**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 Т1 и Т2 и полную подстановку на каждом шаге)

**Решение**

|  |  |  |
| --- | --- | --- |
| **Шаг** | **Сравниваемые термы; результат, подстановка, если есть** | **Дальнейшие действия: прямой ход или откат** |
| 1 | Состояние резольвенты:  owner("Petrov", Own, Price).  Попытка унификации:  owner("Petrov", Own, Price) =  person\_number("Klimov", "89124214959").  Результат: неудача (несовпадение функторов) | Прямой ход |
| 2-32 | … | … |
| 33 | Попытка унификации:  owner("Petrov", Own, Price) =  owner(Person, Own, Total)  Результат: успех  Person = "Petrov", Own = Own, Total = Total | Прямой ход  Состояние резольвенты: person\_car("Petrov", car(Own, Price, \_)) |
| 34 | Попытка унификации:  person\_car(Person, car(Own, Price, \_)) =  person\_number("Klimov", "89124214959").  Результат: неудача (несовпадение функторов) | Прямой ход |
| 35-37 | … | … |
| 38 | Попытка унификации:  person\_car("Petrov", car(Own, Price, \_)) =  person\_car("Ivanov", car("Audi", 400000, "Black")).  Результат: неудача (несовпадение аргумента) | Прямой ход |
| 39 | … | … |
| 40 | Попытка унификации:  person\_car("Petrov", car(Own, Price, \_)) =  person\_car("Petrov", car("BMW", 500000, "Green")).  Результат: успех  Own = "BMW", Price = 500000 | Решение:  Own = BMW,  Price = 500000  Откат  Own = Own,  Price = Price |
| 41-67 | … | … |
| 68 | Попытка унификации:  person\_car(Person, car(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.  Результат: неудача (несовпадение функторов) | Откат  Новое состояние резольвенты:  person\_structure(  "Petrov", structure( Own, Price)) |
| 69-… | … | … |