



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 Попков А.И.

---

*(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).

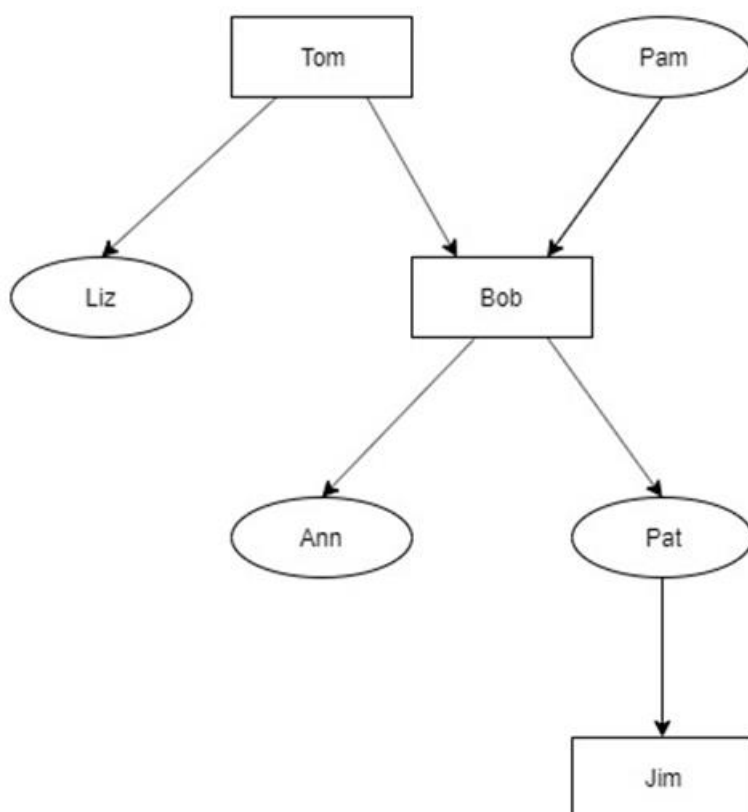
2. Составьте программу, используя отношения likes («нравится») и can\_buy («может купить»).

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)
    female(s)
    male(s)
    mother(s,s)
    father(s,s)
    nondeterm ancestor(s,s)
    child(s,s)
    grandfather(s,s)
    grandmother(s,s)
    nondeterm brother(s,s)
    sister(s, s)
clauses
    parent(pam,bob). /* pam - bob's parent */
    parent(tom,bob).
    parent(tom,liz).
    parent(pam,liz).
    parent(bob,ann).
    parent(bob,pat).
    parent(pat,jim).
    parent(clara,george).
    parent(joseph,george).
    parent(mark,joseph).
    female(pam). /* pam - female */
    female(liz).
    female(ann).
    female(pat).
    female(clara).
    male(tom). /* tom - male */
    male(bob).
    male(jim).
    male(george).
    male(joseph).
    male(mark).
    child(Y,X):- parent(X,Y). /* Y - child of X, if X - parent of Y */
    mother(X,Y):- parent(X,Y),female(X). /* X - mother of Y, if X -
parent of Y and X - female */
    father(X,Y):- parent(X,Y),male(X). /* X - father of Y, if X - parent
of Y and X - male */
    ancestor(X,Z):- parent(X,Z). /* X - ancestor Z, X - relative Z */
    ancestor(X,Z):- parent(X,Y),ancestor(Y,Z).
    grandfather(X,Y):- parent(Z,Y),parent(X,Z),male(X).
    grandmother(X,Y):- parent(Z,Y),parent(X,Z),female(X).
    sister(X,Y):- parent(Z,X), parent(Z,Y),female(X),X<>Y.
    brother(X,Y):- parent(Z,Y),parent(Z,X),male(X),X<>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:

```
grandfather(mark, george).  
male(george).  
child(george, joseph).  
child(clerk, joseph).  
ancestor(mark, george).  
sister(liz, bob).  
brother(bob, liz).
```

### «Машины»

#### Исходный код

```
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).
```

#### Пример запросов

```
goal  
    %truck(Model, Price1, Quantity, black, Price2).  
    car(Model, Price1, 4, Color, Price2).  
    %car(datsun, What1, What2, What3, What4).  
    %truck(toyota, What1, What2, What3, What4).  
    %car(chrysler, 13000, 3, red, 12000).
```

### «Может купить»

#### Исходный код программы:

```
predicates  
    nondeterm can_buy(symbol, symbol)  
    nondeterm 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).
```

### Примеры запросов:

```
goal
    %can_buy(judy, hot_rod).
    %likes(Who,What).
    %car(What).
    %can_buy(Who, lemon).
    person(Who).
```

### «Дни рождения и хобби»

Была реализована программа “Дни рождения и хобби” на языке Пролог  
Исходный код программы:

```
domains
    a=symbol
    b=integer
predicates
    birthday(a,b,a)
    likes(a,a)
clauses
    birthday(nataly, 8, september).
    birthday(yana, 25, august).
    birthday(nina, 28, september).
    birthday(peter, 2, august).
    birthday(ivan, 12, august).
    likes(nataly, books).
    likes(nataly, sport).
    likes(yana, books).
    likes(yana, dances).
    likes(peter, music).
    likes(peter, dances).
    likes(ivan, sport).
    likes(ivan, books).
```

### Примеры запросов

```
goal
    birthday(ivan, Num, Month).
    birthday(Who, Num, september).
    likes(Who, music).
    likes(yana, What).
    birthday(Who, 25, august).
    likes(peter, What).
```

Программа birthday может ответить на такие вопросы как:

- чьи именины в сентябре?
- когда именины у Ивана?
- кто любит музыку?

## **Выводы**

В ходе выполнения лабораторной работы были изучены базовые навыки работы на языке Пролог.