|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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** \_Структура программы на Prolog и ее реализация\_  **Студент** \_Ильясов И. М.\_  **Группа** \_ИУ7-63Б\_  **Оценка (баллы)** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **Преподаватель** \_Толпинская Н. Б., Строганов Ю. В.\_ |  |

Москва, 2020 г.

**Цель работы** –изучить структуру, особенности и принципы оформления программы, и способ 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 лабораторной работы**

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 пункт и **одной** фамилии **составить таблицу**, отражающую конкретный порядок работы системы, с объяснениями порядка работы и особенностей использования доменов (указать конкретные Т1 и Т2 и полную подстановку на каждом шаге).

**Текст программы**

domains

surname, phone, city, street = string.

house, flat = integer.

address = address(city, street, house, flat).

mark, color = string.

cost = integer.

bank\_name, bank\_cardnumber = string.

bank\_sum = integer.

area = integer.

property = car(mark, cost, color);

building(area, cost, city);

territory(area, cost, city);

boat(mark, cost, color).

predicates

abonement(surname, phone, address).

deposit(surname, bank\_name, bank\_cardnumber, bank\_sum, city).

own(surname, property, city).

findAllProperties(surname, string, cost, city).

findAllProperties(surname, string, city).

findCostCar(surname, cost, city).

findCostBuilding(surname, cost, city).

findCostTerritory(surname, cost, city).

findCostBoat(surname, cost, city).

findSumCost(surname, integer, city).

clauses

abonement("Gorbunov", "89251472838", address("Korolev", "Glavnaya", 55, 122)).

abonement("Ilyasov", "89969503880", address("Moscow", "Severnaya", 12, 75)).

abonement("Sidenko", "89691929395", address("Moscow", "Semenovskaya", 25, 53)).

abonement("Stepanov", "185818582839", address("Korolev", "Krilatskaya", 12, 155)).

abonement("Gorbunov", "165615253616", address("Saint-Petersburg", "Nevskaya", 51, 122)).

deposit("Gorbunov", "Sberbank", "123456789", 30000, "Korolev").

deposit("Ilyasov", "Sberbank", "987654321", 20000, "Moscow").

deposit("Sidenko", "Tinkoff", "135798642", 60000, "Moscow").

deposit("Stepanov", "Alfa", "156273727", 20000, "Korolev").

deposit("Gorbunov", "Sberbank", "563281726", 100000, "Saint-Petersburg").

own("Gorbunov", car("Audi", 800000, "White"), "Korolev").

own("Ilyasov", car("Mitsubishi", 600000, "Red"), "Moscow").

own("Sidenko", car("Tesla", 4000000, "Black"), "Moscow").

own("Stepanov", car("Mercedes", 2000000, "Red"), "Korolev").

own("Ilyasov", building(150, 12000000, "Moscow"), "Moscow").

own("Sidenko", building(200, 20000000, "Moscow"), "Moscow").

own("Gorbunov", building(90, 7000000, "Saint-Petersburg"), "Saint-Petersburg").

own("Gorbunov", territory(800, 8000000, "Korolev"), "Korolev").

own("Stepanov", territory(700, 7000000, "Korolev"), "Korolev").

own("Gorbunov", boat("Azimut", 20000000, "White"), "Saint-Petersburg").

findAllProperties(Surname, Property, Cost, City) :- own(Surname, car(\_, Cost, \_), City), Property = "Car".

findAllProperties(Surname, Property, Cost, City) :- own(Surname, building(\_, Cost, \_), City), Property = "Building".

findAllProperties(Surname, Property, Cost, City) :- own(Surname, territory(\_, Cost, \_), City), Property = "Territory".

findAllProperties(Surname, Property, Cost, City) :- own(Surname, boat(\_, Cost, \_), City), Property = "Boat".

findAllProperties(Surname, Property, City) :- findAllProperties(Surname, Property, \_, City).

findCostCar(\_, 0, \_).

findCostCar(Surname, Cost, City) :- own(Surname, car(\_, Cost, \_), City),!.

findCostBuilding(\_, 0, \_).

findCostBuilding(Surname, Cost, City) :- own(Surname, building(\_, Cost, \_), City),!.

findCostTerritory(\_, 0, \_).

findCostTerritory(Surname, Cost, City) :- own(Surname, territory(\_, Cost, \_), City),!.

findCostBoat(\_, 0, \_).

findCostBoat(Surname, Cost, City) :- own(Surname, boat(\_, Cost, \_), City),!.

findSumCost(Surname, Sum, City) :- findCostCar(Surname, CostCar, City),

findCostBuilding(Surname, CostBuilding, City),

findCostTerritory(Surname, CostTerritory, City),

findCostBoat(Surname, CostBoat, City),

Sum = CostCar + CostBuilding + CostTerritory + CostBoat.

goal

% Task 1.

findAllProperties("Sidenko", Property, "Moscow").

% Task 2.

%findAllProperties("Stepanov", Property, Cost, "Korolev").

% Task 3.

%findSumCost("Ilyasov", Sum, "Moscow").

**Примеры работы программы**

На рисунке 1 приведен ответ на вопрос об информации (тип собственности) о всей собственности Сиденко.

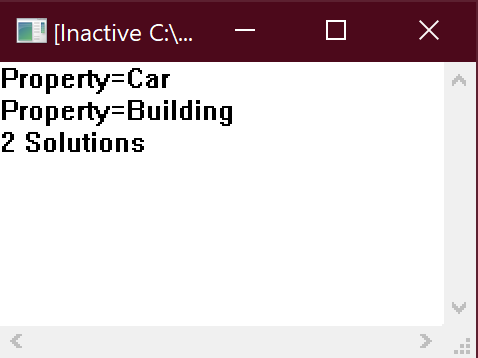


Рисунок 1. Ответ на вопрос об информации (тип собственности) о всей собственности Сиденко.

На рисунке 2 приведен ответ на вопрос об информации (тип собственности, стоимость) о всей собственности Степанова.

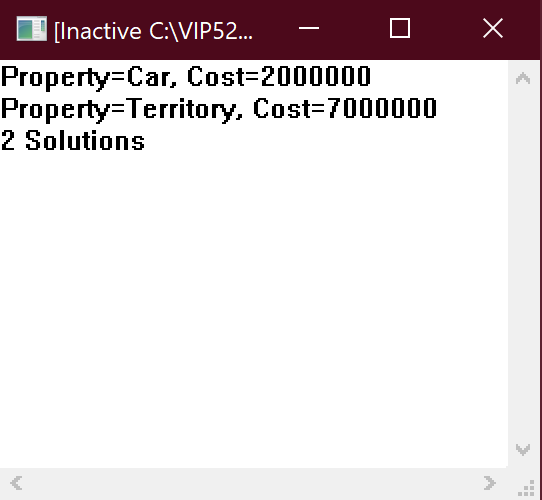


Рисунок 2. Ответ на вопрос об информации (тип собственности, стоимость) о всей собственности Степанова.

На рисунке 3 приведен ответ на вопрос об информации (сумма стоимости всей собственности) о всей собственности Ильясова.

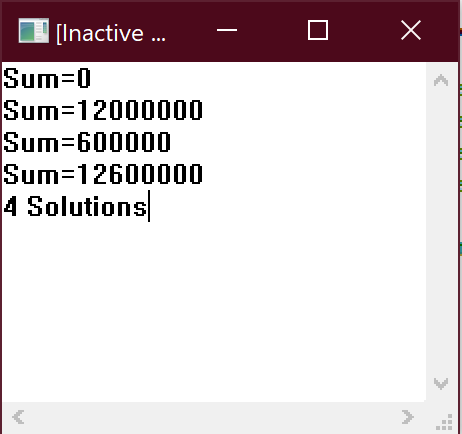


Рисунок 3. Ответ на вопрос об информации (сумма стоимости всей собственности) о всей собственности Ильясова.

**Задание с таблицей**

findAllProperties("Stepanov", Property, Cost, "Korolev").

|  |  |  |
| --- | --- | --- |
| № шага | Сравниваемые термы; результат; подстановка, если есть | Дальнейшие действия: прямой ход или откат (к чему приводит?) |
| 1 | Производится сравнение findAllProperties("Stepanov", Property, Cost, "Korolev") и abonement("Gorbunov", "89251472838", address("Korolev", "Glavnaya", 55, 122)).  Результат – неудача. | Прямой ход к findAllProperties("Stepanov", Property, Cost, "Korolev") |
|  | … | … |
| 2 | Производится сравнение findAllProperties("Stepanov", Property, Cost, "Korolev") и findAllProperties(Surname, Property, Cost, City). Surname конкретизируется с “Stepanov”, Property – с “Car”, City – “Korolev”.  Результат – findAllProperties("Stepanov", Property, Cost, "Korolev"). | Прямой ход, производится переход к унификации own(Surname, car(\_, Cost, \_), City). Surname = “Stepanov”, Color = “White”, City = “Korolev”, Property = “Car”. |
|  | … | … |
| 3 | Производится сравнение own(“Stepanov”, car(\_, Cost, \_), “Korolev”) и abonement("Gorbunov", "89251472838", address("Korolev", "Glavnaya", 55, 122)).  Результат – неудача. | Откат к own(Surname, car(\_, Cost, \_), City), проверка следующей строки |
|  | … | … |
| 4 | Производится сравнение own(“Stepanov”, car(\_, Cost, \_), “Korolev”) и own("Stepanov", car("Mercedes", 2000000, "Red"), “Korolev”). Так как все совпало и истинность подтверждена, выводим “Car” и стоимость “Cost” = 2000000. | Откат к findAllProperties("Stepanov", Property, Cost, "Korolev"), проверка следующей строки |
|  | … | … |
| 5 | Производится сравнение findAllProperties("Stepanov", Property, Cost, "Korolev") и findAllProperties(Surname, Property, Cost, City). Surname конкретизируется с “Stepanov”, Property – с “Building”, City – “Korolev”.  Результат – findAllProperties("Stepanov", Property, Cost, "Korolev"). | Прямой ход, производится переход к унификации own(Surname, building(\_, Cost, \_), City). Surname = “Stepanov”, Color = “White”, City = “Korolev”, Property = “Building”. |
|  | … | … |
| 6 | Производится сравнение own(“Stepanov”, building(\_, Cost, \_), “Korolev”) и abonement("Gorbunov", "89251472838", address("Korolev", "Glavnaya", 55, 122)).  Результат – неудача. | Откат к own(Surname, building(\_, Cost, \_), City), проверка следующей строки |
|  | … | … |
| 7 | Производится сравнение own(“Stepanov”, building(\_, Cost, \_), “Korolev”) и own("Stepanov", car("Mercedes", 2000000, "Red"), “Korolev”). Результат - неудача | Откат к findAllProperties("Stepanov", Property, Cost, "Korolev"), проверка следующей строки |
|  | … | … |
| 8 | Производится сравнение findAllProperties("Stepanov", Property, Cost, "Korolev") и findAllProperties(Surname, Property, Cost, City). Surname конкретизируется с “Stepanov”, Property – с “Territory”, City – “Korolev”.  Результат – findAllProperties("Stepanov", Property, Cost, "Korolev"). | Прямой ход, производится переход к унификации own(Surname, territory(\_, Cost, \_), City). Surname = “Stepanov”, Color = “White”, City = “Korolev”, Property = “Territory”. |
|  | … | … |
| 9 | Производится сравнение own(“Stepanov”, territory(\_, Cost, \_), “Korolev”) и abonement("Gorbunov", "89251472838", address("Korolev", "Glavnaya", 55, 122)).  Результат – неудача. | Откат к own(Surname, territory(\_, Cost, \_), City), проверка следующей строки |
|  | … | … |
| 10 | Производится сравнение own(“Stepanov”, territory(\_, Cost, \_), “Korolev”) и own("Stepanov", territory(700, 7000000, "Korolev"), "Korolev"). Так как все совпало и истинность подтверждена, выводим “Territory” и стоимость “Cost” = 7000000. | Откат к findAllProperties("Stepanov", Property, Cost, "Korolev"), проверка следующей строки |
|  | … | … |
|  | Результат – 2 решения | |

**Ответы на вопросы**

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

Правила – предложение вида , то, что находится слева от знака является заголовком правила, то, что справа – телом правила. Факт (знание) – частный случай правила, у него нет тела. То есть знание сформулировано в заголовке правила.

1. **Что содержит тело правила?**

Тело правила содержит условие истинности заголовка правила.

1. **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?**

Использование переменных в формулировании знаний позволяют уточнять значения и переносить их в пространстве и времени. Формулировка знаний с использованием переменных носит более общий характер по отношению к знанию, состоящему только лишь из констант. Например, использование знаний с одинаковой арностью при использовании одной переменной носит менее общий характер по отношению знания с использованием нескольких переменных.

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

Переменные входят в правило с квантором всеобщности (для любой). Именованные переменные уникальны в пределах одного предложения, анонимные уникальны все.

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

Предложения в разделе DOMAINS используются для объявления используемых доменов, не являющимися стандартными доменами в Prolog. Раздел доменов используется для описания структур (вариантных доменов).

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

В разделе PREDICATES описываются предикаты, их арность (местность) и домены (типы и природа аргументов). С помощью описанных предикатов, можно создавать предложения в базе знаний. Предикаты используются для представления, как фактов, так и правил.

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

На первом шаге работы происходит унификация вопроса и первого предложения базы знаний. Алгоритм унификации необходим для попытки "увидеть одинаковость" – сопоставимость двух термов, может завершаться успехом или тупиковой ситуацией. Результат унификации – ответ «да» или «нет».

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

Механизм отката запускается в 2 случаях:

1. Если алгоритм попал в тупиковую ситуацию.

2. Если резольвента не пуста и решение найдено, но в базе знание остались не отмеченные предложения.