



**Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)**

---

ФАКУЛЬТЕТ \_\_\_\_\_ «Информатика и системы управления»

КАФЕДРА \_\_\_\_\_ «Программное обеспечение ЭВМ и информационные технологии»

## Лабораторная работа №13 Работа программы на Prolog

Студент: Луговой Д.М.

Группа: ИУ7-61Б

Преподаватель: Толпинская Н.Б.

Москва, 2020 г.

**Цель работы:** получить навыки построения модели предметной области, разработки и оформления программы на Prolog, изучить принципы, логику формирования программы и отдельные шаги выполнения программы на Prolog.

**Задачи работы:** приобрести навыки декларативного описания предметной области с использованием фактов и правил. Изучить способы использования термов, переменных, фактов и правил в программе на Prolog, принципы и правила сопоставления и отождествления, порядок унификации.

## Задание

Составить программу, т.е. модель предметной области – базу знаний, объединив в ней информацию – знания:

- «Телефонный справочник»: Фамилия, №тел, Адрес – структура (Город, Улица, №дома, №кв),
- «Автомобили»: Фамилия\_владельца, Марка, Цвет, Стоимость, и др.,
- «Вкладчики банков»: Фамилия, Банк, счет, сумма, др.

Владелец может иметь несколько телефонов, автомобилей, вкладов (Факты). Используя правила, обеспечить возможность поиска:

1. (а) По № телефона найти: Фамилию, Марку автомобиля, Стоимость автомобиля (может быть несколько),  
(б) Используя сформированное в пункте а) правило, по № телефона найти: только Марку автомобиля (автомобилей может быть несколько),
2. Используя простой, не составной вопрос: по Фамилии (уникальна в городе, но в разных городах есть однофамильцы) и Городу проживания найти: Улицу проживания, Банки, в которых есть вклады и №телефона.

Для задания1 и задания2: для одного из вариантов ответов, и для а) и для в), описать словесно порядок поиска ответа на вопрос, указав, как выбираются знания, и, при этом, для каждого этапа унификации, выписать подстановку – наибольший общий унификатор, и соответствующие примеры термов.

## Текст программы

### Листинг 1: База знаний

```
1 domains
2 name, phone, city, street, color, brand, money, bank, account = symbol.
3 house, apartment = integer.
4 addr = address(city, street, house, apartment).
5
6 predicates
7 phonebook(name, phone, addr).
8 car(name, brand, color, money).
9 depositor(name, bank, account, money).
10 find_name_brand_money(phone, name, brand, money).
11 find_street_bank_phone(name, city, street, bank, phone).
12
13 clauses
14 find_name_brand_money(Phone, Name, Brand, Money) :- phonebook(Name,
15     Phone, _), car(Name, Brand, _, Money).
16 find_street_bank_phone(Name, City, Street, Bank, Phone) :-
17     phonebook(Name, Phone, address(City, Street, _, _)), depositor(Name,
18     Bank, _, _).
19
20 phonebook("Ivanov", "79836457823", address("Moscow", "Tverskaya", 4,
21     112)).
22 phonebook("Sidorov", "79285920831", address("Tver", "Orlova", 17, 22)).
23 phonebook("Ivanov", "79260112361", address("Moscow", "Tverskaya", 4,
24     112)).
25 phonebook("Petrov", "79256239576", address("St-Petersburg",
26     "Leninskaya", 19, 26)).
27 phonebook("Sidorov", "79278456344", address("Moscow", "Puskinskaya", 2,
28     34)).
29
30 car("Petrov", "BMW", "Black", "3500000").
31 car("Ivanov", "Mercedes", "Black", "5000000").
32 car("Sidorov", "BMW", "Red", "2000000").
33 car("Petrov", "Audi", "Blue", "3000000").
34
35 depositor("Sidorov", "Sberbank", "1238123127", "5000000").
36 depositor("Ivanov", "Tinkoff", "5872874928", "300000").
37 depositor("Sidorov", "VTB", "123123213", "2000000").
38 depositor("Petrov", "Sberbank", "123213213", "3000000").
```

Предикат *find\_name\_brand\_money(phone, name, brand, money)* обеспечивает возможность поиска имени, марки автомобиля и его стоимости по номеру телефона, предикат *find\_street\_bank\_phone(name, city, street, bank, phone)*. позволяет найти улицу проживания, банк и номер телефона по имени и городу.

## Примеры работы:

goal	Результат
find_name_brand_money("79260112361", Name, Brand, Money).	Name=Ivanov, Brand=Mercedes, Money=5000000 1 Solution
find_name_brand_money("79256239576", _, Brand, _).	Brand=BMW Brand=Audi 2 Solutions
find_street_bank_phone("Sidorov", "Moscow", Street, Bank, Phone).	Street=Puskinskaya, Bank=Sberbank, Phone=79278456344 Street=Puskinskaya, Bank=VTB, Phone=79278456344 2 Solutions

Порядок формирования результата для 1-го вопроса:

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
1	Сравниваются find_name_brand_money( "79260112361", Name, Brand, Money) и find_name_brand_money(Phone, Name, Brand, Money), подстановка - {Phone="79260112361"}	Прямой ход
2	Сравниваются phonebook(Name, "79260112361", _) и find_name_brand_money(Phone, Name, Brand, Money), они имеют разные функторы	Термы не унифицируемы, переход к следующему предложению
3	Сравниваются phonebook(Name, "79260112361", _) и find_street_bank_phone(Name, City, Street, Bank, Phone), они имеют разные функторы	Термы не унифицируемы, переход к следующему предложению
4	Сравниваются phonebook(Name, "79260112361", _) и phonebook("Ivanov", "79836457823", address("Moscow", "Tverskaya", 4, 112)), термы "79260112361" и "79836457823" не унифицируемы	Термы не унифицируемы, переход к следующему предложению
5	Сравниваются phonebook(Name, "79260112361", _) и phonebook("Sidorov", "79285920831", address("Tver", "Orlova", 17, 22)), термы "79260112361" и "79285920831" не унифицируемы	Термы не унифицируемы, переход к следующему предложению

6	Сравниваются phonebook(Name, "79260112361", _) и phonebook("Ivanov", "79260112361", address("Moscow", "Tverskaya", 4, 112)), подстановка - {Name="Ivanov", Phone="79260112361", _=address("Moscow", "Tverskaya", 4, 112)}	Прямой ход, занесение Name="Ivanov" в результирующую ячейку
7	Сравниваются car("Ivanov", Brand, _, Money) и find_name_brand_money(Phone, Name, Brand, Money), они имеют разные функторы	Термы не унифицируемы, переход к следующему предложению
8	Сравниваются car("Ivanov", Brand, _, Money) и find_street_bank_phone(Name, City, Street, Bank, Phone), они имеют разные функторы	Термы не унифицируемы, переход к следующему предложению
9	Сравниваются car("Ivanov", Brand, _, Money) и phonebook("Ivanov", "79836457823", address("Moscow", "Tverskaya", 4, 112)), они имеют разные функторы	Термы не унифицируемы, переход к следующему предложению
10	Сравниваются car("Ivanov", Brand, _, Money) и phonebook("Sidorov", "79285920831", address("Tver", "Orlova", 17, 22)), они имеют разные функторы	Термы не унифицируемы, переход к следующему предложению
11	Сравниваются car("Ivanov", Brand, _, Money) и phonebook("Ivanov", "79260112361", address("Moscow", "Tverskaya", 4, 112)), они имеют разные функторы	Термы не унифицируемы, переход к следующему предложению
12	Сравниваются car("Ivanov", Brand, _, Money) и phonebook("Petrov", "79256239576", address("St-Petersburg", "Leninskaya", 19, 26)), они имеют разные функторы	Термы не унифицируемы, переход к следующему предложению
13	Сравниваются car("Ivanov", Brand, _, Money) и phonebook("Sidorov", "79278456344", address("Moscow", "Puskinskaya", 2, 34)), они имеют разные функторы	Термы не унифицируемы, переход к следующему предложению
14	Сравниваются car("Ivanov", Brand, _, Money) и car("Petrov", "BMW", "Black", "3500000"), термы "Ivanov" и "Petrov" не унифицируемы	Термы не унифицируемы, переход к следующему предложению
15	Сравниваются car("Ivanov", Brand, _, Money) и car("Ivanov", "Mercedes", "Black", "5000000"), подстановка - {Brand="Mercedes", _="Black", Money="5000000"}	Прямой ход, занесение Brand="Mercedes", Money="5000000" в результирующую ячейку

16	Результат: подстановка {Name="Ivanov", Brand="Mercedes", Money="5000000"}	
----	---	--

Порядок формирования результата для 2-го вопроса:

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
1	Сравниваются find_name_brand_money("79256239576", _, Brand, _) и find_name_brand_money(Phone, Name, Brand, Money), подстановка - {Phone="79256239576"}	Прямой ход
2	Сравниваются phonebook(Name, "79256239576", _) и find_name_brand_money(Phone, Name, Brand, Money), они имеют разные функторы	Термы не унифицируемы, переход к следующему предложению
3	Сравниваются phonebook(Name, "79256239576", _) и find_street_bank_phone(Name, City, Street, Bank, Phone), они имеют разные функторы	Термы не унифицируемы, переход к следующему предложению
4	Сравниваются phonebook(Name, "79256239576", _) и phonebook("Ivanov", "79836457823", address("Moscow", "Tverskaya", 4, 112)), термы "79256239576" и "79836457823" не унифицируемы	Термы не унифицируемы, переход к следующему предложению
5	Сравниваются phonebook(Name, "79256239576", _) и phonebook("Sidorov", "79285920831", address("Tver", "Orlova", 17, 22)), термы "79256239576" и "79285920831" не унифицируемы	Термы не унифицируемы, переход к следующему предложению
6	Сравниваются phonebook(Name, "79256239576", _) и phonebook("Ivanov", "79260112361", address("Moscow", "Tverskaya", 4, 112)), термы "79256239576" и "79260112361" не унифицируемы	Термы не унифицируемы, переход к следующему предложению
7	Сравниваются phonebook(Name, "79256239576", _) и phonebook("Petrov", "79256239576", address("St-Petersburg", "Leninskaya 19, 26)), подстановка - {Name="Petrov", _=address("St-Petersburg", "Leninskaya 19, 26)}	Прямой ход

8	Сравниваются car("Petrov", Brand, _, Money) и find_name_brand_money(Phone, Name, Brand, Money), они имеют разные функторы	Термы не унифицируемы, переход к следующему предложению
9	Сравниваются car("Petrov", Brand, _, Money) и find_street_bank_phone(Name, City, Street, Bank, Phone), они имеют разные функторы	Термы не унифицируемы, переход к следующему предложению
10	Сравниваются car("Petrov", Brand, _, Money) и phonebook("Ivanov", "79836457823", address("Moscow", "Tverskaya", 4, 112)), они имеют разные функторы	Термы не унифицируемы, переход к следующему предложению
11	Сравниваются car("Petrov", Brand, _, Money) и phonebook("Sidorov", "79285920831", address("Tver", "Orlova", 17, 22)), они имеют разные функторы	Термы не унифицируемы, переход к следующему предложению
12	Сравниваются car("Petrov", Brand, _, Money) и phonebook("Ivanov", "79260112361", address("Moscow", "Tverskaya", 4, 112)), они имеют разные функторы	Термы не унифицируемы, переход к следующему предложению
13	Сравниваются car("Petrov", Brand, _, Money) и phonebook("Petrov", "79256239576", address("St-Petersburg", "Leninskaya", 19, 26)), они имеют разные функторы	Термы не унифицируемы, переход к следующему предложению
14	Сравниваются car("Petrov", Brand, _, Money) и phonebook("Sidorov", "79278456344", address("Moscow", "Puskinskaya", 2, 34)), они имеют разные функторы	Термы не унифицируемы, переход к следующему предложению
15	Сравниваются car("Petrov", Brand, _, Money) и car("Petrov", "BMW", "Black", "3500000"), подстановка - {Brand="BMW", _="Black", Money="3500000"}	Прямой ход, занесение Brand="BMW" в результирующую ячейку
16	Результат: подстановка {Brand="BMW"}	

Порядок формирования результата для 3-го вопроса:

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
1	Сравниваются find_street_bank_phone("Sidorov", "Moscow", Street, Bank, Phone) и find_name_brand_money(Phone, Name, Brand, Money), они имеют разные функторы	Термы не унифицируемы, переход к следующему предложению
2	Сравниваются find_street_bank_phone("Sidorov", "Moscow", Street, Bank, Phone) и find_street_bank_phone(Name, City, Street, Bank, Phone), подстановка - {Name="Sidorov", City="Moscow"}	Прямой ход
3	Сравниваются phonebook("Sidorov", Phone, address("Moscow", Street, _, _)) и find_name_brand_money(Phone, Name, Brand, Money), они имеют разные функторы	Термы не унифицируемы, переход к следующему предложению
4	Сравниваются phonebook("Sidorov", Phone, address("Moscow", Street, _, _)) и find_street_bank_phone(Name, City, Street, Bank, Phone), они имеют разные функторы	Термы не унифицируемы, переход к следующему предложению
5	Сравниваются phonebook("Sidorov", Phone, address("Moscow", Street, _, _)) и phonebook("Ivanov", "79836457823", address("Moscow", "Tverskaya", 4, 112)), термы "Sidorov" и "Ivanov" не унифицируемы	Термы не унифицируемы, переход к следующему предложению
6	Сравниваются phonebook("Sidorov", Phone, address("Moscow", Street, _, _)) и phonebook("Sidorov", "79285920831", address("Tver", "Orlova", 17, 22)), термы address("Moscow", Street, _, _) и address("Tver", "Orlova", 17, 22) не унифицируемы, так как термы "Moscow" и "Tver" не унифицируемы	Термы не унифицируемы, переход к следующему предложению
7	Сравниваются phonebook("Sidorov", Phone, address("Moscow", Street, _, _)) и phonebook("Ivanov", "79260112361", address("Moscow", "Tverskaya", 4, 112)), термы "Sidorov" и "Ivanov" не унифицируемы	Термы не унифицируемы, переход к следующему предложению



8	Сравниваются phonebook("Sidorov", Phone, address("Moscow", Street, _, _)) и phonebook("Petrov", "79256239576", address("St-Petersburg", "Leninskaya", 19, 26)), термы "Sidorov" и "Petrov" не унифицируемы	Термы не унифицируемы, откат, переход к следующему предложению
9	Сравниваются phonebook("Sidorov", Phone, address("Moscow", Street, _, _)) и phonebook("Sidorov", "79278456344", address("Moscow", "Puskinskaya", 2, 34)), подстановка {Phone="79278456344", Street="Puskinskaya", _=2, _=34}	Прямой ход, занесение Phone="79278456344", Street="Puskinskaya" в результирующую ячейку
10	Сравниваются depositor("Sidorov", Bank, _, _) и find_name_brand_money(Phone, Name, Brand, Money), они имеют разные функторы	Термы не унифицируемы, переход к следующему предложению
11	Сравниваются depositor("Sidorov", Bank, _, _) и find_street_bank_phone(Name, City, Street, Bank, Phone), они имеют разные функторы	Термы не унифицируемы, переход к следующему предложению
12	Сравниваются depositor("Sidorov", Bank, _, _) и phonebook("Ivanov", "79836457823", address("Moscow", "Tverskaya", 4, 112)), они имеют разные функторы	Термы не унифицируемы, переход к следующему предложению
13	Сравниваются depositor("Sidorov", Bank, _, _) и phonebook("Sidorov", "79285920831", address("Tver", "Orlova", 17, 22)), они имеют разные функторы	Термы не унифицируемы, переход к следующему предложению
14	Сравниваются depositor("Sidorov", Bank, _, _) и phonebook("Ivanov", "79260112361", address("Moscow", "Tverskaya", 4, 112)), они имеют разные функторы	Термы не унифицируемы, переход к следующему предложению
15	Сравниваются depositor("Sidorov", Bank, _, _) и phonebook("Petrov", "79256239576", address("St-Petersburg", "Leninskaya", 19, 26)), они имеют разные функторы	Термы не унифицируемы, переход к следующему предложению
16	Сравниваются depositor("Sidorov", Bank, _, _) и phonebook("Sidorov", "79278456344", address("Moscow", "Puskinskaya", 2, 34)), они имеют разные функторы	Термы не унифицируемы, переход к следующему предложению
17	Сравниваются depositor("Sidorov", Bank, _, _) и car("Petrov", "BMW", "Black", "3500000"), они имеют разные функторы	Термы не унифицируемы, переход к следующему предложению

18	Сравниваются <code>depositor("Sidorov", Bank, _, _)</code> и <code>car("Ivanov", "Mercedes", "Black", "5000000")</code> , они имеют разные функторы	Термы не унифицируемы, переход к следующему предложению
19	Сравниваются <code>depositor("Sidorov", Bank, _, _)</code> и <code>car("Sidorov", "BMW", "Red", "2000000")</code> , они имеют разные функторы	Термы не унифицируемы, переход к следующему предложению
20	Сравниваются <code>depositor("Sidorov", Bank, _, _)</code> и <code>car("Petrov", "Audi", "Blue", "3000000")</code> , они имеют разные функторы	Термы не унифицируемы, переход к следующему предложению
21	Сравниваются <code>depositor("Sidorov", Bank, _, _)</code> и <code>depositor("Sidorov", "Sberbank", "1238123127", "5000000")</code> , подстановка <code>{Bank="Sberbank", _="1238123127", _="5000000"}</code>	Прямой ход, занесение <code>Bank="Sberbank"</code> в результирующую ячейку
22	Результат: подстановка <code>{Phone="79278456344", Street="Puskinskaya", Bank="Sberbank"}</code>	

## Теоретические вопросы

### Что такое терм?

Терм - основной элемент языка Prolog. Терм – это:

#### 1. Константа:

- Число (целое, вещественное),
- Символьный атом (комбинация символов латинского алфавита, цифр и символа подчеркивания, начинающаяся со строчной буквы),
- Строка: последовательность символов, заключенных в кавычки.

#### 2. Переменная:

- Именованная – обозначается комбинацией символов латинского алфавита, цифр и символа подчеркивания, начинающейся с прописной буквы или символа подчеркивания,
- Анонимная - обозначается символом подчеркивания

#### 3. Составной терм: Это средство организации группы отдельных элементов знаний в единый объект, синтаксически представляется: $f(t_1, t_2, \dots, t_m)$ , где $f$ - функтор (отношение между объектами), $t_1, t_2, \dots, t_m$ – термы, в том числе и составные.

## Что такое предикат в матлогике (математике)?

Предикат в математической логике - это утверждение, высказанное о субъекте. Предикат является функцией со значениями 0, 1 (истина/ложь), определенной на некотором множестве параметров. Предикат называю n-арным, если он определен на n-ой декартовой степени множества M.

## Что описывает предикат в Prolog?

Предикат в Prolog описывает отношение между аргументами процедуры. Процедурой в Prolog является совокупность всех правил, описывающих определенное отношение.

**Назовите виды предложений в программе и приведите примеры таких предложений из Вашей программы. Какие предложения являются основными, а какие – не основными? Каковы: синтаксис и семантика (формальный смысл) этих предложений (основных и неосновных)?**

В Prolog есть два типа предложений: правила и факты. Правило имеет вид:  $A :- B_1, \dots, B_n$ . A называется заголовком правила, а  $B_1, \dots, B_n$  – телом правила. Заголовок содержит некоторое знание, а тело - условие истинности этого знания. Факт является частным случаем правила - в нем отсутствует тело.

Пример факта из программы: *car("Petrov", "BMW", "Black", "3500000")*.

Пример правила из программы: *find\_name\_brand\_money(Phone, Name, Brand, Money) :- phonebook(Name, Phone, \_), car(Name, Brand, \_, Money)*.

Основными называются предложения, не содержащие переменных. Предложения, содержащие переменные называются неосновными.

Синтаксис предложения: заголовок(составной терм) :- тело(один или последовательность термов). Предложения используются для формирования базы знаний о некоторой предметной области.

**Каковы назначение, виды и особенности использования переменных в программе на Prolog? Какое предложение БЗ сформулировано в более общей – абстрактной форме: содержащее или не содержащее переменных?**

Переменные предназначены для обозначения некоторого неизвестного объекта предметной области. Переменные бывают именованными и анонимными. Именованные переменные уникальны в рамках предложения, а анонимная переменная – любая уникальна. В разных предложениях может использоваться одно имя переменной для обозначения разных объектов.

В ходе выполнения программы выполняется связывание переменных с различными объектами, этот процесс называется конкретизацией. Это относится только к именованным переменным. Анонимные переменные не могут быть связаны со значением.

В более общей форме сформулировано предложение, содержащее переменные, так как заранее неизвестно, каким объектом будет конкретизирована переменная.

### **Что такое подстановка?**

Пусть дан терм:  $(X_1, X_2, \dots, X_n)$ . Подстановка - множество пар, вида:  $\{X_i = t_i\}$ , где  $X_i$  - переменная, а  $t_i$  - терм.

### **Что такое пример терма? Как и когда строится? Как Вы думаете, система строит и хранит примеры?**

Пусть  $\Theta = \{X_1 = t_1, X_2 = t_2, \dots, X_n = t_n\}$  - подстановка,  $A$  - терм. Результат применения подстановки к терму обозначается  $A\Theta$ .

Примером терма  $A$  называется терм  $B$ , если существует подстановка  $\Theta$  такая, что  $B = A\Theta$ .

Примеры термов строятся в ходе логического вывода. Для построения примера терма его переменные конкретизируются.