



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 по лабораторной работе №13 по курсу  
«Функциональное и логическое  
программирование»**

Тема Структура программы на Prolog и ее реализация

Студент Климов И.С.

Группа ИУ7-62Б

Оценка (баллы) \_\_\_\_\_

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

Москва — 2022 г.

## Задание 1

Создать базу знаний «**Собственники**», дополнив (и минимально изменив) 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.

Дополнить базу знаний 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, 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

## Решение

### domains

```
person = string.
number = string.

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

own_name = string.
own_price = unsigned.

car_color = string.

bank, bank_account = string.
bank_sum = unsigned.

car = car(own_name, own_price, car_color).
structure = structure(own_name, own_price).
plot = plot(own_name, own_price).
water = water(own_name, own_price).
own = car(own_name, own_price, car_color); structure(own_name, own_price);
      plot(own_name, own_price); water(own_name, own_price).
```

### predicates

```
person_number(person, number).
person_car(person, car).
person_bank(person, bank, bank_account, bank_sum).
person_address(person, address).

number_car(number, person, own_name, own_price).

person_city_street_bank_number(person, city, street, bank, number).

car_person_city_number_bank(own_name, car_color, person, city, number, bank).

person_structure(person, structure).
person_plot(person, plot).
person_water(person, water).
owner(person, own_name, own_price).

car_price(person, own_price).
structure_price(person, own_price).
plot_price(person, own_price).
water_price(person, own_price).
owner_total_price(person, own_price).
```

## clauses

```
person_number("Klimov", "89124214959").
person_number("Ivanov", "89424412959").
person_number("Klimov", "89414215521").
person_number("Petrov", "89412451252").

person_car("Ivanov", car("Audi", 400000, "Black")).
person_car("Klimov", car("Nissan", 200000, "White")).
person_car("Petrov", car("BMW", 500000, "Green")).

person_bank("Petrov", "Sber", "123456789012345", 2300000).
person_bank("Klimov", "Tinkoff", "1491256789012195", 420000).
person_bank("Klimov", "Alpha", "41455789212241", 10000000).
person_bank("Ivanov", "Sber", "12344212412345", 42100000).

person_address("Klimov", address("Moscow", "Street_1", 1, 12)).
person_address("Ivanov", address("Moscow", "Street_2", 2, 32)).
person_address("Petrov", address("St.Peterburg", "Street_3", 3, 42)).
person_address("Klimov", address("St. Peterburg", "Street_4", 4, 52)).

number_car(Number, Person, Car_model, Car_price) :-
    person_number(Person, Number),
    person_car(Person, car(Car_model, Car_price, _)).

person_city_street_bank_number(Person, City, Street, Bank, Number) :-
    person_address(Person, address(City, Street, _, _)),
    person_bank(Person, Bank, _, _), person_number(Person, Number).

car_person_city_number_bank(Model, Color, Person, City, Number, Bank) :-
    person_car(Person, car(Model, _, Color)),
    person_address(Person, address(City, _, _, _)),
    person_number(Person, Number), person_bank(Person, Bank, _, _).

person_structure("Klimov", structure("Build_1", 1000000)).
person_structure("Petrov", structure("Build_2", 2000000)).
person_structure("Ivanov", structure("Build_3", 300000)).

person_plot("Ivanov", plot("Plot_1", 500000)).
person_plot("Klimov", plot("Plot_2", 100000)).

person_water("Klimov", water("Water_transport_1", 10000000)).
person_water("Petrov", water("Water_transport_2", 15000000)).

car_price(Person, Price) :- person_car(Person, car(_, Price, _)), !.
car_price(_, 0).
```

```

structure_price(Person, Price) :-
    person_structure(Person, structure(_, Price)), !.
structure_price(_, 0).

plot_price(Person, Price) :- person_plot(Person, plot(_, Price)), !.
plot_price(_, 0).

water_price(Person, Price) :- person_water(Person, water(_, Price)), !.
water_price(_, 0).

owner(Person, Own, Price) :- person_car(Person, car(Own, Price, _));
                             person_structure(Person, structure(Own, Price));
                             person_plot(Person, plot(Own, Price));
                             person_water(Person, water(Own, Price)).

owner_total_price(Person, Total) :- car_price(Person, Price_1),
                                    structure_price(Person, Price_2),
                                    plot_price(Person, Price_3),
                                    water_price(Person, Price_4),
                                    Total = Price_1 + Price_2 + Price_3 + Price_4.

goal

% owner("Klimov", Own, _).
% owner("Petrov", Own, Price).
% owner_total_price("Ivanov", Total).

```

## Задание 2

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)

## Решение

| Шаг   | Сравниваемые термы; результат, подстановка, если есть  | Дальнейшие действия: прямой ход или откат  |
|-------|--|--|
| 1     | Состояние резольвенты:<br>owner("Petrov", Own, Price).<br>Попытка унификации:<br>owner("Petrov", Own, Price) =<br>person_number("Klimov", "89124214959").<br>Результат: неудача (несовпадение функторов) | Прямой ход   |
| 2-32  | ...  | ...  |
| 33    | Попытка унификации:<br>owner("Petrov", Own, Price) =<br>owner(Person, Own, Total)<br>Результат: успех<br>Person = "Petrov", Own = Own, Total = Total   | Прямой ход<br>Состояние резольвенты:<br>person_car("Petrov", car(Own, Price, _)) |
| 34    | Попытка унификации:<br>person_car(Person, car(Own, Price, _)) =<br>person_number("Klimov", "89124214959").<br>Результат: неудача (несовпадение функторов)  | Прямой ход   |
| 35-37 | ...  | ...  |
| 38    | Попытка унификации:<br>person_car("Petrov", car(Own, Price, _)) =<br>person_car("Ivanov", car("Audi", 400000, "Black")).<br>Результат: неудача (несовпадение аргумента)                                  | Прямой ход   |
| 39    | ...  | ...  |
| 40    | Попытка унификации:<br>person_car("Petrov", car(Own, Price, _)) =  | Решение:<br>Own = BMW,   |

|        |  |  |
|--------|--|--|
|        | <code>person_car("Petrov", car("BMW", 500000, "Green")).</code><br>Результат: успех<br><code>Own = "BMW", Price = 500000</code>  | <code>Price = 500000</code><br>Откат<br><code>Own = Own,</code><br><code>Price = Price</code>  |
| 41-67  | ...  | ...  |
| 68     | Попытка унификации:<br><code>person_car(Person, car(Own, Price, _)) =</code><br><code>owner_total_price(Person, Total) :- car_price(</code><br><code>Person, Price_1), structure_price(Person,</code><br><code>Price_2), plot_price(Person, Price_3),</code><br><code>water_price(Person, Price_4),</code><br><code>Total = Price_1 + Price_2 + Price_3 + Price_4.</code><br>Результат: неудача (несовпадение функторов) | Откат<br>Новое состояние<br>резольвенты:<br><code>person_structure(</code><br><code>"Petrov", structure(</code><br><code>Own, Price))</code> |
| 69-... | ...  | ...  |