



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

*По предмету: «Логическое и Функциональное
Программирование»*

Тема:

Работа программы на Prolog

Студент: Юмаев Артур Русланович
Группа: ИУ7-65Б

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),

в) 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.

Программа – база знаний

```
domains
sname = symbol
% hasBankDeposit
bank = symbol
account = symbol
amount = real
% record
phoneNum = symbol
address = address(symbol city, symbol street, integer house, integer apNum)
city = symbol
street = symbol
% owns
brand = symbol
color = symbol
price = real

predicates
record(sname, phoneNum, address)
owns(sname, brand, color, price)
hasBankDeposit(sname, bank, account, amount)

% Rules
viewCar(phoneNum, sname, brand, price)
brandOnly(phoneNum, brand)
viewData(sname, city, street, bank, phoneNum)

clauses
owns(yumaev, toyota, white, 14000).
owns(yumaev, mercedes, black, 40000).
owns(petrov, renault, black, 45000).
owns(vetrov, toyota, black, 30000).
record(petrov, "89068887745", address(moscow, yuzhnaya, 14, 24)).
record(yumaev, "89036609896", address(moscow, yuzhnaya, 15, 22)).
record(yumaev, "89036600505", address(moscow, yuzhnaya, 15, 22)).
record(yumaev, "89851101112", address(ryazan, soltsevo, 13, 14)).
record(vetrov, "89068886695", address(vologda, vetrovaya, 1, 86)).
record(zhorin, "89068887745", address(moscow, lubanka, 36, 78)).
record(kotova, "89068880045", address(ryazan, lenina, 13, 2)).
hasBankDeposit(yumaev, rocket_bank, "123456789", 12345).
hasBankDeposit(yumaev, sberbank, "326856789", 15000).
hasBankDeposit(petrov, rocket_bank, "123456000", 45678).
hasBankDeposit(kotova, sberbank, "326556000", 0.0).

% Rules
viewCar(PhoneNum, Sname, Brand, Price) :- record(Sname, PhoneNum, _),
                                           owns(Sname, Brand, _, Price).

brandOnly(PhoneNum, Brand) :- viewCar(PhoneNum, _, Brand, _).

viewData(Sname, City, Street, Bank, PhoneNum) :- record(Sname,
                                                         PhoneNum,
                                                         address(City, Street, _, _)),
                                                         hasBankDeposit(Sname, Bank, _, _).

goal
% Task 1.a
% viewCar("89036609896", Sname, Brand, Price).
% Task 1.b
% brandOnly("89036609896", Brand).
% Task 2
viewData(yumaev, moscow, Street, Bank, PhoneNum).
```

Таблица унификации

1.a

```
viewCar("89036609896", Sname, Brand, Price).
```

```
viewCar(PhoneNum, Sname, Brand, Price) :- record(Sname, PhoneNum, _),
                                             owns(Sname, Brand, _, Price).
```

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
1	<pre>viewCar("89036609896", Sname, Brand, Price).</pre> <pre>viewCar(PhoneNum, Sname, Brand, Price) :- ...</pre> <p>Могут быть унифицированы</p> <p>Происходит унификация (переменные не конкретизированные)</p> <ol style="list-style-type: none"> 1. PhoneNum = "89036609896" (PhoneNum конкретизируется) 2. Sname = Sname 3. Brand = Brand 4. Price = Price <p>Результат: <pre>viewCar("89036609896", Sname, Brand, Price)</pre></p>	<p>Унификация выполнена. Прямой ход. Установка в качестве подцели первого предиката правила <pre>viewCar</pre>.</p> <pre>record(Sname, PhoneNum, _)</pre>
2	<p>Унификация <pre>record(Sname, "89036609896", _)</pre> с первым фактом из базы знаний <pre>record(petrov, "89068887745", address(...))</pre>.</p> <ol style="list-style-type: none"> 1. Sname = petrov 2. "89036609896" не равно "89068887745" <p>Результат: унификация не выполнена</p>	<p>Прямой ход. Переход к следующему факту.</p>
3	<p>Унификация <pre>record(Sname, "89036609896", _)</pre> с первым фактом из базы знаний <pre>record(yumaev, "89036609896", address(...))</pre>.</p> <ol style="list-style-type: none"> 1. Sname = yumaev 2. Константа "89036609896" совпадает с "89036609896" 3. _ игнорируется <p>Унификация выполнена.</p> <p>Результат: <pre>record(yumaev, "89036609896", _)</pre></p>	<p>Прямой ход. Переход к следующему предикату <pre>owns(Sname, Brand, _, Price)</pre>.</p>
4	<p>Установка в качестве подцели <pre>owns(Sname, Brand, _, Price)</pre>.</p> <p>Унификация <pre>owns(yumaev, Brand, _, Price)</pre>.</p> <p>и</p> <pre>owns(yumaev, toyota, white, 14000).</pre> <p>Унификация:</p> <ol style="list-style-type: none"> 1. yumaev и yumaev две одинаковые константы 2. Sname унифицируется с toyota и конкретизируется значением toyota 3. _ игнорируется 4. Price унифицируется с 14000 и конкретизируется значением 14000 <p>Унификация выполнена.</p> <p>Результат <pre>owns(yumaev, toyota, _, 14000)</pre>.</p>	<p>Sname = yumaev Brand = toyota Price = 14000</p> <p>Откат. К унификации <pre>owns(Sname, Brand, _, Price)</pre>.</p> <p>Поиск альтернативных решений</p>

5	<p>Установка в качестве подцели <code>owns (Sname, Brand, _, Price) .</code></p> <p>Унификация <code>owns (yumaev, Brand, _, Price) .</code> и <code>owns (yumaev, mercedes, black, 40000) .</code></p> <p>Унификация:</p> <ol style="list-style-type: none"> 1. <code>yumaev</code> и <code>yumaev</code> две одинаковые константы 2. <code>Sname</code> унифицируется с <code>mercedes</code> и конкретизируется значением <code>mercedes</code> 3. <code>_</code> игнорируется 4. <code>Price</code> унифицируется с <code>40000</code> и конкретизируется значением <code>40000</code> <p>Унификация выполнена. Результат <code>owns (yumaev, mercedes, _, 40000) .</code></p>	<p><code>Sname = yumaev</code> <code>Brand = mercedes</code> <code>Price = 40000</code></p> <p>Откат. К унификации <code>owns (Sname, Brand, _, Price) .</code></p> <p>Поиск альтернативных решений</p>
6 - 8	<p>Установка в качестве подцели <code>owns (Sname, Brand, _, Price) .</code></p> <p>Унификация со следующими двумя фактами из базы знаний не приведет к унификации, а приведет к двум откатам.</p>	<p>Откат.</p> <p>Откат к унификации <code>record (Sname, PhoneNum, _)</code></p>
9 – 14	<p>Установка в качестве подцели <code>record (Sname, "89036609896", _)</code></p> <p>Унификация со следующими пятью фактами из базы знаний не приведет к унификации по переменной <code>PhoneNum</code>.</p>	<p>Откат.</p>
15	<p>Результат</p> <p><code>Sname=yumaev, Brand=toyota, Price=14000</code> <code>Sname=yumaev, Brand=mercedes, Price=40000</code> 2 Solutions</p>	

1.b

Унификация `brandOnly ("89036609896", Brand) .`

и `brandOnly (PhoneNum, Brand) :- viewCar (PhoneNum, _, Brand, _)`.

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
1	<p><code>brandOnly ("89036609896", Brand) .</code> <code>brandOnly (PhoneNum, Brand)</code></p> <p>Могут быть унифицированы</p> <p>Происходит унификация (переменные не конкретизированные)</p> <ol style="list-style-type: none"> 1. <code>PhoneNum = "89036609896"</code> (<code>PhoneNum</code> конкретизируется) 2. <code>Brand = Brand</code> <p>Результат: <code>brandOnly ("89036609896", Brand) .</code></p>	<p>Унификация выполнена. Прямой ход. Установка в качестве подцели первого предиката <code>viewCar (PhoneNum, _, Brand, _)</code>.</p>
2	<p>Унификация <code>viewCar ("89036609896", _, Brand, _)</code> <code>viewCar (PhoneNum, _, Brand, _)</code></p> <p>Унификация</p> <ol style="list-style-type: none"> 1. <code>PhoneNum = "89036609896"</code> 2. <code>_</code> игнорируется 3. <code>Brand = Brand</code> 4. <code>_</code> игнорируется 	<p>Унификация выполнена. Прямой ход. Установка в качестве подцели первого предиката правила <code>viewCar</code>. <code>record (Sname, PhoneNum, _)</code></p>

	Результат: viewCar("89036609896", _, Brand, _).	
3	<p>Унификация record(Sname, "89036609896", _) с первым фактом из базы знаний record(petrov, "89068887745", address(...)).</p> <ol style="list-style-type: none"> 1. Sname = petrov 2. "89036609896" не равно "89068887745" <p>Результат: унификация не выполнена</p>	Прямой ход. Переход к следующему факту.
4	<p>Унификация record(Sname, "89036609896", _) с первым фактом из базы знаний record(yumaev, "89036609896", address(...)).</p> <ol style="list-style-type: none"> 1. Sname = yumaev 2. Константа "89036609896" совпадает с "89036609896" 3. _ игнорируется <p>Унификация выполнена. Результат: record(yumaev, "89036609896", _)</p>	Прямой ход. Переход к следующему предикату owns(Sname, Brand, _, Price).
5	<p>Установка в качестве подцели owns(Sname, Brand, _, Price).</p> <p>Унификация owns(yumaev, Brand, _, Price). и owns(yumaev, toyota, white, 14000).</p> <p>Унификация:</p> <ol style="list-style-type: none"> 1. yumaev и yumaev две одинаковые константы 2. Sname унифицируется с toyota и конкретизируется значением toyota 3. _ игнорируется 4. Price унифицируется с 14000 и конкретизируется значением 14000 <p>Унификация выполнена. Результат owns(yumaev, toyota, _, 14000).</p>	<p>Sname = yumaev Brand = toyota Price = 14000</p> <p>Откат. К унификации owns(Sname, Brand, _, Price).</p> <p>Поиск альтернативных решений</p>
6	<p>Установка в качестве подцели owns(Sname, Brand, _, Price).</p> <p>Унификация owns(yumaev, Brand, _, Price). и owns(yumaev, mercedes, black, 40000).</p> <p>Унификация:</p> <ol style="list-style-type: none"> 1. yumaev и yumaev две одинаковые константы 2. Sname унифицируется с mercedes и конкретизируется значением mercedes 3. _ игнорируется 4. Price унифицируется с 40000 и конкретизируется значением 40000 <p>Унификация выполнена. Результат owns(yumaev, mercedes, _, 40000).</p>	<p>Sname = yumaev Brand = mercedes Price = 40000</p> <p>Откат. К унификации owns(Sname, Brand, _, Price).</p> <p>Поиск альтернативных решений</p>
7 - 9	<p>Установка в качестве подцели owns(Sname, Brand, _, Price).</p> <p>Унификация со следующими двумя фактами из базы знаний не приведет к унификации, а приведет к двум откатам.</p>	Откат. Откат к унификации record(Sname, PhoneNum, _)
10 – 15	Установка в качестве подцели record(Sname, "89036609896", _)	Откат.

	Унификация со следующими пятью фактами из базы знаний не приведет к унификации по переменной PhoneNum.	
16	Результат Brand=toyota Brand=mercedes 2 Solutions	

2.

Унификация `viewData(yumaev, moscow, Street, Bank, PhoneNum)`.

`viewData(Sname, City, Street, Bank, PhoneNum) :-`

`record(Sname, PhoneNum, address(City, Street, _, _)),`

`hasBankDeposit(Sname, Bank, _, _).`

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
1	<p>Унифицируется <code>viewData(yumaev, moscow, Street, Bank, PhoneNum)</code> . и <code>viewData(Sname, City, Street, Bank, PhoneNum)</code></p> <ol style="list-style-type: none"> 1. Sname унифицируется с yumaev (далее факт унификации обозначается знаком “=”), не конкретизированные переменные 2. City = moscow, city контретизируется moscow 3. Street = Street 4. Bank = Bank 5. PhoneNum <p>Результат: <code>viewData(yumaev, moscow, Street, Bank, PhoneNum)</code> .</p>	<p>Унификация прошла успешно, переход к первой подцели</p> <p><code>record(Sname, PhoneNum, address(City, Street, _, _))</code></p>
2	<p>Унификация <code>record(yumaev, PhoneNum, address(moscow, Street, _, _))</code> и <code>record(petrov, "89068887745", address(moscow, yuzhnaya, 14, 24))</code> .</p> <ol style="list-style-type: none"> 1. Petrov и yumaev разные константы <p>Унификация не выполнена.</p>	<p>Прямой ход. Переход к следующему факту.</p>
3	<p>Унификация предикатов одинаковой арности <code>record(yumaev, PhoneNum, address(moscow, Street, _, _))</code> и <code>record(yumaev, "89036609896", address(moscow, yuzhnaya, 15, 22))</code> .</p> <ol style="list-style-type: none"> 1. yumaev и yumaev – две одинаковые константы 2. PhoneNum = "89036609896" 3. moscow и moscow – две одинаковые константы 4. Street = yuzhnaya 5. _ игнорируется 6. _ игнорируется <p>Результат: <code>record(yumaev, "89036609896", address(moscow, yuzhnaya, _, _))</code> .</p>	<p>Унификация выполнена успешно. Переход к следующей подцели</p> <p><code>hasBankDeposit(Sname, Bank, _, _)</code> .</p>

4	<p>Унификация</p> <p>hasBankDeposit(yumaev, Bank, _, _) и hasBankDeposit(yumaev, rocket_bank, "123456789", 12345).</p> <ol style="list-style-type: none"> 1. yumaev и yumaev две одинаковые константы 2. Bank = rocket_bank 3. _ игнорируется 4. _ игнорируется <p>Результат: hasBankDeposit(yumaev, rocket_bank, _, _).</p>	<p>Унификация выполнена.</p> <p>Street=yuzhnaya Bank=rocket_bank PhoneNum="89036609896"</p> <p>Откат к hasBankDeposit(yumaev, Bank, _, _)</p>
5	<p>Унификация</p> <p>hasBankDeposit(yumaev, Bank, _, _) и hasBankDeposit(yumaev, sberbank, "326856789", 15000).</p> <ol style="list-style-type: none"> 1. yumaev и yumaev две одинаковые константы 2. Bank = sberbank 3. _ игнорируется 4. _ игнорируется <p>Результат: hasBankDeposit(yumaev, sberbank, _, _).</p>	<p>Унификация выполнена.</p> <p>Street=yuzhnaya Bank=sberbank PhoneNum="89036609896"</p> <p>Откат к hasBankDeposit(yumaev, Bank, _, _)</p>
6-7	<p>Слудующие 2 факта не будут унифицированы</p>	<p>Унификация не выполнена.</p> <p>Откат к record(yumaev, PhoneNum, address(moscow, Street, _, _))</p>
8	<p>Унификация предикатов одинаковой арности</p> <p>record(yumaev, PhoneNum, address(moscow, Street, _, _)) и record(yumaev, "89036600505", address(moscow, yuzhnaya, 15, 22)).</p> <ol style="list-style-type: none"> 1. yumaev и yumaev – две одинаковые константы 2. PhoneNum = "89036600505" 3. moscow и moscow – две одинаковые константы 4. Street = yuzhnaya 5. _ игнонируется 6. _ игнорируется <p>Результат: record(yumaev, "89036600505", address(moscow, yuzhnaya, _, _)).</p>	<p>Унификация выполнена успешно. Переход к следующей подцели</p> <p>hasBankDeposit(Sname, Bank, _, _).</p>
9	<p>Унификация</p> <p>hasBankDeposit(yumaev, Bank, _, _) и hasBankDeposit(yumaev, rocket_bank, "123456789", 12345).</p> <ol style="list-style-type: none"> 1. yumaev и yumaev две одинаковые константы 2. Bank = rocket_bank 3. _ игнорируется 4. _ игнорируется 	<p>Унификация выполнена.</p> <p>Street=yuzhnaya Bank=rocket_bank PhoneNum="89036600505"</p> <p>Откат к hasBankDeposit(yumaev, Bank, _, _)</p>

	<p>Результат:</p> <pre>hasBankDeposit(yumaev, rocket_bank, _, _).</pre>	
10	<p>Унификация</p> <pre>hasBankDeposit(yumaev, Bank, _, _)</pre> <p>и</p> <pre>hasBankDeposit(yumaev, sberbank, "326856789", 15000).</pre> <ol style="list-style-type: none"> 1. yumaev и yumaev две одинаковые константы 2. Bank = sberbank 3. _ игнорируется 4. _ игнорируется <p>Результат:</p> <pre>hasBankDeposit(yumaev, sberbank, _, _).</pre>	<p>Унификация выполнена.</p> <p>Street=yuzhnaya Bank=sberbank PhoneNum="89036600505"</p> <p>Откат к</p> <pre>hasBankDeposit(yumaev, Bank, _, _)</pre>
11-15	<p>Следующие 4 терма не будут унифицированные.</p>	<p>Унификация не выполнена.</p> <p>Откат к</p> <pre>record(Sname, PhoneNum, address(City, Street, _, _)).</pre>
16	<pre>Street=yuzhnaya, Bank=rocket_bank, PhoneNum=89036609896 Street=yuzhnaya, Bank=sberbank, PhoneNum=89036609896 Street=yuzhnaya, Bank=rocket_bank, PhoneNum=89036600505 Street=yuzhnaya, Bank=sberbank, PhoneNum=89036600505 4 Solutions</pre>	

Примеры вопросов и ответов

```
viewCar("89036609896", Sname, Brand, Price).
```

```
Sname=yumaev, Brand=toyota, Price=14000  
Sname=yumaev, Brand=mercedes, Price=40000  
2 Solutions
```

```
brandOnly("89036609896", Brand).
```

```
Brand=toyota  
Brand=mercedes  
2 Solutions
```

```
viewData(yumaev, moscow, Street, Bank, PhoneNum).
```

```
Street=yuzhnaya, Bank=rocket_bank, PhoneNum=89036609896  
Street=yuzhnaya, Bank=sberbank, PhoneNum=89036609896  
Street=yuzhnaya, Bank=rocket_bank, PhoneNum=89036600505  
Street=yuzhnaya, Bank=sberbank, PhoneNum=89036600505  
4 Solutions
```

Ответы на вопросы

1. Что такое терм?

Термом называется выражение, образованное из переменных и констант, возможно, с применением функций, а точнее:

1. всякая переменная или константа есть терм;
2. если t_1, \dots, t_n — термы, а f — n -местный функциональный символ, то $f(t_1, \dots, t_n)$ терм;
3. других термов нет.

2. Что такое предикат в матлогике (математике)?

Предикатом называется функция, аргументы которой принимают значения из некоторого множества, а сама функция – значение 0 («ложь») или 1 («истина»).

3. Что описывает предикат в Prolog?

Предикаты в Пролог это утверждения программы. Структура предиката – это структура знания, отраженного в заголовке правил. Структура предиката описывается в разделе PREDICATES.

4. 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)?

Предложения бывают двух видов: факты, правила.

Пример факта из программы: `owns(yumaev, toyota, white, 14000).`

Пример правила из программы:

```
brandOnly(PhoneNum, Brand) :- viewCar(PhoneNum, _, Brand, _).
```

Если составные термы, факты, правила и вопросы не содержат переменных, то они называются основными. Составные термы, факты, правила и вопросы в момент фиксации в программе могут содержать переменные, тогда они называются неосновными.

5. 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 в Пролог бывают двух видов: именованные и анонимные.

Именованная – обозначается комбинацией символов латинского алфавита, цифр и символа подчеркивания, начинающейся с прописной буквы или символа подчеркивания (X , $A21$, $_X$). Анонимная - обозначается символом подчеркивания ($_$). Переменная может быть аргументом или параметром основного терма. Переменные предназначены для передачи значений «во времени и в пространстве». Именованные переменные уникальны в рамках предложения, а анонимная переменная – любая уникальна. В разных предложениях может использоваться одно имя переменной для обозначения разных объектов. В процессе выполнения программы переменные могут связываться с различными объектами – конкретизироваться. Это относится только к именованным переменным. Анонимные переменные не могут быть связаны со значением. В более абстрактной форме сформулировано предложение, содержащие переменные, так как неизвестно какое примет значение переменная.

6. Что такое подстановка?

Пусть дан терм: $A(X_1, X_2, X_n)$

Подстановкой называется множество пар, вида: $\{x_i = t_i\}$,
где x_i – переменная, а t_i – терм.

7. 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, тогда результат применения подстановки к терму обозначается: $A\Theta$. Применение подстановки заключается в замене **каждого** вхождения переменной x_i на соответствующий терм. Терм **В** называется **примером** терма **А**, если существует такая подстановка Θ , что $B = A\Theta$. Пролог не хранит значения, он только связывается со значением константы на время, пока не будет достигнута заданная цель (подцель), т.е. ее истинность или ложь. Примеры термов строятся при доказательстве цели или при вычислении подцелей, хранятся примеры до получения ответа, либо до момента «отката» в стеке.