



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

---

ФАКУЛЬТЕТ «Информатика и системы управления»

---

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

---

## Лабораторная работа № 15

Дисциплина	Функциональное и логическое программирование.
Тема	Структура программы на Prolog и ее реализация
Студент	Степанов А. О.
Группа	ИУ7-63Б
Оценка (баллы)	
Преподаватель	Толпинская Н.Б.

Москва, 2020 г.

## ЗАДАНИЕ

Создать базу знаний «Собственники», дополнив базу знаний, хранящую знания (лаб. 13):

- «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.,
- знаниями о дополнительной собственности владельца. Преобразовать знания об автомобиле к форме знаний о собственности.
- д собственности (кроме автомобиля):
- Строение, стоимость и другие его характеристики;
- Участок, стоимость и другие его характеристики;
- Водный\_транспорт, стоимость и другие его характеристики.

Описать и использовать вариантный домен: Собственность. Владелец может иметь, но только один объект каждого вида собственности (это касается и автомобиля), или не иметь некоторых видов собственности.

Используя конъюнктивное правило и разные формы задания одного вопроса (пояснять для какого №задания – какой вопрос), обеспечить возможность поиска:

1. Названий всех объектов собственности заданного субъекта,
2. Названий и стоимости всех объектов собственности заданного субъекта,
3. \* Разработать правило, позволяющее найти суммарную стоимость всех объектов собственности заданного субъекта.

### Листинг 1: Текст программы

```
1 domains
2     lastname , phonenumber = string .
3
```

```

4      city, street = string.
5      house, flat = integer.
6      address = address(city, street, house, flat).
7
8      mark, color = string.
9      cost = integer.
10
11     bank, bankaccount = string.
12     sum = integer.
13
14     area = integer.
15
16     type = string.
17
18     property = car(cost, mark, color);
19               building(cost, area, city);
20               territory(cost, area, city);
21               boat(cost, type, mark, color).
22
23 predicates
24     telephone(lastname, phonenumber, address).
25     deposit(lastname, bank, bankaccount, sum, city).
26     owner(lastname, city, property).
27
28     allProperties(lastname, city, string).
29     allProperties(lastname, city, string, cost).
30     sumOfCost(lastname, city, integer).
31     costCar(lastname, city, cost).
32     costBuilding(lastname, city, cost).
33     costTerritory(lastname, city, cost).
34     costBoat(lastname, city, cost).
35
36 clauses
37     telephone("Ivanov", "123456",
38               address("Moscow", "Pyshkinskaya", 12, 13)).
39     telephone("Ivanova", "123421",
40               address("Moscow", "Pyshkinskaya", 12, 13)).
41     telephone("Ivanov", "654321",
42               address("Saint-Petersburg", "Sadovaya", 10, 127)).
43     telephone("Ivanov", "222444",
44               address("Saint-Petersburg", "Sadovaya", 10, 127)).
45     telephone("Petrov", "554322",

```

```

46         address("Moscow", "Baymanskaya", 7, 53)).
47 telephone("Petrov", "223544",
48         address("Moscow", "Baymanskaya", 7, 53)).
49
50 deposit("Ivanov", "Tinkoff", "111222333444", 40000, "Moscow").
51 deposit("Ivanova", "Tinkoff", "117227337447", 20000, "Moscow").
52 deposit("Ivanov", "Sberbank", "444333222111", 100000, "Moscow").
53 deposit("Ivanov", "Tinkoff", "123456789000", 150000, "Saint-Petersburg").
54 deposit("Petrov", "Alpha-bank", "222333111444", 10000, "Moscow").
55 deposit("Petrov", "Sberbank", "123123321321", 120000, "Moscow").
56
57 owner("Ivanov", "Moscow", car(3000000, "Mercedes", "Black")).
58 owner("Ivanova", "Moscow", car(3000000, "Mercedes", "Black")).
59 owner("Ivanov", "Saint-Petersburg",
60     car(6000000, "Lamborghini", "Yellow")).
61 owner("Petrov", "Moscow", car(2000000, "Audi", "Red")).
62
63 owner("Ivanov", "Moscow", building(15000000, 150, "Moscow")).
64 owner("Ivanov", "Saint-Petersburg",
65     building(12000000, 90, "Saint-Petersburg")).
66 owner("Petrov", "Moscow", building(8000000, 50, "Moscow")).
67
68 owner("Ivanova", "Moscow", territory(20000000, 500, "Shakhty")).
69 owner("Ivanov", "Saint-Petersburg",
70     territory(25000000, 700, "Beloostrov")).
71
72 owner("Ivanov", "Saint-Petersburg",
73     boat(15000000, "Yacht", "Mangusta", "White")).
74
75 allProperties(Lastname, City, Property, Cost) :-
76     owner(Lastname, City, car(Cost, _, _)), Property = "Car".
77 allProperties(Lastname, City, Property, Cost) :-
78     owner(Lastname, City, building(Cost, _, _)), Property = "Building".
79 allProperties(Lastname, City, Property, Cost) :-
80     owner(Lastname, City, territory(Cost, _, _)), Property = "Territory".
81 allProperties(Lastname, City, Property, Cost) :-
82     owner(Lastname, City, boat(Cost, _, _, _)), Property = "Boat".
83
84 allProperties(Lastname, City, Property) :-
85     allProperties(Lastname, City, Property, _).
86
87 costCar(Lastname, City, Cost) :-

```

```

88         owner(Lastname, City, car(Cost, _, _)),!.
89     costCar(_, _, 0).
90
91     costBuilding(Lastname, City, Cost) :-
92         owner(Lastname, City, building(Cost, _, _)),!.
93     costBuilding(_, _, 0).
94
95     costTerritory(Lastname, City, Cost) :-
96         owner(Lastname, City, territory(Cost, _, _)),!.
97     costTerritory(_, _, 0).
98
99     costBoat(Lastname, City, Cost) :-
100         owner(Lastname, City, boat(Cost, _, _, _)),!.
101     costBoat(_, _, 0).
102
103     sumOfCost(Lastname, City, Sum) :-
104         costCar(Lastname, City, CostCar),
105         costBuilding(Lastname, City, CostBuilding),
106         costTerritory(Lastname, City, CostTerritory),
107         costBoat(Lastname, City, CostBoat),
108         Sum = CostCar + CostBuilding + CostTerritory + CostBoat.
109
110 goal
111     allProperties("Ivanov", "Moscow", Prop, Cost);
112     sumOfCost("Ivanov", "Moscow", Sum).

```

## ФОРМИРОВАНИЕ ОТВЕТА

Для 2-го пункт и одной фамилии составить таблицу, отражающую конкретный порядок работы системы, с объяснениями порядка работы и особенностей использования доменов (указать конкретные T1 и T2 и полную подстановку на каждом шаге)

Таблица 1: allProperties("Ivanov", "Moscow", Prop, Cost).

№ ша- га	Сравниваемые термы: результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат
1	Подстановка: Lastname = "Ivanov", City = "Moscow", Prop = Property, Cost = Cost allProperties("Ivanov", "Moscow", Prop, Cost). allProperties(Lastname, City, Prop, Cost).	Прямой ход
2	Сравнение: "Ivanov" и "Ivanov", "Moscow" и "Moscow" owner(Lastname, City, car(_, _, _)) owner("Ivanov", "Moscow", car(3000000, "Mercedes", "Black"))	Прямой ход
3	Подстановка: Cost = 3000000	Прямой ход
4	Подстановка: Property = "Car"	Прямой ход
5	<b>Результат</b> Cost = 3000000, Property = "Car"	Обратный ход
6	Сравнение: "Ivanov" и "Ivanova" owner(Lastname, City, car(_, _, _)) owner("Ivanova", "Moscow", car(3000000, "Mercedes", "Black")).	Прямой ход
7	Сравнение: "Ivanov" и "Ivanov", "Moscow" и "Saint-Petersburg" owner(Lastname, City, car(_, _, _)) owner("Ivanov", "Saint-Petersburg", car(6000000, "Lamborghini", "Yellow"))	Прямой ход
8	Сравнение: "Ivanov" и "Petrov" owner(Lastname, City, car(_, _, _)) owner("Petrov", "Moscow", car(2000000, "Audi", "Red"))	Обратный ход
9	Подстановка: Lastname = "Ivanov", City = "Moscow", Prop = Property, Cost = Cost	Прямой ход

	allProperties("Ivanov", "Moscow", Prop, Cost). allProperties(Lastname, City, Prop, Cost).	
10	Сравнение: "Ivanov" и "Ivanov", "Moscow" и "Moscow" owner(Lastname, City, building(_, _, _)) owner("Ivanov", "Moscow", building(15000000, 150, "Moscow")).	Прямой ход
11	Подстановка: Cost = 15000000	Прямой ход
12	Подстановка: Property = "Building"	
13	<b>Результат</b> Cost = 15000000, Property = "Building"	Обратный ход
14	Сравнение: "Ivanov" и "Ivanov", "Moscow" и "Saint-Petersburg" owner(Lastname, City, building(_, _, _)) owner("Ivanov", "Saint-Petersburg", building(12000000, 90, "Saint-Petersburg")).	Прямой ход
15	Сравнение: "Ivanov" и "Petrov" owner(Lastname, City, building(_, _, _)) owner("Petrov", "Moscow", building(8000000, 50, "Moscow")).	Обратный ход
16	Подстановка: Lastname = "Ivanov", City = "Moscow", Prop = Property, Cost = Cost allProperties("Ivanov", "Moscow", Prop, Cost). allProperties(Lastname, City, Prop, Cost).	Прямой ход
17	Сравнение: "Ivanov" и "Ivanova" owner(Lastname, City, territory(_, _, _)) owner("Ivanova", "Moscow", territory(20000000, 500, "Shakhty")).	Прямой ход
18	Сравнение: "Ivanov" и "Ivanov", "Moscow" и "Saint-Petersburg" owner(Lastname, City, territory(_, _, _)) owner("Ivanov", "Saint-Petersburg", territory(25000000, 700, "Beloostrov")).	Обратный ход
19	Подстановка: Lastname = "Ivanov", City = "Moscow", Prop = Property, Cost = Cost allProperties("Ivanov", "Moscow", Prop, Cost). allProperties(Lastname, City, Prop, Cost).	Прямой ход
20	Сравнение: "Ivanov" и "Ivanov", "Moscow" и "Saint-Petersburg"	Обратный ход

<pre>owner(Lastname, City, boat(_, _, _, _)) owner("Ivanov",          "Saint-Petersburg", boat(15000000,    "Yacht",    "Mangusta", "White")).</pre>	
--	--



## ВОПРОСЫ

1. **В каком фрагменте программы сформулировано знание? Это знание о чем на формальном уровне?**

Знания о предметной области выражаются на языке Пролог в виде предложений, называемых утверждениями (clauses). Каждое утверждение заканчивается точкой и описывает какое-либо отношение, свойство, объект или закономерность. Структура утверждения проста и имеет одну из форм:

- $\langle \text{заголовок} \rangle$ . – факт
- $\langle \text{заголовок} \rangle :- \langle \text{тело} \rangle$ . – правило, где заголовок является предикатом и полностью характеризует описываемое отношение.

2. **Что содержит тело правила?**

Тело правила содержит условие истинности для этого правила.

3. **Что дает использование переменных при формулировании знаний? В чем отличие формулировки знания с помощью термов с одинаковой арностью при использовании одной переменной и при использовании нескольких переменных?**

Множество фактов образуют простейшую программу Пролога. Но атомарный предикат факта может содержать переменные в качестве аргументов или неосновные термы. В этом случае по умолчанию считается, что добавлен квантор всеобщности  $\forall$  с переменными предиката. Такие факты называются универсальными: они истинны для любых значений переменных. Например,  $\text{любит}(X, \text{яблоко}) \leftarrow$  означает, что любой объект программы “любит яблоко”. Универсальные факты сокращают запись программы.

4. **С каким квантором переменные входят в правило, в каких пределах переменная уникальна?**

На все переменные в имени предиката наложен квантор всеобщности  $\forall$ , на переменные в теле предиката, которые отсутствуют в имени, наложен квантор существования  $\exists$ .

Областью действия переменной в Prolog является одно предложение. В разных предложениях может использоваться одно имя переменной для обозначения

ния разных объектов. Исключением является анонимная переменная. Каждая анонимная переменная – это отдельный объект.

**5. Какова семантика (смысл) предложений раздела DOMAINS? Когда, где и с какой целью используется это описание?**

Предложения раздела DOMAINS описывают новые типы данных, которые используются в разделах PREDICATES, для описания отношений, в которых можно будет использовать более частные случаи типов данных.

**6. Какова семантика (смысл) предложений раздела PREDICATES? Когда, и где используется это описание? С какой целью?**

Предикаты используются для описания фактов в разделе CLAUSES. Предикаты описывают какие домены будет принимать то или иное правило. Это описание используется для создания договоренности о том, в каком порядке будут идти термы в отношениях.

**7. Унификация каких термов запускается на самом первом шаге работы системы? Каковы назначение и результат использования алгоритма унификации?**

Первым запускается унификация термов вопроса.

Для выполнения логического вывода используется механизм (алгоритм) унификации, встроенный в систему. Унификация – операция, которая позволяет формализовать процесс логического вывода (наряду с правилом резолюции). 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.

**8. В каком случае запускается механизм отката?**

В том месте программы, где возможен выбор нескольких вариантов, Пролог сохраняет в специальный стек точку возврата для последующего возвращения в эту позицию. Точка возврата содержит информацию, необходимую

для возобновления процедуры при откате. Выбирается один из возможных вариантов, после чего продолжается выполнение программы.

Во всех точках программы, где существуют альтернативы, в стек заносятся указатели. Если впоследствии окажется, что выбранный вариант не приводит к успеху, то осуществляется откат к последней из имеющихся в стеке точек программы, где был выбран один из альтернативных вариантов. Выбирается очередной вариант, программа продолжает свою работу. Если все варианты в точке уже были использованы, то регистрируется неудачное завершение и осуществляется переход на предыдущую точку возврата, если такая есть. При откате все связанные переменные, которые были означены после этой точки, опять освобождаются.