|  |  |  |
| --- | --- | --- |
| МИНОБРНАУКИ РОССИИ  Федеральное государственное  бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |
| Институт Информационных Технологий  Кафедра МОСИТ | | |
| **ЛАБОРАТОРНАЯ РАБОТА** | | |
| по дисциплине «**Функциональное и логическое программирование**» | | |
| **Тема лабораторной работыОСНОВЫ ЯЗЫКА ПРОЛОГ** | | |
| **Студент группы** | ИКБО-12-17 Герман К.М. | *(подпись студента)* |
| **Руководитель работы** | СмольяниноваВ.А. | *(подпись руководителя)* |
| Москва, 2019 | | |

# Постановака задачи

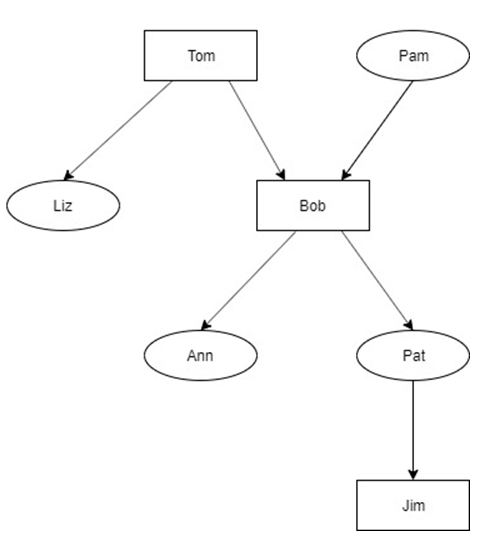
1. Составьте программу Родственные отношения, которая кроме родственных отношений parent (родитель) и ancestor (предок) программа должна содержать одно или несколько из следующих отношений: brother (брат); sister (сестра); grand-father (дедушка); grand-mother (бабушка); uncle (дядя).
2. Составьте программу, используя отношения likes «нравится») и can\_buy («может купить»).

Составьте собственную программу, состоящую из фактов и правил. Проверьте ее работу.

# Ход работы

## Родственные отношения

Схема родственных отношений



Исходный код программы «Родственные отношения»

domains

s=symbol

predicates

nondeterm parent(s,s)

nondeterm ancestor(s,s)

nondeterm brother(s,s)

female(s)

male(s)

mother(s,s)

father(s,s)

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).

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

Тут добавлен предикат brother: Xявляется братом Y, если Xи Yимеют общего предка Z.

Такая программа способна дать ответы, например, на вопросы:

brother(ann, pat).

parent(X, ann).

ancestor(X,tom).

brother(ann, X).

mother(M, Child).

father(F, Child).

## Может купить

Исходный код программы «Родственные отношения». Тут используются предикаты can buy и likes

predicates

can\_buy(symbol, symbol)

person(symbol)

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).

goal

can\_buy(judy, hot\_rod).

Отношение can\_buyпроверяет: хочет ли человек эту вещь и продается ли эта вещь.

Это позволяет проанализировать:

* Что хочет конкретный человек.
* Что может купить конкретный человек.
* Купит ли конкретный человек конкретную вещь.
* Многие другие вопросы.

Примеры некоторых возможных запросов:

Мной была составлена программа, состоящая из фактов и правил и проверена ее работа.

can\_buy(kelly, hot\_rod).

can\_buy(judy, pizza).

likes(kelly, What).

domains

s=symbol

predicates

have(s,s).

likes(s,s).

noalergic(s,s).

clauses

likes(tom,dog).

likes(juli,cat).

likes(dimon,dog)

noalergic(tom,dog).

noalergic(juli,dog).

noalergic(dimon,parrot).

have(tom, dog):-likes(tom,dog),noalergic(tom,dog).

have(juli, cat):- likes(juli,cat), noalergic(juli,cat).

have(dimon, parrot):- likes(dimon,parrot), noalergic(dimon,parrot).

goal

have(tom,dog).

Эта программа позволяет проанализировать:

* Есть ли у определенного человека определенный питомец
* Какого питомца определенный человек любит
* Есть ли аллергия у определенного человека на определенного питомца

# Выводы

Я научился создавать базовые программы на Прологе.