(X, Y) :-

person(X),

car(Y),

likes(X, Y),

for\_sale(Y).

person(kelly).

person(judy).

food(lemon).

food(pizza).

car(hot\_rod).

likes(kelly, hot\_rod).

likes(judy, pizza).

for\_sale(pizza).

for\_sale(lemon).

for\_sale(hot\_rod).

Проверка работы (цели):

goal

% can\_buy(judy, hot\_rod) % no

% can\_buy(kelly, hot\_rod). % yes

can\_buy(judy, pizza). % yes

1. Составить собственную программу.

Программа реализует систему взаимоотношений питомца с его хозяином, а также принадлежность питомца к определенной породе.

Программа 3:

domains

animal\_name, color, name = symbol

predicates

nondeterm cat(animal\_name).

nondeterm dog(animal\_name).

nondeterm color(animal\_name, color).

nondeterm have(name, animal\_name).

nondeterm rodoslovnaya(animal\_name).

nondeterm animal(animal\_name).

clauses

cat(butsi).

cat(korni).

cat(mac).

dog(rover).

dog(fles).

dog(spot).

color(butsi, brown).

color(korni, black).

color(mac, yellow).

color(rover, yellow).

color(spot, white).

color(fles, black\_and\_white).

/\*----------------------------\*/

animal(X):-cat(X); dog(X).

/\*----------------------------\*/

rodoslovnaya(X):-animal(X),have(tom, X).

rodoslovnaya(X):-animal(X), have(keit, X).

/\*----------------------------\*/

have(tom, X):-color(X, black); color(X, brown).

/\*----------------------------\*/

have(keit, X):-dog(X),

not(color(X, white)),

not(have(tom, X)).

/\*----------------------------\*/

have(alan, mac):-not(have(keit, butsi)),

not(rodoslovnaya(spot)).

goal

%animal(X).

%have(tom, X).

%have(keit, X).

/\*animals, which have no owner\*/

animal(X), not( have(\_,X)),write(X).

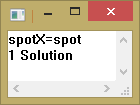


Рисунок 1. Питомцы, у которых нет хозяина

# **Заключение**

В ходе лабораторной работы были рассмотрены основные элементы языка логического программирования Prolog. Были составлены программы на этом языке с использованием таких элементов, как: домены, факты и правила; реализованы запросы к программе в разделе целей. Также были рассмотрены и составлены списковые запросы.