



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 № 15

Дисциплина Функциональное и логическое программирование

Тема Работа программы на Prolog

Студент Куприй А. А.

Группа ИУ7-63Б

Оценка (баллы) _____

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

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:

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.

Вид 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.

Для 2-го пункта 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).

Вопросы:

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

Знания сформулированы в clauses (факты и правила). Это знания о предметной области.

Что содержит тело правила?

В заголовке правила находится знание о предметной области, а в теле содержится условия истинности этого знания.

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

Связанная с каким-то значением переменная, в рамках одного предложения, может быть использована в других местах.

Чем больше переменных содержит формулировка правила, тем более общим будет являться терм.

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

Переменные входят в правило с квантором всеобщности. Именованная переменная уникальна в рамках предложения, в котором она используется.

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

DOMAINS — раздел описания доменов. Этот раздел используется для описания используемых структур данных.

Какова семантика (смысл) предложений раздела PREDICATES? Когда, и где используется это описание?

PREDICATES — раздел описания предикатов. Это описание используется для проверки корректности «типов» знаний.

Унификация каких термов запускается на самом первом шаге работы системы?

Вопроса и первого терма в clauses.

Каковы назначение и результат использования алгоритма унификации?

Унификация – попытка сопоставить два терма. Результат: успех/неудача.

В каком случае запускается механизм отката?

Механизм отката запустится в случае неудачи алгоритма унификации.

Текст программы

domains

firstname, lastname = symbol.

number = integer.

city, street = symbol.

house, apartment = integer.

label, color, gcolor = symbol.

price = integer.

bank = symbol.

account, amount = integer.

address = address(city, street, house, apartment)

ownership = automobile(label, price, color);

arrea(label, price, gcolor);

boat(label, price);

house(label, price).

person = person(firstname, lastname).

predicates

nondeterm depositor (person, bank, account, amount).

nondeterm phonebook (person, number, address).

nondeterm type(symbol, ownership).

nondeterm price(integer, ownership).

nondeterm owns(person, ownership).

nondeterm owns(person, symbol, integer).

nondeterm count_prices(person, integer).

clauses

phonebook(person(fname1, lname1), 22211100, address(moscow, lenina, 14, 128)).

phonebook(person(fname1, lname1), 22211100, address(moscow, baumanskaya, 1, 10)).

phonebook(person(fname1, lname1), 88877766, address(sp, lenina, 14, 128)).

phonebook(person(fname2, lname2), 44433322, address(moscow, ahmatovoy, 12, 153)).


```
owns(Person, Name3, Price3),
Name<>Name3,
Name1<>Name3,
Name2<>Name3,
Sum = Price+Price1+Price2+Price3, !.
```

```
count_prices(Person, Sum) :- owns(Person, Name, Price),
                               owns(Person, Name1, Price1),
                               Name<>Name1,
                               owns(Person, Name2, Price2),
                               Name1<>Name2,
                               Name <>Name2,
                               Sum = Price+Price1+Price2, !.
```

```
count_prices(Person, Sum) :- owns(Person, Name, Price),
                               owns(Person, Name1, Price1),
                               Name<>Name1,
                               Sum = Price+Price1, !.
```

```
count_prices(Person, Sum) :- owns(Person, _, Sum), !.
```

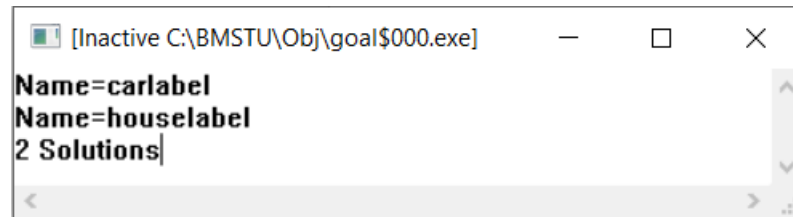
goal

```
owns(person(fname2, lname2), Name, _).
%owns(person(fname1, lname1), Name, Price).
%count_prices(person(fname1, lname1), Sum).
```


Результаты работы программы:

goal

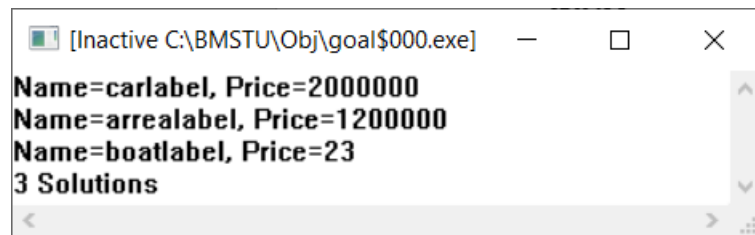
owns(person(fname2, lname2), Name, _).



```
[Inactive C:\BMSTU\Obj\goal$000.exe]
Name=carlabel
Name=housetlabel
2 Solutions
```

goal

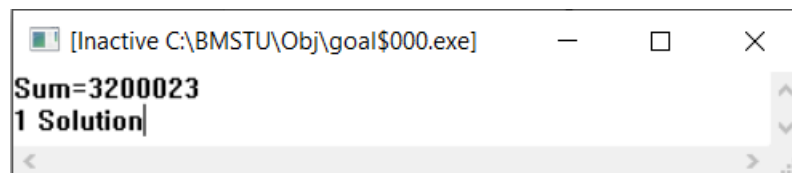
owns(person(fname1, lname1), Name, Price).



```
[Inactive C:\BMSTU\Obj\goal$000.exe]
Name=carlabel, Price=2000000
Name=arrealabel, Price=1200000
Name=boatlabel, Price=23
3 Solutions
```

goal

count_prices(person(fname1, lname1), Sum).



```
[Inactive C:\BMSTU\Obj\goal$000.exe]
Sum=3200023
1 Solution
```

Описание порядка поиска объектов

Цель: owns(person(fname2, lname2), Name, Price).

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
0	-попытка унификации: $T1=T2 \dots$ -результат: Успех и подстановка, или, нет	Комментарий, вывод
1	<ul style="list-style-type: none">Попытка унификации: owns(person(fname2, lname2), Name, Price), phonebook(person(fname1, lname1),...).Результат: неудача, главные функторы не совпадают, термы не унифицируемы.	Прямой ход, переход к следующему предложению.
2	<ul style="list-style-type: none">Попытка унификации: owns(person(fname2, lname2), Name, Price), phonebook(person(fname1, lname1),...).Результат: неудача, главные функторы не совпадают, термы не унифицируемы.	Прямой ход, переход к следующему предложению.
3	<ul style="list-style-type: none">Попытка унификации: owns(person(fname2, lname2), Name, Price), phonebook(person(fname1, lname1),...).Результат: неудача, главные функторы не совпадают, термы не унифицируемы.	Прямой ход, переход к следующему предложению.
4	<ul style="list-style-type: none">Попытка унификации: owns(person(fname2, lname2), Name, Price), phonebook(person(fname2, lname2),...).Результат: неудача, главные функторы не совпадают, термы не унифицируемы.	Прямой ход, переход к следующему предложению.
5	<ul style="list-style-type: none">Попытка унификации: owns(person(fname2, lname2), Name, Price), phonebook(person(fname3, lname3),...).Результат: неудача, главные функторы не совпадают, термы не унифицируемы.	Прямой ход, переход к следующему предложению.
6	<ul style="list-style-type: none">Попытка унификации: owns(person(fname2, lname2), Name, Price), phonebook(person(fname3, lname3),...).Результат: неудача, главные функторы не совпадают, термы не унифицируемы.	Прямой ход, переход к следующему предложению.
7	<ul style="list-style-type: none">Попытка унификации: owns(person(fname2, lname2), Name, Price), depositor(person(fname1, lname1),...).Результат: неудача, главные функторы не	Прямой ход, переход к следующему предложению.

	совпадают, термы не унифицируемы.	
8	<ul style="list-style-type: none"> Попытка унификации: owns(person(fname2, lname2), Name, Price), depositor(person(fname1, lname1),...). Результат: неудача, главные функторы не совпадают, термы не унифицируемы. 	Прямой ход, переход к следующему предложению.
9	<ul style="list-style-type: none"> Попытка унификации: owns(person(fname2, lname2), Name, Price), depositor(person(fname2, lname2),...). Результат: неудача, главные функторы не совпадают, термы не унифицируемы. 	Прямой ход, переход к следующему предложению.
10	<ul style="list-style-type: none"> Попытка унификации: owns(person(fname2, lname2), Name, Price), depositor(person(fname3, lname3),...). Результат: неудача, главные функторы не совпадают, термы не унифицируемы. 	Прямой ход, переход к следующему предложению.
11	<ul style="list-style-type: none"> Попытка унификации: owns(person(fname2, lname2), Name, Price), type(Name, automobile(CarName, _, _)). Результат: неудача, главные функторы не совпадают, термы не унифицируемы. 	Прямой ход, переход к следующему предложению
...	Аналогично выполняется программа со всеми предложениями type(...).	Прямой ход, переход к следующему предложению
18	<ul style="list-style-type: none"> Попытка унификации: owns(person(fname2, lname2), Name, Price), owns(person(fname1, lname1), automobile (car, 2000000, black)). Результат: неудача, несовпадающая арифность. 	Прямой ход, переход к следующему предложению
...	Аналогично выполняется программа со всеми термами owns(person(...), ...) с отличной от вопроса арифностью.	Прямой ход, переход к следующему предложению
23	<ul style="list-style-type: none"> Попытка унификации: owns(person(fname2, lname2), Name, Price), owns(Person, Name, Price). Результат: успех Подстановка: {Person= person(fname2, lname2), Name=Name, Price=Price}. 	Прямой ход, переход в тело правила.
...	<ul style="list-style-type: none"> Попытка унификации: owns(person(fname2, lname2), Ownership), и первых 21 терма Результат: неудача 	Прямой ход, переход к следующему предложению

45	<ul style="list-style-type: none"> Попытка унификации: owns(person(fname2, lname2), Ownership), owns(person(fname2, lname2), automobile (car, 2300000, red)). Результат: успех Подстановка { person(fname2, lname2)= person(fname2, lname2), Ownership=automobile (car, 2300000, red)} 	Прямой ход, переход в к следующей цели правила.
...	<ul style="list-style-type: none"> Попытка унификации: type(Name, automobile (car, 2300000, red)), и первых 10 термов Результат: неудача 	Прямой ход, переход к следующему предложению
56	<ul style="list-style-type: none"> Попытка унификации: type(Name, automobile (car, 2300000, red)), type(Name, automobile (CarName, _ , _)), Результат: успех Подстановка: { Name=Name, automobile (car, 2300000, red) = automobile (CarName, _ , _) } 	Прямой ход, переход в тело правила
57	<ul style="list-style-type: none"> Попытка унификации: Name=car Результат: успех Подстановка: {CarName=car} 	Прямой ход, выполнение предиката отсечения. Выход из правила, переход к следующей подцели.
58	<ul style="list-style-type: none"> Попытка унификации: price(Price, automobile (car, 2300000, red)), и первых 14 термов Результат: неудача 	Прямой ход, переход к следующему предложению
...	<ul style="list-style-type: none"> Попытка унификации. 	Прямой ход, переход к следующему предложению
73	<ul style="list-style-type: none"> Попытка унификации: price(Price, automobile (car, 2300000, red)), price(Price, automobile (_ , Price , _)), Результат: успех Подстановка: { Price=Price, automobile (car, 2300000, red) = automobile (_ , Price , _)} 	Прямой ход, переход в тело правила
74	<ul style="list-style-type: none"> Попытка унификации: Price=2300000 Результат: успех Подстановка: {Price=2300000} 	Прямой ход, выполнение предиката отсечения. Выход из правила. Name=car, Price=2300000. Откат.

...	<ul style="list-style-type: none"> Попытка унификации: price(Price, automobile (car, 2300000, red)), и price(Price, arrea(_, ArreaPrice, _)) price(Price, boat(_, TransportPrice)) price(Price, house(_, BPrice)) Результат: неудача 	Прямой ход переход к следующим предложениям.
...	<ul style="list-style-type: none"> Попытка унификации: price(Price, automobile (car, 2300000, red)), термов owns(...). Результат: неудача, главные функторы не совпадают, термы не унифицируемы. 	Конец БЗ. Откат
...	<ul style="list-style-type: none"> Попытка унификации: type(Name, automobile(car, 2300000, red)) и type(Name, arrea(ArreaName, _, _)) type(Name, boat(BoatName, _)) type(Name, house(BName, _)) Результат: неудача 	Прямой ход переход к следующим предложениям.
...	<ul style="list-style-type: none"> Попытка унификации: type(Name, automobile (car, 2300000, red)) со всеми термами до конца БЗ. Результат: неудача 	Конец БЗ. Откат