|  |  |  |
| --- | --- | --- |
| 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 работы ОСНОВЫ ЯЗЫКА ПРОЛОГ** | | |
| **Студент группы** | ИКБО-12-17 Драгош С.В. | *(подпись студента)* |
| **Руководитель работы** | Смольянинова В.А. | *(подпись руководителя)* |
| Москва, 2019 | | |

# Постановка задачи

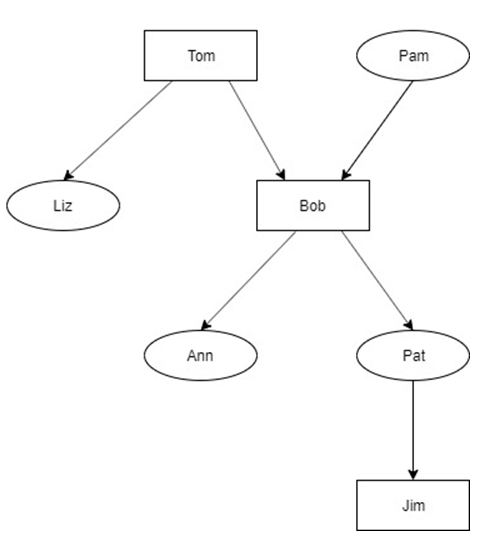
1. Составьте программу Родственные отношения, которая кроме родственных отношений parent (родитель) и ancestor (предок) программа должна содержать одно или несколько из следующих отношений: brother (брат); sister (сестра); grand-father (дедушка); grand-mother (бабушка); uncle (дядя).
2. Составьте программу, 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: X является братом Y, если X и Y имеют 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

Этой программы не было в стандартном наборе примеров.

domains

s=symbol

n=integer

predicates

whants(s, s)

costs(s, n)

money(s, n)

can\_buy(s, s)

clauses

whants(tom, car).

whants(jim, pizza).

costs(pizza, 200).

costs(car, 300000).

money(tom, 4000).

money(jim, 4000).

can\_buy(Person, Item) :- whants(Person, Item), money(Person, Has), costs(Item, Cost), Cost<=Has.

goal

can\_buy(jim, pizza).

Отношение can\_buy проверяет: хочет ли человек эту вещь и достаточно ли у него денег для её покупки.

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:

can\_buy(jim, X).

can\_buy(X, car).

can\_buy(X, Y).

whants(X, Y).

whants(tim, X).

# Выводы

В ходе выполнения этой лабораторной работы я научился создавать простые программы на Прологе.