



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»

**Отчет по лабораторной работе №11 - 13  
по дисциплине «Функциональное и логическое  
программирование»**

Тема Среда Visual Prolog. Структура программы. Работа программы

Студент Романов А.В.

Группа ИУ7-63Б

Оценка (баллы)

Преподаватель Толпинская Н.Б., Строганов Ю. В.

# Лабораторная работа №11

## Постановка задачи

Разработать свою программу — «Телефонный справочник». Протестировать работы программы.

## Решение

```
1 domains
2   name = string
3   city = string
4   street = string
5   home = integer
6   phone = integer
7
8 predicates
9   record(name, city, street, home, phone)
10
11 clauses
12   record("Alexey", "Krasnogorsk", "Lesnaya", 15, 8965258).
13   record("Pavel", "Moscow", "Inessa Armand", 79, 8996666).
14   record("Dmitry", "Moscow", "Golubinskaya", 140, 8853535).
15   record("Sergey", "Moscow", "Dorms 5", 655, 8578728).
16   record("Mikhail", "Ekaterinburg", "Sovetskaya", 140, 8920909).
17
18 goal
19   record(Name, "Moscow", Street, Home, Phone).
```

# Лабораторная работа №12

## Постановка задачи

Составить программу — базу знаний, с помощью которой можно определить, например, множество студентов, обучающихся в одном ВУЗе. Студент может одновременно обучаться в нескольких ВУЗах. Привести примеры возможных вариантов вопросов и варианты ответов (не менее 3-х), Описать порядок формирования вариантов ответа.

## Решение

```
1 domains
2   name, surname, university, group = string
3   course = integer
4
5 predicates
6   student(name, surname, university, group, course)
7
8 clauses
9   student("Alexey", "Romanov", "BMSTU", "IU7-63B", 3).
10  student("Pavel", "Perestoronin", "BMSTU", "IU7-63B", 5).
11  student("Ivan", "Cvetkov", "BMSTU", "IU7-43B", 2).
12  student("Alexey", "Voyakin", "BMSTU", "IU7-64B", 3).
13  student("Alexey", "Romanov", "MSU", "VMK-1", 1).
14  student("Pavel", "Perestoronin", "MAI", "EM-8", 2).
15  student("Ilya", "Bryukhov", "MAI", "ACS12", 2).
16  student("Vladimir", "Nesterov", "MIPT", Group, Course)
17  :- student("Alexey", "Romanov", "BMSTU", Group, _), student("Pavel", "
    Perestoronin", "BMSTU", Group, Course).
18
19 goal
20  student(Name, Surname, "MIPT", _, _).
21  student(Name, Surname, "BMSTU", _, 3).
22  student("Alexey", \_, \_, "IU7-63B", \_)
```

Данная база знаний содержит информацию о студентах (имя, фамилия, вуз, группа, курс).

С помощью первого вопроса получают все студенты, которые учатся в МФТИ. Происходит проход сверху вниз по всем фактам предиката *student(name, surname, university, group,*

*course*) и осуществляется унификация с *study(Name, Surname, "BMSTU", \_, \_)*. Унификацию успешно проходит один факт: *study("Vladimir", "Nesterov", "BMSTU", "IU7-63B", 5)*.

С помощью второго вопроса получаются все студенты МГТУ, которые обучаются на 3 курсе. Происходит проход по всем фактам предиката *student(name, surname, university, group, course)* и осуществляется унификация с *study(Name, Surname, "BMSTU", \_, 3)*. Успешно унификацию проходят факты *study("Alexey", "Romanov", "BMSTU", \_, 3)* и *study("Alexey", "Voyakin", "BMSTU", \_, 3)*.

С помощью второго вопроса получаются все Алексей, которые обучаются в группе ИУ7-63Б. Происходит проход по всем фактам предиката *student(name, surname, university, group, course)* и осуществляется унификация с *study(Name, \_, \_, "IU7-63B", \_)*. Успешно унификацию проходит *study("Alexey", "Romanov", "BMSTU", "IU7-63B", 3)*.

# Лабораторная работа №13

## Постановка задачи

Составить программу, то есть модель предметной области — базу знаний, объединив в ней информацию — знания:

- «Телефонный справочник»: фамилия, № телефона, адрес - структура (город, улица, № дома, № квартиры);
- «Автомобили»: фамилия владельца, марка, цвет, стоимость и др.;
- «Вкладчики банков»: фамилия, банк, счет, сумма и др.

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

1.
  - по № телефона найти: фамилию, марку автомобиля, стоимость автомобиль (может быть несколько);
  - 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 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 унификатор, и соответствующие примеры термов.

## Решение

```
1 domains
2   surname, phone, city, street, house, apartment = string
3   address = address(city, street, house, apartment)
4   model, color, cost = string
5   bank, sum = string
6
7 predicates
```

```

8  man(surname, phone, address)
9  car(surname, model, color, cost)
10 deposit(surname, bank, sum)
11
12 car_by_phone(phone, surname, model, cost)
13 car_model_by_phone(phone, model)
14 bank_and_street_by_surname_and_city(surname, city, bank, street)
15
16 clauses
17  man("Vladimir", "890955550987", address("Moscow", "Sovetskaya", "134", "15
18    ")).
19  man("Mikhail", "8100500321", address("Khimki", "Lesnaya", "27", "501")).
20  man("Pavel", "87654329867", address("Moscow", "Tikhaya", "105", "52")).
21  man("Alexey", "89096412389", address("Krasnogorsk", "Lesnaya", "17", "5"))
22  .
23  man("Alexey", "89096421381", address("Moscow", "Baumanskaya", "170", "1"))
24  .
25
26 car("Pavel", "Nissan", "White", "709000").
27 car("Mikhail", "Cadillac", "Black", "1000000").
28 car("Alexey", "AMG", "Black", "100000").
29
30 deposit("Pavel", "Sber", "0").
31 deposit("Mikhail", "Alpha", "10").
32 deposit("Alexey", "Tinkoff", "100500").
33
34 car_by_phone(Phone, Surname, Model, Cost)
35   :- man(Surname, Phone, _), car(Surname, Model, _, Cost).
36
37 car_model_by_phone(Phone, Model) :- car_by_phone(Phone, _, Model, _).
38
39 bank_and_street_by_surname_and_city(Surname, City, Bank, Street)
40   :- man(Surname, _, address(City, Street, _, _)), deposit(Surname, Bank,
41     _).
42
43 goal
44  %car_by_phone("89096412389", Surname, Model, Cost).
45  %car_model_by_phone("89096412389", Model).
46  bank_and_street_by_surname_and_city("Alexey", "Moscow", Bank, Street).

```

Порядок формирования результата для 1-го вопроса:

№ шага	Сравниваемые термы; результаты; подстановка, если есть	Дальнейшие действия
1	Сравниваются car_by_phone("89096412389 Surname, Model, Cost) и car_by_phone(Phone, Surname, Model, Cost). Подстановка (Phone - "89096412389")	Прямой ход
2	Сравниваются man(Surname, "89096412389", _) и car_by_phone(Phone, Surname, Model, Cost). Они имеют разные функторы.	Термы не унифицируемы. Переход к следующему предложению
3	Сравниваются man(Surname, "89096412389", _) и car_model_by_phone(Phone, Model). Они имеют разные функторы.	Термы не унифицируемы. Переход к следующему предложению
4	Сравниваются man(Surname, "89096412389", _) и man("Vladimir", "890955550987", address(. "Moscow", "Sovetskaya", "134", "15")). Термы не унифицируемы.	Термы не унифицируемы. Переход к следующему предложению.
5	Сравниваются man(Surname, "89096412389", _) и man("Vladimir", "8100500321", address(. "Moscow", "Sovetskaya", "27", "501")). Термы не унифицируемы.	Термы не унифицируемы. Переход к следующему предложению.
6	Сравниваются man(Surname, "89096412389", _) и man("Vladimir", "87654329867", address(. "Moscow", "Tikhaya", "105", "52")). Термы не унифицируемы.	Термы не унифицируемы. Переход к следующему предложению.
7	Сравниваются man(Surname, "89096412389", _) и man("Vladimir", "89096421381", address(. "Moscow", "Baumanskaya", "170", "1")). Термы не унифицируемы.	Термы не унифицируемы. Переход к следующему предложению.
8	Сравниваются man(Surname, "89096412389", _) и man("Alexey", "89096412389", address(. "Krasnogorsk", "Lesnaya", "17", "5")). Подстановка (Surname = "Alexey Phone = "89096412389", _ = address("Krasnogorsk", "Lesnaya", "17", "5"))	Прямой ход. Занесение Surname = "Alexey" в ячейку
9	Сравниваются car("Alexey", Brand, _, Cost) и car_by_phone(Phone, Surname, Model, Cost). Они имеют разные функторы.	Термы не унифицируемы. Переход к следующему предложению
10	Сравниваются car("Alexey", Brand, _, Cost) и car_model_by_phone(Phone, Model). Они имеют разные функторы.	Термы не унифицируемы. Переход к следующему предложению
11	Сравниваются car("Alexey", Brand, _, Cost) и man("Vladimir", "890955550987", address(. "Moscow", "Sovetskaya", "134", "15")). Разные функторы.	Термы не унифицируемы. Переход к следующему предложению.

12	Сравниваются car("Alexey", Brand, _, Cost) и man("Vladimir", "8100500321", address("Moscow", "Sovetskaya", "27", "501")). Разные функторы.	Термы не унифицируемы. Переход к следующему предложению.
13	Сравниваются car("Alexey", Brand, _, Cost) и man("Vladimir", "87654329867", address("Moscow", "Tikhaya", "105", "52")). Разные функторы.	Термы не унифицируемы. Переход к следующему предложению.
14	Сравниваются car("Alexey", Brand, _, Cost) и man("Vladimir", "89096421381", address("Moscow", "Baumanskaya", "170", "1")). Разные функторы.	Термы не унифицируемы. Переход к следующему предложению.
15	Сравниваются car("Alexey", Brand, _, Cost) и man("Vladimir", "89096421381", address("Moscow", "Baumanskaya", "170", "1")). Разные функторы.	Термы не унифицируемы. Переход к следующему предложению.
16	Сравниваются car("Alexey", Brand, _, Cost) и car("Pavel", "Nissan", "White", "709000"). Термы не унифицированы.	Термы не унифицируемы. Переход к следующему предложению.
17	Сравниваются car("Alexey", Brand, _, Cost) и car("Mikhail", "Cadillac", "Black", "1000000"). Термы не унифицированы.	Термы не унифицируемы. Переход к следующему предложению.
18	Сравниваются car("Alexey", Brand, _, Cost) и car("Alexey", "AMG", "Black", "100000"). Подстановка (Model = "AMG", _ = "Black", Cost = "100000")	Прямой ход. Занесение Model = "AMG", Cost = "100000" в результирующую ячейку.
19	Результат: подстановка (Name = "Alexey", Model = "AMG", Cost = "100000")	

Порядок формирования результата для 2-го вопроса:



№ шага	Сравниваемые термы; результаты; подстановка, если есть	Дальнейшие действия
1	Сравниваются car_by_phone(Phone, Surname, Model, Cost) и car_by_phone("89096412389", Brand, _). Подстановка (Phone - "89096412389")	Прямой ход
2	Сравниваются man(Surname, "89096412389", _) и car_by_phone(Phone, Surname, Model, Cost). Они имеют разные функторы.	Термы не унифицируемы. Переход к следующему предложению
3	Сравниваются man(Surname, "89096412389", _) и car_model_by_phone(Phone, Model). Они имеют разные функторы.	Термы не унифицируемы. Переход к следующему предложению
4	Сравниваются man(Surname, "89096412389", _) и man("Vladimir", "890955550987", address(. "Moscow", "Sovetskaya", "134", "15")). Термы не унифицируемы.	Термы не унифицируемы. Переход к следующему предложению.
5	Сравниваются man(Surname, "89096412389", _) и man("Vladimir", "8100500321", address(. "Moscow", "Sovetskaya", "27", "501")). Термы не унифицируемы.	Термы не унифицируемы. Переход к следующему предложению.
6	Сравниваются man(Surname, "89096412389", _) и man("Vladimir", "87654329867", address(. "Moscow", "Tikhaya", "105", "52")). Термы не унифицируемы.	Термы не унифицируемы. Переход к следующему предложению.
7	Сравниваются man(Surname, "89096412389", _) и man("Vladimir", "89096421381", address(. "Moscow", "Baumanskaya", "170", "1")). Термы не унифицируемы.	Термы не унифицируемы. Переход к следующему предложению.
8	Сравниваются man(Surname, "89096412389", _) и man("Alexey", "89096412389", address(. "Krasnogorsk", "Lesnaya", "17", "5")). Подстановка (Surname = "Alexey Phone = "89096412389", _ = address("Krasnogorsk", "Lesnaya", "17", "5"))	Прямой ход. Занесение Surname = "Alexey" в ячейку
9	Сравниваются car("Alexey", Brand, _, Cost) и car_by_phone(Phone, Surname, Model, Cost). Они имеют разные функторы.	Термы не унифицируемы. Переход к следующему предложению
10	Сравниваются car("Alexey", Brand, _, Cost) и car_model_by_phone(Phone, Model). Они имеют разные функторы.	Термы не унифицируемы. Переход к следующему предложению
11	Сравниваются car("Alexey", Brand, _, Cost) и man("Vladimir", "890955550987", address(. "Moscow", "Sovetskaya", "134", "15")). Разные функторы.	Термы не унифицируемы. Переход к следующему предложению.

12	Сравниваются car("Alexey", Brand, _, Cost) и man("Vladimir", "8100500321", address("Moscow", "Sovetskaya", "27", "501")). Разные функторы.	Термы не унифицируемы. Переход к следующему предложению.
13	Сравниваются car("Alexey", Brand, _, Cost) и man("Vladimir", "87654329867", address("Moscow", "Tikhaya", "105", "52")). Разные функторы.	Термы не унифицируемы. Переход к следующему предложению.
14	Сравниваются car("Alexey", Brand, _, Cost) и man("Vladimir", "89096421381", address("Moscow", "Baumanskaya", "170", "1")). Разные функторы.	Термы не унифицируемы. Переход к следующему предложению.
15	Сравниваются car("Alexey", Brand, _, Cost) и man("Vladimir", "89096421381", address("Moscow", "Baumanskaya", "170", "1")). Разные функторы.	Термы не унифицируемы. Переход к следующему предложению.
16	Сравниваются car("Alexey", Brand, _, Cost) и car("Pavel", "Nissan", "White", "709000"). Термы не унифицированы.	Термы не унифицируемы. Переход к следующему предложению.
17	Сравниваются car("Alexey", Brand, _, Cost) и car("Mikhail", "Cadillac", "Black", "1000000"). Термы не унифицированы.	Термы не унифицируемы. Переход к следующему предложению.
18	Сравниваются car("Alexey", Brand, _, Cost) и car("Alexey", "AMG", "Black", "100000"). Подстановка (Model = "AMG", _ = "Black", Cost = "100000")	Прямой ход. Занесение Model = "AMG" в результирующую ячейку.
19	Результат: подстановка (Model = "AMG")	

Порядок формирования результата для 3-го вопроса:

№ шага	Сравниваемые термы; результаты; подстановка, если есть	Дальнейшие действия
1	Сравниваются bank_and_street("Alexey", "Moscow", Bank) и car_by_phone(Phone, Surname, Model, Cost). Разные функторы.	Термы не унифицируемы. Переход к следующему предложению.
2	Сравниваются bank_and_street("Alexey", "Moscow", Bank, Street) и bank_and_street(Name, City, Street, Bank, Street). Подстановка (Name = "Alexey", City = "Moscow")	Прямой ход
3	Сравниваются man("Alexey", Phone, address( "Moscow", Street, _, _)). и car_by_phone(Phone, Surname, Model, Cost). Разные функторы.	Термы не унифицируемы. Переход к следующему предложению
4	Сравниваются man("Alexey", Phone, address( "Moscow", Street, _, _)). и bank_and_street(Name, City, Street, Bank) Разные функторы.	Термы не унифицируемы. Переход к следующему предложению
5	Сравниваются man("Alexey", Phone, address( "Moscow", Street, _, _)). и man("Alexey", "890955550987", address("Moscow", "Sovetskaya", "134", "15")). Термы не унифицируемы.	Термы не унифицируемы. Переход к следующему предложению.
6	Сравниваются man("Alexey", Phone, address( "Moscow", Street, _, _)). и man("Mikhail", "8100500321", address("Khimki", "Lesnaya", "27", "501")). Термы не унифицируемы.	Термы не унифицируемы. Переход к следующему предложению.
7	Сравниваются man("Alexey", Phone, address( "Moscow", Street, _, _)). и man("Pavel", "87654329867", address("Moscow", "Tikhaya", "105", "52")). Термы не унифицируемы.	Термы не унифицируемы. Переход к следующему предложению.
8	Сравниваются man("Alexey", Phone, address( "Moscow", Street, _, _)). и man("Alexey", "89096412389", address("Krasnogorsk", "Lesnaya", "17", "5")). Термы не унифицируемы.	Термы не унифицируемы. Переход к следующему предложению.
9	Сравниваются man("Alexey", Phone, address( "Moscow", Street, _, _)). и man("Alexey", "89096421381", address("Moscow", "Baumanskaya", "170", "1")). Подстановка (Phone = "89096421381", Street = "Baumanskaya",) _ = "170", _ = "34")	Прямой ход, занесение Phone = "89096421381" Street = "Baumanskaya" в результирующую ячейку
10	Сравниваются deposit("Alexey", Bank, _) и car_by_phone(Phone, Surname, Model, Cost). Разные функторы.	Термы не унифицируемы. Переход к следующему предложению
11	Сравниваются deposit("Alexey", Bank, _) и man("Alexey", "890955550987", address("Moscow", "Moscow", "Sovetskaya", "134", "15")). "Sovetskaya", "134", "15")). Разные функторы	Термы не унифицируемы. Переход к следующему Предложению

12	Сравниваются deposit("Alexey", Bank, _) и man("Alexey", "890955550987", address("Moscow", "Sovetskaya", "134", "15")). Разные функторы	Термы не унифицируемы. Переход к следующему предложению
13	Сравниваются deposit("Alexey", Bank, _) и man("Mikhail", "8100500321", address("Khimki", "Lesnaya", "27", "501")). Разные функторы	Термы не унифицируемы. Переход к следующему предложению.
14	Сравниваются deposit("Alexey", Bank, _) и man("Pavel", "87654329867", address("Moscow", "Tikhaya", "105", "52")). Разные функторы	Термы не унифицируемы. Переход к следующему предложению.
15	Сравниваются deposit("Alexey", Bank, _) и man("Alexey", "89096412389", address("Krasnogorsk", "Lesnaya", "17", "5")). Разные функторы	Термы не унифицируемы. Переход к следующему предложению.
16	Сравниваются deposit("Alexey", Bank, _) и man("Alexey", "89096421381", address("Moscow", "Baumanskaya", "170", "1")). Разные функторы	Термы не унифицируемы. Переход к следующему предложению
17	Сравниваются deposit("Alexey", Bank, _) и car("Pavel", "Nissan", "White", "709000") Разные функторы.	Термы не унифицируемы. Переход к следующему предложению
18	Сравниваются deposit("Alexey", Bank, _) и car("Mikhail", "Cadillac", "Black", "1000000") Разные функторы.	Термы не унифицируемы. Переход к следующему предложению
19	Сравниваются deposit("Alexey", Bank, _) и car("Alexey", "AMG", "Black", "100000") Разные функторы.	Термы не унифицируемы. Переход к следующему предложению
20	Сравниваются deposit("Alexey", Bank, _) и deposit("Pavel", "Sber", "0") Термы не унифицируемы.	Термы не унифицируемы. Переход к следующему предложению
21	Сравниваются deposit("Alexey", Bank, _) и deposit("Mikhail", "Alpha", "10") Термы не унифицируемы.	Термы не унифицируемы. Переход к следующему предложению
22	Сравниваются deposit("Alexey", Bank, _) и deposit("Alexey", "Tinkoff", "100500"). Подстановка (Bank = "Tinkoff", _ = "100500")	Прямой ход Занесение Bank = "Tinkoff" в результирующую ячейку.
23	Результат: подстановка (Street = "Baumanskaya", Bank = "Tinkoff")	

# Теоретическая часть

## 1. Что собой представляет программа на языке пролог?

Программа на Prolog представляет собой набор фактов и правил, обеспечивающих получение заключений на основе этих утверждений. Программа содержит базу знаний и вопрос. База знаний содержит истинные значения, используя которые программа выдает ответ на вопрос.

Основным элементом языка является терм. База знаний состоит из предложений. Каждое предложение заканчивается точкой. Вопрос состоит только из тела – составного терма (или нескольких составных термов). Вопросы используются для выяснения выполнимости некоторого отношения между описанными в программе объектами. Система рассматривает вопрос как цель, к которой (к истинности которой) надо стремиться. Ответ на вопрос может оказаться логически положительным или отрицательным, в зависимости от того, может ли быть достигнута соответствующая цель.

## 2. Какова структура программы на Prolog?

Программа на Prolog состоит из следующих разделов:

- директивы компилятора — зарезервированные символьные константы,
- CONSTANTS — раздел описания констант,
- DOMAINS — раздел описания доменов,
- DATABASE — раздел описания предикатов внутренней базы данных,
- PREDICATES — раздел описания предикатов,
- CLAUSES — раздел описания предложений базы знаний,
- GOAL — раздел описания внутренней цели (вопроса). В программе не обязательно должны быть все разделы.

## 3. Как реализуется программа на Prolog? Как формируются результаты работы программы?

Ответ на поставленный вопрос система дает в логической форме - «Да» или «Нет». Цель системы состоит в том, чтобы на поставленный вопрос найти возможность, исходя из ба-

зы знаний, ответить «Да». Вариантов ответить «Да» на поставленный вопрос может быть несколько. В нашем случае система настроена в режим получения всех возможных вариантов ответа. При поиске ответов на вопрос рассматриваются альтернативные варианты и находятся все возможные решения (методом проб и ошибок) - множества значений переменных, при которых на поставленный вопрос можно ответить - «Да».

Для выполнения логического вывода используется механизм унификации, встроенный в систему. Унификация – операция, которая позволяет формализовать процесс логического вывода. С практической точки зрения - это основной вычислительный шаг, с помощью которого происходит:

- Двухнаправленная передача параметров процедурам,
- Неразрушающее присваивание,
- Проверка условий (доказательство).

В процессе работы система выполняет большое число унификаций. Попытка "увидеть одинаковость" – сопоставимость двух термов, может завершаться успехом или тупиковой ситуацией (неудачей). В последнем случае включается механизм отката к предыдущему шагу.

## 4. 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.

## 5. 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  $M$ .

## 6. Что описывает предикат в Prolog?

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

## 7. 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 (формальных 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("Mikhail", "Cadillac", "Black", "500000")`.
- Пример правила из программы: `car_by_phone(Phone, Surname, Model, Cost) :- man(Surname, Phone, _), car(Name, Model, _, Cost)`.

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.

## 8. 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.

## 9. 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.