



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«МИРЭА – Российский технологический университет»

**РТУ МИРЭА**

---

Институт Информационных технологий

Кафедра общей информатики

## **ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №4**

**по дисциплине**

«Функциональное программирование»

“Программирование на языке Prolog”

Выполнили студенты группы ИКБО-11-17

Ходарченко В.В.

Принял

Аввакумов Г.

Лабораторная работа выполнена « \_\_\_\_ » \_\_\_\_\_ 2019г. \_\_\_\_\_  
(подпись руководителя)

«Зачтён» « \_\_\_\_ » \_\_\_\_\_ 2019г. \_\_\_\_\_  
(подпись руководителя)

Москва 2019

## Содержание

Содержание.....	2
Цель работы.....	3
Постановка задачи.....	3
Этапы выполнения.....	4

Цель работы

Освоение основных понятий языка Пролог.

Постановка задачи

1. Нарисуйте схему родства. 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.

### Этапы выполнения

1. Нарисуйте схему родства. 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).

Во время анализа программы была составлена схема родства.

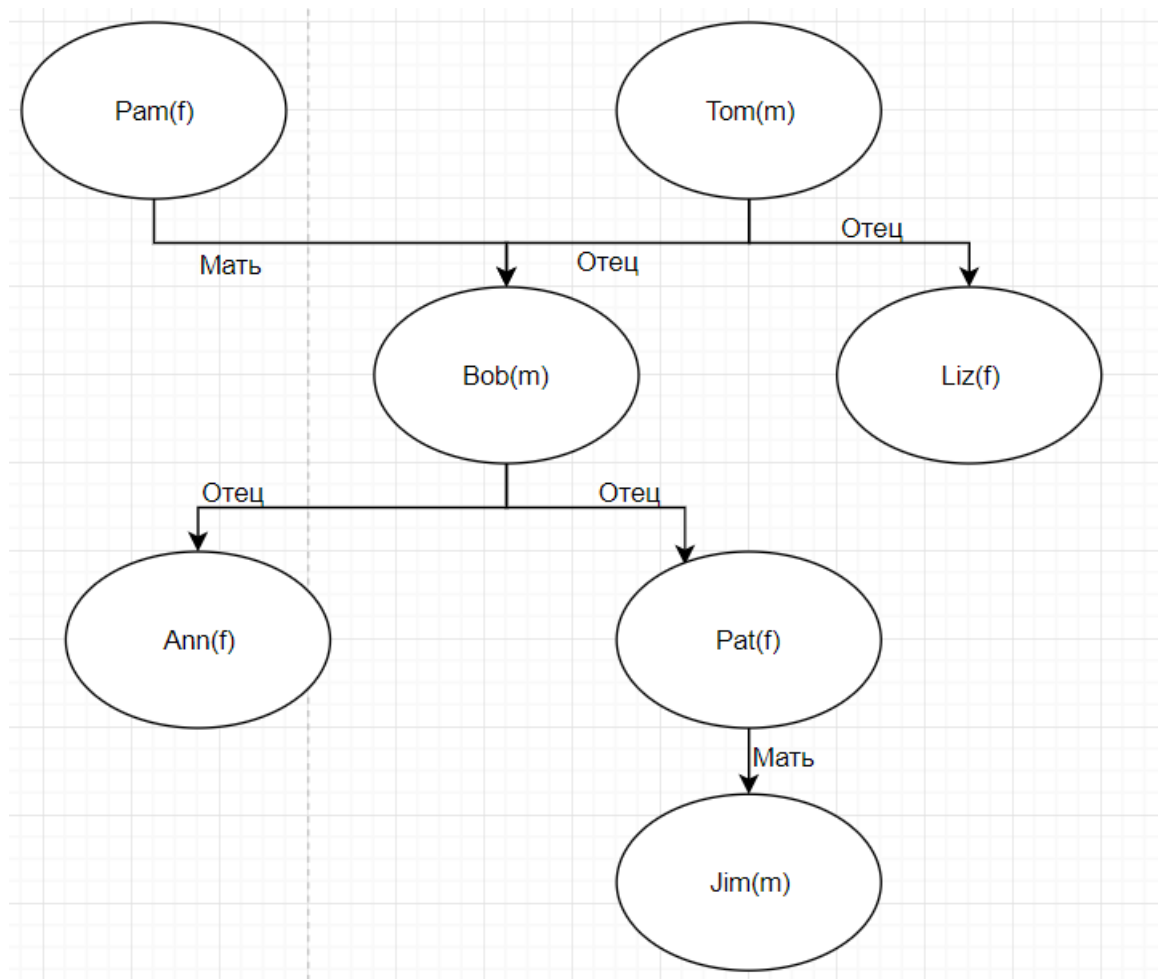


Рис. 1. Схема родства.

Далее была создана программа с новыми реализованными отношениями.

---

```
predicates
nondeterm parent(symbol,symbol)
nondeterm female(symbol)
nondeterm male(symbol)
nondeterm mother(symbol,symbol)
nondeterm father(symbol,symbol)
nondeterm ancestor(symbol,symbol)
```

---

---

---

```

nondeterm child(symbol,symbol)
nondeterm brother(symbol,symbol)
nondeterm sister(symbol,symbol)
nondeterm grandfather(symbol,symbol)
nondeterm grandmother(symbol,symbol)
nondeterm uncle(symbol,symbol)

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), male(X), X<>Y.
    sister(X,Y) :- parent(Z,X), parent(Z,Y), female(X), X<>Y.
    grandfather(X,Y) :- father(X,Z), parent(Z,Y).
    grandmother(X,Y) :- mother(X,Z), parent(Z,Y).
    uncle(X,Y) :- parent(Z,Y), brother(X,Z), X<>Z.

goal
    %sister(X,Y).
    %brother(X,Y).
    %grandfather(X,Y).
    %grandmother(X,Y).
    %uncle(X,Y).

```

---

---

Листинг 1. Исходный код программы.

При последовательном тестировании каждой закомментированной строки выдаёт следующий результат:

- sister(X,Y) результат:  
X=liz, Y=bob

X=ann, Y=pat

X=pat, Y=ann

3 Solutions

- brother(X,Y) результат:

X=bob, Y=liz

1 Solution

- grandfather(X,Y) результат:

X=tom, Y=ann

X=tom, Y=pat

X=bob, Y=jim

3 Solutions

- grandmother(X,Y) результат:

X=pam, Y=ann

X=pam, Y=pat

2 Solutions

- uncle(X,Y) результат:

No Solution

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

---

---

predicates

```
nondeterm likes(symbol,symbol)
nondeterm can_buy(symbol, symbol)
nondeterm person(symbol)
nondeterm exist(symbol)
nondeterm haveMoney(symbol)
```

clauses

```
person(vlad).
person(christina).
person(den).
person(ruslan).
person(eduard).
person(caesar).
person(roman).
```

```
haveMoney(christina).
haveMoney(ruslan).
```

---

---

---

---

```
haveMoney(caesar).
haveMoney(roman).
haveMoney(eduard).

likes(vlad, juice).
likes(christina, milk).
likes(den, apple).
likes(ruslan, orange).
likes(eduard, melon).
likes(caesar, bun).

exist(juice).
exist(milk).
exist(apple).
exist(orange).
exist(bun).

can_buy(X, Y) :-
    person(X),
    exist(Y),
    likes(X, Y),
    haveMoney(X).

goal

can_buy(X, Y).
```

---

---

### Листинг 2. Исходный код программы.

- can\_buy(X, Y). результат:  
X=christina, Y=milk  
X=ruslan, Y=orange  
X=caesar, Y=bun  
3 Solutions

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

---

---

```
predicates
    nondeterm standOnRaid(symbol,symbol)
    nondeterm have(symbol,symbol)
    nondeterm raisedTo(symbol,symbol)
    nondeterm deliveredOn(symbol,symbol)
    nondeterm situateOn(symbol,symbol)
    nondeterm accompany(symbol,symbol)
    nondeterm carryOnshoulder(symbol,symbol)
    nondeterm returnOn(symbol,symbol)
    nondeterm mooredTo(symbol,symbol)
    nondeterm know(symbol,symbol)
```

---

---

---

---

```
nondeterm ask(symbol,symbol)
nondeterm to(symbol,symbol)
nondeterm yellow(symbol)
nondeterm spanish(symbol)
nondeterm placed(symbol,symbol)
nondeterm called(symbol,symbol)
clauses
  yellow(galeon).

  spanish(flag).

  placed(port, puertoRico).
  placed(dock, puertoRico).

  called(port, santiago).

  have(galleon,flag).
  have(galleon,topOfMainmast).
  have(port,dock).
  have(galleon,boat).
  have(stranger,bag).
  have(galleon,board).
  have(governor,house).

  raisedTo(flag,topOfMainmast).

  situateOn(stranger,galleon).
  situateOn(servant,port).
  situateOn(stranger,dock) :-
    deliveredOn(stranger,dock).
  situateOn(boat,dock) :-
    deliveredOn(stranger,dock).
  situateOn(stranger,port) :-
    situateOn(stranger,dock),
    have(port,dock),
    situateOn(servant,port),
    accompany(servant,stranger),
    carryOnShoulder(servant,bag),
    have(stranger,bag),
    placed(port, puertoRico),
    called(port, santiago).

  accompany(servant,stranger).

  carryOnShoulder(servant,bag).

  ask(stranger,commander).

  to(way,house).

  know(commander,way).
  know(stranger,way) :-
```

---

---



---

```

        know(commander,way),
        situateOn(stranger,port),
        ask(stranger,commander),
        to(way,house),
        have(governor,house).

standOnRaid(galleon,port) :-
    have(galleon,flag),
    have(galleon,topOfMainmast),
    raisedTo(flag,topOfMainmast),
    yellow(galeon),
    spanish(flag),
    placed(port, puertoRico),
    called(port, santiago).

deliveredOn(stranger,dock) :-
    standOnRaid(galleon,port),
    have(galleon,boat),
    situateOn(stranger,galleon),
    have(port,dock),
    placed(dock, puertoRico).

returnOn(boat,galleon) :-
    situateOn(stranger,dock),
    situateOn(boat,dock).

mooredTo(boat,board) :-
    returnOn(boat,galleon),
    have(galleon,board).

goal
    standOnRaid(galleon,port),
    deliveredOn(X, Y),
    situateOn(X, Y),
    situateOn(Z, Y),
    situateOn(X, port),
    returnOn(Z, galleon),
    mooredTo(Z, board),
    know(X, way).

```

---

### Листинг 3. Исходный код программы.

```

standOnRaid(galleon,port),
deliveredOn(X, Y),
situateOn(X, Y),
situateOn(Z, Y),
situateOn(X, port),
returnOn(Z, galleon),
mooredTo(Z, board),
know(X, way). результат:

```

X=stranger, Y=dock, Z=boat  
1 Solution