|  |  |
| --- | --- |
|  | **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**

*к лабораторной работе №15*

*По курсу: «Функциональное и логическое программирование»*

Студентка ИУ7-65Б

Оберган Т.М.

Преподаватели

Толпинская Н.Б.

Строганов Ю.В.

*Москва, 2020 г.*

Оглавление

[Задание 3](#_Toc39493907)

[Вопросы 4](#_Toc39493908)

[Листинг 6](#_Toc39493909)

[Таблица 7](#_Toc39493910)

# 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 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)

|  |  |  |
| --- | --- | --- |
| № шага | Сравниваемые термы; результат; подстановка, если есть | Дальнейшие действия: прямой ход или откат (к чему приводит?) |
| 1… | -попытка унификации: Т1=Т2 …  -результат: Успех и подстановка,  или Нет | Комментарий, вывод… |
| 2 | … | … |

При желании, можно усложнить свою базу знаний, введя варианты: строение: (Дом, офис, торговый центр), участок: (садовый, территория под застройку, территория под агро-работы), Водный\_транспорт: варианты названий.

# 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 арностью 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.

# Листинг

domains

lname, phone, city, street = symbol.

building\_n, apartment, amount = **integer**.

model, color, bank, account = symbol.

address = address(city, street, building\_n, apartment).

cost, size = **integer**.

property =

building(cost, size);

sector(cost, size);

ship(cost, color);

car(model, color, cost).

predicates

phone\_book(lname, phone, address).

investors(lname, bank, account, amount).

own(lname, property).

own\_type(lname, symbol, cost).

own\_cost(lname, symbol, cost).

total\_cost(lname, cost).

clauses

phone\_book("Winter", "+123", address("Moscow", "Postal", 1, 2)).

phone\_book("Winter", "+000", address("SPB", "Neva", 2, 4)).

phone\_book("Summer", "+345", address("Sevastopol", "Anotherstrt", 4, 8)).

phone\_book("Autumn", "+333", address("Moscow", "Postal", 8, 16)).

investors("Winter", "RNCB", private, 999999999).

investors("Autumn", "Sberbank", public, 10).

investors("Summer", "Raiffeisen", public, 10000).

own("Winter", building(12000, 40)).

own("Winter", car("SmallOne", black, 150)).

own("Summer", ship(1000, grey)).

% Type of property by lastname

own\_type(Lname, building, Cost) :- own(Lname, building(Cost, \_)).

own\_type(Lname, sector, Cost) :- own(Lname, sector(Cost, \_)).

own\_type(Lname, ship, Cost) :- own(Lname, ship(Cost, \_)).

own\_type(Lname, car, Cost) :- own(Lname, car(\_, \_, Cost)).

% Get cost or 0

own\_cost(Lname, building, Cost) :- own(Lname, building(Cost, \_)), !.

own\_cost(Lname, sector, Cost) :- own(Lname, sector(Cost, \_)), !.

own\_cost(Lname, ship, Cost) :- own(Lname, ship(Cost, \_)), !.

own\_cost(Lname, car, Cost) :- own(Lname, car(\_, \_, Cost)), !.

own\_cost(\_, \_, 0).

% Total cost of all property

% NOTE: owner CAN'T have more than one property of each type

total\_cost(Lname, Total) :-

own\_cost(Lname, building, Cost1),

own\_cost(Lname, sector, Cost2),

own\_cost(Lname, ship, Cost3),

own\_cost(Lname, car, Cost4),

Total = Cost1 + Cost2 + Cost3 + Cost4.

goal

%own("Winter", Property).

% 1. Type of property by lastname

%own\_type("Winter", PropType, \_).

% 2. Type of property and cost by lastname

%own\_type("Winter", PropType, Cost).

%own\_type("Autumn", PropType, Cost).

%3. Total cost of property

total\_cost("Winter", TotalCost).

# Таблица

Вопрос: own\_type("Winter", PropType, Cost).

|  |  |  |
| --- | --- | --- |
| № шага | Сравниваемые термы; результат; подстановка, если есть | Дальнейшие действия: прямой ход или откат |
| 1 | -попытка унификации: own\_type("Winter", PropType, Cost). = phone\_book("Winter", "+123", address("Moscow", "Postal", 1, 2)).  -результат: Нет |  |
| 2 | -попытка унификации: own\_type("Winter", PropType, Cost). = investors("Winter", "RNCB", private, 999999999).  -результат: Нет |  |
| 3 | -попытка унификации:  own\_type("Winter", PropType, Cost). = own("Winter", building(12000, 40)).  -результат: Нет |  |
| 4 | -попытка унификации:  own\_type("Winter", PropType, Cost). = own\_type(Lname, building, Cost)  -результат: Успех и подстановка  Lname = “Winter”  PropType = building |  |
| 5 | -попытка унификации:  own("Winter", building(Cost, \_)). = phone\_book("Winter", "+123", address("Moscow", "Postal", 1, 2)).  -результат: Нет |  |
| 6 | -попытка унификации:  own("Winter", building(Cost, \_)). = investors("Winter", "RNCB", private, 999999999).  -результат: Нет |  |
| 7 | -попытка унификации:  own("Winter", building(Cost, \_)). = own("Winter", building(12000, 40)).  -результат: Успех и подстановка  Cost = 12000 | Резольвента пуста. Решение найдено:  PropType = building  Cost = 12000 |
| 8 | -попытка унификации: own("Winter", building(Cost, \_)). = own("Winter", car("SmallOne", black, 150)).  -результат: Нет |  |
| 9 | -попытка унификации: own("Winter", building(Cost, \_)). = own("Summer", ship(1000, grey)).  -результат: Нет | Откат к 4. |
| 10 | -попытка унификации: own\_type("Winter", PropType, Cost). = own\_type(Lname, sector, Cost)  -результат: Успех и подстановка  Lname = “Winter”  PropType = sector |  |
| 11-15 | -попытка унификации:  own("Winter", sector(Cost, \_)).  С разными знаниями  -результат: Нет | Откат к 10. |
| 16 | -попытка унификации: own\_type("Winter", PropType, Cost). = own\_type(Lname, ship, Cost)  -результат: Успех и подстановка  Lname = “Winter”  PropType = ship |  |
| 17-21 | -попытка унификации:  own("Winter", ship(Cost, \_)).  С разными знаниями  -результат: Нет | Откат к 16. |
| 22 | -попытка унификации:  own\_type("Winter", PropType, Cost). = own\_type(Lname, car, Cost)  -результат: Успех и подстановка  Lname = “Winter”  PropType = car |  |
| 23-25 | -попытка унификации:  own("Winter", car(\_, \_, Cost)).  -результат: Нет |  |
| 26 | -попытка унификации:  own("Winter", car(\_, \_, Cost)). = own("Winter", car("SmallOne", black, 150)).  -результат: Успех и подстановка  Cost = 150 | Резольвента пуста. Решение найдено:  PropType = car  Cost = 150 |
| 27 | -попытка унификации: own("Winter", car(\_, \_, Cost)). = own("Summer", ship(1000, grey)).  -результат: Нет | Откат к 22.  Система завершает работу. |