



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 № 8
по курсу «Функциональное и логическое программирование»
на тему: «Среда Visual Prolog»

Студент ИУ7-61Б
(Группа)

(Подпись, дата)

Д. В. Шубенина
(И. О. Фамилия)

Преподаватель

(Подпись, дата)

Н. Б. Толпинская
(И. О. Фамилия)

Преподаватель

(Подпись, дата)

Ю. В. Строганов
(И. О. Фамилия)

2024 г.

СОДЕРЖАНИЕ

1	Практическая часть	3
1.1	Задание	3
1.2	Порядок работы системы	7

1 Практическая часть

1.1 Задание

Создать базу знаний «Собственники», дополнив (и минимально изменив) базу знаний, хранящую знания (лаб. 7), знаниями о дополнительной собственности владельца:

- «Телефонный справочник»: Фамилия, №тел, Адрес — структура (Город, Улица, №дома, №кв),
- «Автомобили»: Фамилия_владельца, Марка, Цвет, Стоимость и др.,
- «Вкладчики банков»: Фамилия, Банк, Счет, Сумма и др.

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).

```

1 domains
2     surname,city,street = string
3     brand,carno = string
4     prop_type = symbol
5     bank = string
6
7     houseno,flatno,phoneno = integer
8     price,area = integer
9     account,money = integer
10
11     sex,color = symbol
12
13     address = make_address(city,street,houseno,flatno)
14     property = car(price,brand,color,carno);
15                 building(price,address);
16                 land(price,area);
17                 water_transport(price,color)
18
19 predicates
20     phonebook(surname,phoneno,address)
21     owner(surname,property)
22     bank_depositor(surname,bank,account,money)
23     parent(surname,surname,sex)
24
25     has_property(surname,symbol)
26     property_price(surname,prop_type,price)
27     property_cost(surname,prop_type,price)
28     sum_cost(surname,price)
29
30     grandchild(surname,surname,sex)
31
32 clauses
33     phonebook("Ivanov",100,
34                 make_address("Moscow","Taganskaya",18,34)).
35     phonebook("Ivanov",600,
36                 make_address("Moscow","Taganskaya",18,34)).
37     phonebook("Markov",500,
38                 make_address("Kazan","Lenina",10,100)).
39     phonebook("Prohorov",111,
40                 make_address("SPb","Brodskogo",4,1)).
41     phonebook("Kalinina",444,

```

```

38     make_address("SPb", "Pushkinskaya", 31, 90)).
phonebook("Orlov", 676,
39     make_address("Moscow", "Chertanovksaya", 90, 4)).
phonebook("Vaselinina", 555,
40     make_address("Sochi", "Roz", 3, 10)).
41
42 bank_depositor("Markov", "VTB", 100, 5000).
bank_depositor("Prohorov", "Sber", 90, 100000).
43
44 owner("Ivanov", car(5400000, "BMW", black, "B7770P777")).
45 owner("Ivanov", car(20000000, "Maybach", black, "T7770K777")).
46 owner("Markov", car(670000, "Mitsubishi", gray, "B900BC116")).
47 owner("Prohorov", car(500000, "Lada", crimson, "T007EC178")).
48 owner("Kalinina", car(3500000, "VW", green, "T888BC178")).
49 owner("Orlov", car(5100000, "Lada", black, "P132YC797")).
50 owner("Vaselinina", car(1100000, "Nissan", gray, "K452HH123")).
51 owner("Vaselinina", car(500000, "Lada", crimson, "T643AK123")).
52
53 owner("Ivanov",
54     building(1000000, make_address("Moscow", "Taganskaya", 18, 34))).
owner("Orlov",
55     building(1000000, make_address("Murmansk", "Lenina", 1, 9))).
56
57 owner("Vaselinina", land(300000, 56)).
owner("Ivanov", land(300000, 56)).
58
59 owner("Vaselinina", water_transport(40000, yellow)).
60 owner("Prohorov", water_transport(55000, white)).
61
62 parent("Ivanov", "Markov", m).
63 parent("Orlov", "Ivanov", m).
64 parent("Ivanov", "Vaselinina", f).
65
66 grandchild(Name, Elder, Sex) :-
67     parent(Name, Parent, _),
68     parent(Parent, Elder, Sex).
69
70 % task 1
71 has_property(Surname, PropType) :-
72     property_price(Surname, PropType, _).
73

```

```

74 % task 2
75 property_price(Surname,land,Price) :-
    owner(Surname,land(Price,_)).
76 property_price(Surname,water_transport,Price) :-
    owner(Surname,water_transport(Price,_)).
77 property_price(Surname,car,Price) :-
    owner(Surname,car(Price,_,_,_)).
78 property_price(Surname,building,Price) :-
    owner(Surname,building(Price,_)).
79
80 % task 3
81 property_cost(Surname,"car",Cost) :- owner(Surname,
    car(Cost,_,_,_)), !.
82 property_cost(Surname,"building",Cost) :- owner(Surname,
    building(Cost,_)), !.
83 property_cost(Surname,"land",Cost) :- owner(Surname,
    land(Cost,_)), !.
84 property_cost(Surname,"water_transport",Cost) :-
    owner(Surname, water_transport(Cost,_)), !.
85 property_cost(_,_,0).
86
87 sum_cost(Surname,Cost) :-
88     property_cost(Surname,"building",Cost1),
89     property_cost(Surname,"land",Cost2),
90     property_cost(Surname,"water_transport",Cost3),
91     property_cost(Surname,"car",Cost4),
92     Cost = Cost1 + Cost2 + Cost3 + Cost4.
93 goal
94 %has_property("Ivanov",Type).
95 property_price("Orlov",PropType,Price).
96 %sum_cost("Ivanov",Cost).

```

1.2 Порядок работы системы

Вопрос: `property_price("Orlov", PropType, Price)`.

№ шага	Сравниваемые термы: результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат
1	Сравнение: <code>property_price("Orlov", PropType, Price) == phonebook("Ivanov", 100, make_address("Moscow", "Taganskaya", 18, 34))</code> Унификация: неудача	Прямой ход, переход к следующему предложению
2 –
	Сравнение: <code>property_price("Orlov", PropType, Price) == property_price(Surname, car, Price)</code> Унификация: успех Подстановка: <code>{Surname = "Orlov", PropType = car, Price = Price}</code>	Переход к телу правила. Унификация <code>owner("Orlov", car(Price, _, _, _))</code>
	Сравнение: <code>owner("Orlov", car(Price, _, _, _)) == phonebook("Ivanov", 100, make_address("Moscow", "Taganskaya", 18, 34))</code> Унификация: неудача	Прямой ход, переход к следующему предложению
–
	Сравнение: <code>owner("Orlov", car(Price, _, _, _)) == owner("Orlov", car(5100000, "Lada", black, "P132YC797"))</code> Унификация: успех Подстановка: <code>{Price = 5100000}</code>	Решение найдено, в качестве побочного эффекта получена подстановка <code>{Property = car, Price = 5100000}</code> Откат с целью найти все возможные решения. Реконкретизация переменной Price. Переход к следующему предложению.
	Сравнение: <code>owner("Orlov", car(Price, _, _, _)) == owner("Vaselinina", car(1100000, "Nissan", gray, "K452HH123"))</code> Унификация: неудача	Прямой ход, переход к следующему предложению.
–
		Конец базы знаний — откат. Реконкретизация переменных Surname, Property. Переход к следующему предложению относительно шага N.

	<p>Сравнение: <code>property_price("Orlov", PropType, Price) == property_price(Surname, building, Price)</code></p> <p>Унификация: успех</p> <p>Подстановка: <code>{Surname = "Orlov", PropType = building, Price = Price}</code></p>	<p>Переход к телу правила.</p> <p>Унификация <code>owner("Orlov", building(Price, _))</code></p>
	<p>Сравнение: <code>owner("Orlov", car(Price, _, _, _)) == phonebook("Ivanov", 100, make_address("Moscow", "Taganskaya", 18, 34))</code></p> <p>Унификация: неудача</p>	<p>Прямой ход, переход к следующему предложению</p>
—
	<p>Сравнение: <code>owner("Orlov", building(Price, _)) == owner("Orlov", building(1000000, make_address("Murmansk", "Lenina", 1, 9)))</code>.</p> <p>Унификация: успех</p> <p>Подстановка: <code>{Price = 1000000}</code></p>	<p>Решение найдено, в качестве побочного эффекта получена подстановка <code>{Property = building, Price = 1000000}</code></p> <p>Откат с целью найти все возможные решения. Реконкретизация переменной Price. Переход к следующему предложению.</p>
	<p>Сравнение: <code>owner("Orlov", building(Price, _, _, _)) == owner("Vaselinina", land(300000, 56))</code></p> <p>Унификация: неудача</p>	<p>Прямой ход, переход к следующему предложению</p>
—
		<p>Конец базы знаний — откат. Реконкретизация переменных Surname, Property. Переход к следующему предложению относительно шага N.</p>
	<p>Сравнение: <code>property_price("Orlov", PropType, Price) == property_cost(Surname, car, Cost)</code></p> <p>Унификация: неудача</p>	<p>Прямой ход, переход к следующему предложению</p>
—
		<p>Конец базы знаний. Завершение работы.</p> <p>На вопрос удалось получить ответ «да», поэтому в качестве побочного эффекта возвращено 2 подстановки.</p>