|  |  |
| --- | --- |
| |  | | --- | | МИНОБРНАУКИ РОССИИ  Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | |

Институт Информационных технологий

Кафедра МОСИТ

**ЛАБОРАТОРНАЯ РАБОТА №1**

по дисциплине «Функциональное и Логическое Программирование»

**Студент группы** ИКБО-12-17 Карева М.А. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись студента)*

**Руководитель работы** Смольянинова В.А. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись руководителя)*

Москва, 2019

# Постановка задачи

Задание к примерам программ на языке Пролог.

* Наберите программу. Запустите на выполнение. Придумайте и задайте 5-6 вопросов к каждой программе.

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

# Ход выполнения работы

Первым примером была программа, описывающая схему родства, показанную на рисунке 1.

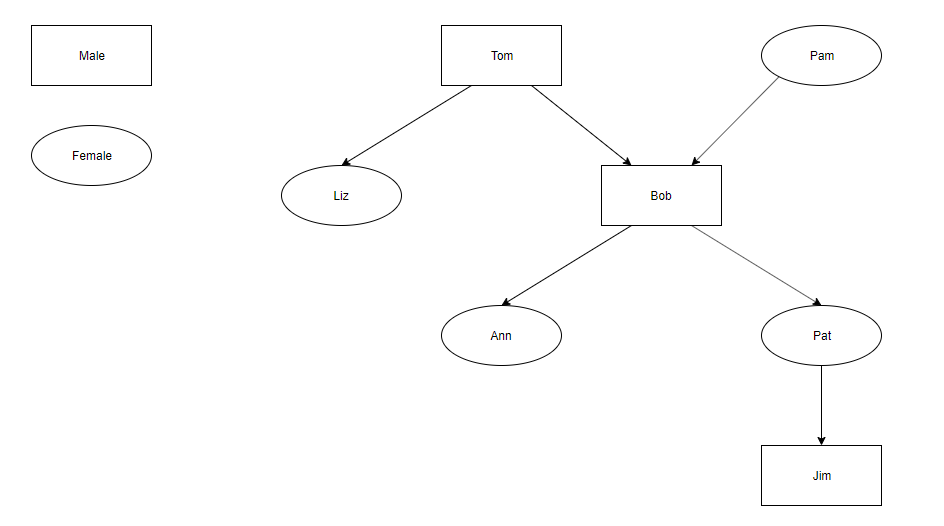


Рисунок 1. Генеалогическое древо

Код программы:

domains

s=symbol

predicates

nondeterm parent(s,s)

female(s)

male(s)

mother(s,s)

father(s,s)

nondeterm ancestor(s,s)

nondeterm child(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).

Для данной программы были заданы следующие вопросы:

* Перечислить детей ann

goal

child(ann,Who).

* Определить, женщина ли bob

goal

female(bob).

* Перечислить детей pam

goal

child(Who,pam).

* Является ли Pam предком tom-а?

goal

ancestor(pam, tom).

* Перечислить всех потомков jim

goal

ancestor(jim,Who).

* Перечислить всех предков liz

goal

ancestor(Who,liz).

Следующий пример рассматривает отношение «нравится»:

predicates

nondeterm likes(symbol,symbol)

clauses

likes(ellen, tennis).

likes(john, football).

likes(tom, baseball).

likes(eric, swimming).

likes(mark, tennis).

Были заданы следующие вопросы:

* Перечислить тех, кому нравится плавание

goal

likes(Who,swimming).

* Перечислить тех, кому нравится футбол

goal

likes(Who,football).

* Что нравится mark?

goal

likes(mark,What).

* Нравится ли mark футбол?

goal

likes(mark,football).

* Нравится ли tom-у теннис?

goal

likes(tom,tennis).

Третий пример рассматривает отношение «может купить»:

predicates

nondeterm can\_buy(symbol, symbol)

person(symbol)

nondeterm car(symbol)

likes(symbol, symbol)

for\_sale(symbol)

clauses

can\_buy(X, Y) :- person(X), car(Y),likes(X, Y), for\_sale(Y).

person(kelly).

person(judy).

car(lemon).

car(hot\_rod).

likes(kelly, hot\_rod).

likes(judy, pizza).

for\_sale(pizza).

for\_sale(lemon).

for\_sale(hot\_rod).

Были заданы следующие вопросы:

* Перечислить, что judy может купить

goal

can\_buy(judi, What).

* Является ли hot\_rod машиной?

goal

car(hot\_rod).

* Нравится ли judy lemon?

goal

likes(judy, lemon).

* Продается ли pizza?

goal

for\_sale(pizza).

* Может ли kelly купить lemon?

goal

can\_buy(kelly, lemon).

Четвертый пример рассматривает отношения «машина» и «грузовик»:

predicates

nondeterm car(symbol,real,integer,symbol,integer)

nondeterm truck(symbol,real,integer,symbol,integer)

clauses

car(chrysler,13000,3,red,12000).

car(ford,90000,4,gray,25000).

car(datsun,8000,1,red,30000).

truck(ford,80000,6,blue,8000).

truck(datsun,50000,5,orange,20000).

truck(toyota,25000,5,black,25000).

goal

/\* truck(What1,What2,What3,gray,What4). \*/

/\* car(What1,What2,What3,orange,What4). \*/

/\* car(datsun,What1,What2,What3,What4). \*/

/\* truck(toyota,What1,What2,What3,What4). \*/

truck(ford,80000,6,blue,8000).

Были заданы следующие вопросы:

* Перечислить серые грузовики

goal

truck(What1,What2,What3,gray,What4).

* Перечислить оранжевые машины

goal

car(What1,What2,What3,orange,What4).

* Перечислить машины марки datsun

goal

car(datsyn,What1,What2,What3,What4).

* Перечислить грузовики марки toyota

goal

truck(toyota,What1,What2,What3,What4).

* Существует ли данный грузовик?

goal

truck(ford,80000,6,blue,8000).

Далее была составлена программа «Родственные отношения». За основу была взята программа из первого примера. Были добавлены отношения «брат», «сестра», «бабушка», «дедушка»:

domains

s=symbol

predicates

nondeterm parent(s,s)

female(s)

male(s)

nondeterm mother(s,s)

nondeterm father(s,s)

ancestor(s,s)

child(s,s)

nondeterm brother(s,s).

nondeterm sister(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).

brother(X,Y):- male(X), father(F,X), father(F,Y), mother(M,X), mother(M,Y), X<>Y.

sister(X,Y):- female(X), father(F,X), father(F,Y), mother(M,X), mother(M,Y), X<>Y.

grand\_father(X,Y):- male(X), parent(P,Y), parent(X,P).

grand\_mother(X,Y):- female(X), parent(P,Y), parent(X,P).

goal

grand\_mother(liz,Who).

Следующим шагом была составлена программа, использующая отношения «нравится» и «может купить»:

predicates

likes(symbol,symbol).

for\_sale(symbol).

can\_afford(symbol,symbol).

can\_buy(symbol,symbol).

clauses

likes(ellen, onions).

likes(john, cheese).

likes(tom, pizza).

likes(tom, cheese).

likes(eric, potatoes).

for\_sale(onions).

for\_sale(cheese).

for\_sale(pizza).

can\_afford(tom,pizza).

can\_afford(eric, potatoes).

can\_buy(X,Y):- for\_sale(Y), likes(X,Y), can\_afford(X,Y).

goal

can\_afford(eric, potatoes).

Наконец, мной была составлена собственная программа, использующая отношения «врач», «пациент» и т. д.:

domains

s=symbol

predicates

nondeterm doctor(s).

nondeterm pacient(s).

speciality(s,s).

nondeterm ill(s,s).

nondeterm has\_pacients(s).

clauses

doctor(john).

doctor(alexander).

doctor(sergey).

pacient(jill).

pacient(marina).

pacient(kate).

speciality(alexander,dysentery).

speciality(sergey,plague).

speciality(john,smallpox).

ill(marina,smallpox).

ill(kate,plague).

ill(jill,dysentery).

has\_pacients(X):- doctor(X), pacient(Z), speciality(X,Y), ill(Z,Y).

goal

has\_pacients (Who).

# Выводы

Выполняя данную лабораторную работу я научилась создавать базовые программы на языке Prolog.