



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 №13

*По предмету: «Функциональное и логическое
программирование»*

Студент: Лаврова А. А.,
Группа: ИУ7-65Б
Преподаватель: Толпинская Н. Б.

Москва, 2020 г.

Практическая часть

Задание

Составить программу, т.е. модель предметной области – базу знаний, объединив в ней информацию – знания:

- **«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.

Владелец может иметь несколько телефонов, автомобилей, вкладов (Факты).

Используя правила, обеспечить возможность поиска:

1. **а)** По № телефона найти: Фамилию, Марку автомобиля, Стоимость автомобиля (может быть несколько),
в) Используя сформированное в пункте **а)** правило, по № телефона найти: только Марку автомобиля (автомобилей может быть несколько),
2. Используя простой, не составной вопрос: по Фамилии (уникальна в городе, но в разных городах есть однофамильцы) и Городу проживания найти: Улицу проживания, Банки, в которых есть вклады и №телефона.

Листинг программы:

```
domains
    surname = symbol.
    telephone = symbol.
    city = symbol.
    street = symbol.
    house = integer.
    address = address(city, street, house).

    model = symbol.
    color = symbol.
    price = integer.

    name_of_bank = symbol.
    account = integer.
    amount = integer.

predicates
    phonebook(surname, telephone, address).
    automobile(surname, model, color, price).
    bank(surname, name_of_bank, account, amount).

    find_by_number(telephone, surname, model, price).
    find_by_surname_city(surname, city, street, name_of_bank, telephone).

clauses
    phonebook("Ivanov", "8111111111", address("Moscow", "Street1", 1)).
    phonebook("Petrov", "8222222222", address("Moscow", "Street2", 2)).
    phonebook("Sidorov", "8333333333", address("Kazan", "Street3", 3)).
    phonebook("Vasechkin", "8444444444", address("Chelyabinsk", "Street4", 4)).
    phonebook("Lolaev", "8555555555", address("Zlatoust", "Street5", 5)).
    phonebook("Lolaev", "8666666666", address("Ekaterinburg", "Street6", 6)).

    automobile("Ivanov", "Kia", "red", 1550000).
    automobile("Ivanov", "Lada", "grey", 1000000).
    automobile("Vasechkin", "Land Rover", "black", 5000000).
```

```
automobile("Sidorov", "Honda", "white", 2000000).
```

```
bank("Lolaev", "HomeCreditBank", 100, 100).  
bank("Ivanov", "Sberbank", 200, 200).  
bank("Vasechkin", "VTB", 300, 300).  
bank("Lolaev", "Pochta Bank", 400, 400).
```

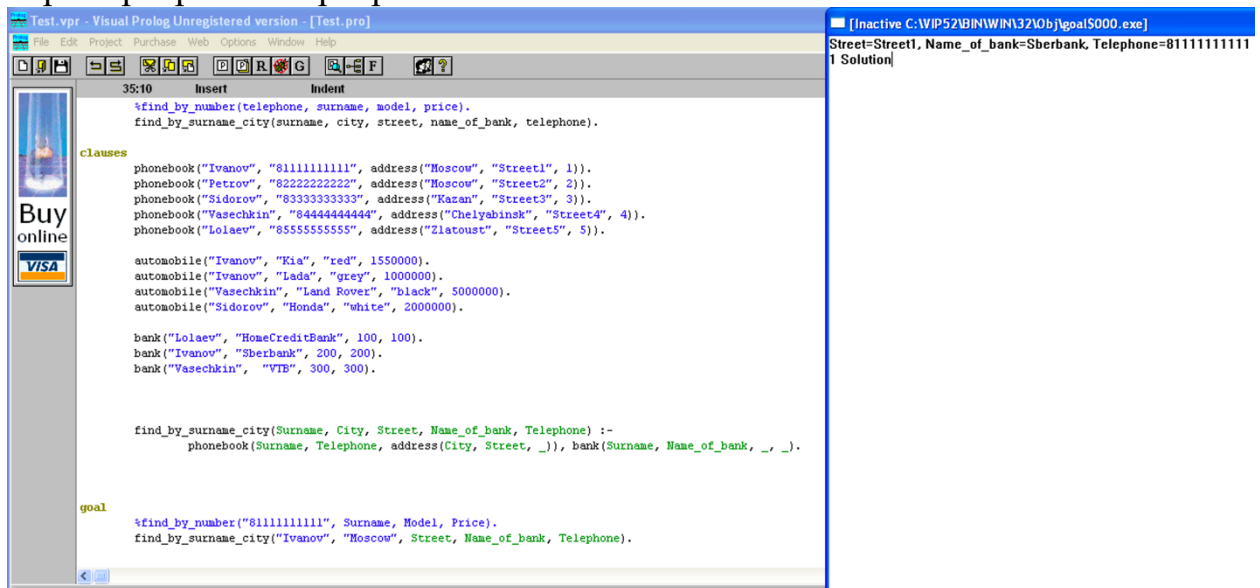
```
find_by_number(Telephone, Surname, Model, Price) :-  
phonebook(Surname, Telephone, _), automobile(Surname, Model, _, Price).
```

```
find_by_surname_city(Surname, City, Street, Name_of_bank, Telephone) :-  
phonebook(Surname, Telephone, address(City, Street, _)), bank(Surname, Name_of_bank, _, _).
```

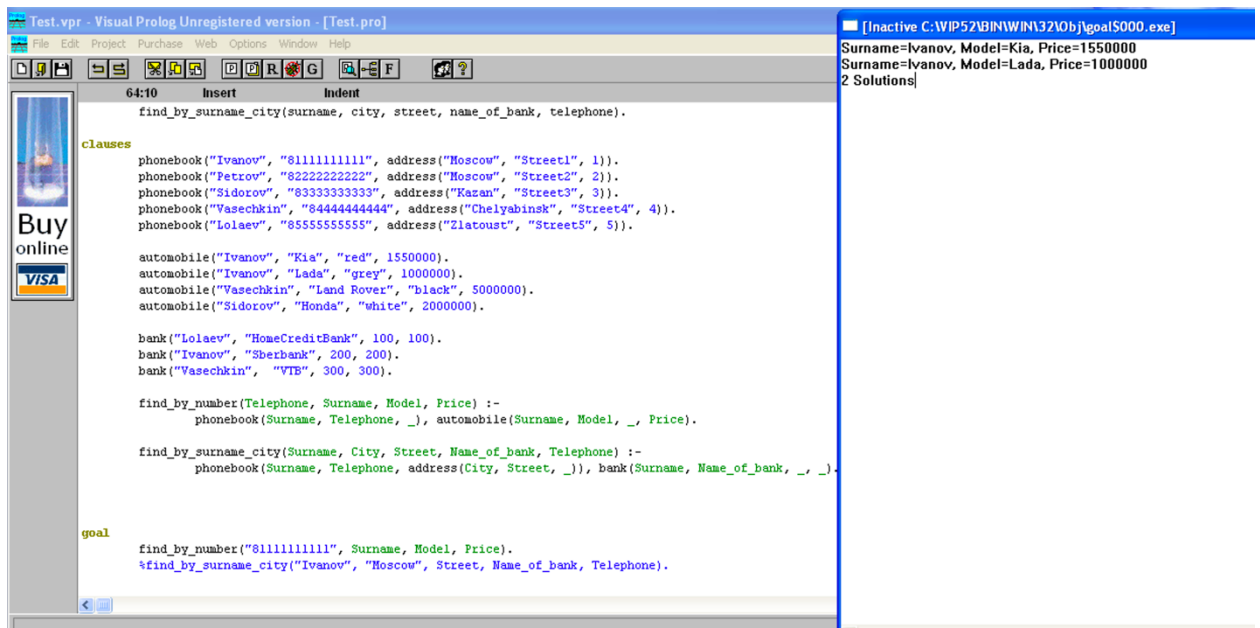
goal

```
%find_by_number("8111111111", Surname, Model, Price).  
find_by_surname_city("Ivanov", "Moscow", Street, Name_of_bank, Telephone).
```

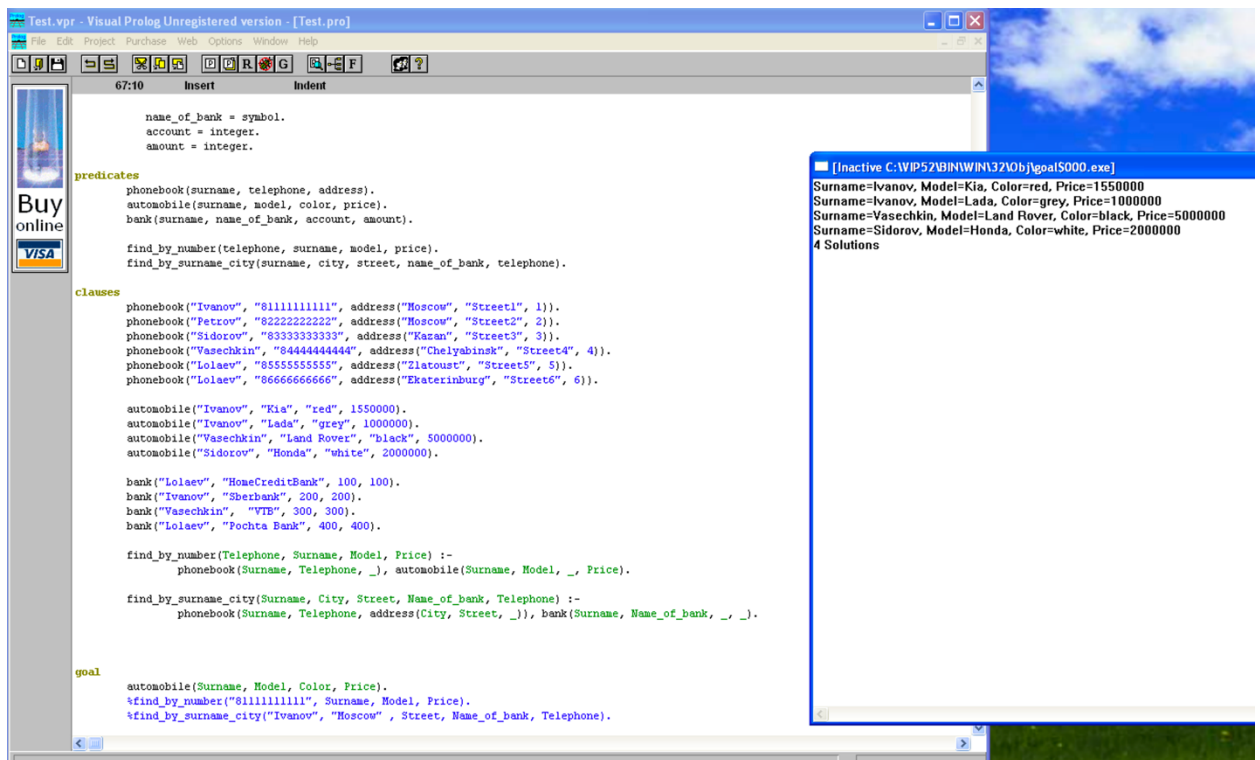
Примеры работы программы:



Вопрос: на какой улице живет, в каком банке имеет вклад и какой имеет телефон Иванов из Москвы?



Вопрос: Кто и какими машинами владеет человек с телефоном 8111111111?



Вопрос: Какие есть знания в базе данных автомобилей?

Теоретическая часть

1. Что такое терм?

Терм – это константа (число, символьный атом (комбинация символов латинского алфавита, цифр и символа подчеркивания) или строка); переменная (именованная или анонимная (обозначается символом подчеркивания “_”)); составной терм (средство организации групп отдельных элементов знаний в единый объект).

2. Что такое предикат в матлогике (математике)?

Предикат - утверждение, которое содержит переменные, принимающие значение 1 или 0 (истинно или ложно) в зависимости от значений переменных.

3. Что описывает предикат в Prolog?

Предикат описывает отношение, определяемое процедурой.

4. Назовите виды предложений в программе и приведите примеры таких предложений из Вашей программы. Какие предложения являются

основными, а какие – не основными? Каковы: синтаксис и семантика (формальный смысл) этих предложений (основных и неосновных)?

Предложения бывают двух видов: факты и правила. Предложение более общего вида – правило имеет вид:

$A :- B_1, \dots, B_n.$

A называется заголовком правила, а B_1, \dots, B_n – телом правила. Основные предложения – это предложения не содержащие переменных.

`find_by_number("8111111111", "Ivanov", "Kia", 1000000).`

Не основные – содержащие переменные.

`find_by_surname_city(Surname, City, Street, Name_of_bank, Telephone).`

5. Каковы назначение, виды и особенности использования переменных в программе на Prolog? Какое предложение БЗ сформулировано в более общей – абстрактной форме: содержащее или не содержащее переменных?

Переменные нужны для обозначения некоторого неизвестного объекта. Переменные бывают именованными или анонимными. Переменная уникальна в рамках предложения. В более абстрактной форме сформулировано БЗ не содержащее переменных.

6. Что такое подстановка?

Подстановка необходима для замены каждого вхождения неизвестной переменной предиката на соответствующий терм (задается в вопросе).

7. Что такое пример терма? Как и когда строится? Как Вы думаете, система строит и хранит примеры?

Пример терма – это результат подстановки некоторых конкретных значений в предикат, частный случай предиката. Пример терма строится после того, как задан вопрос и хранится до окончания работы программы. Примеры строятся во время алгоритма унификации.

find_by_number("8111111111", Surname, Model, Price).

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
1	find_by_number("8111111111", Surname, Model, Price) и find_by_number(Telephone, Surname, Model, Price) Связываются через Telephone = "8111111111"	find_by_number убирается из стека phonebook(Surname, Telephone, _) помещается в стек automobile(Surname, Model, _, Price) помещается в стек
2	phonebook(Surname, "8111111111", _) и phonebook("Ivanov", "8111111111", address("Moscow", "Street1", 1)) Связываются через Surname = "Ivanov"	phonebook(Surname, Telephone, _) убирается из стека
3	automobile("Ivanov", Model, _, Price) и automobile("Ivanov", "Kia", "red", 1550000). Связываются через Model = "Kia", Price = 1550000	automobile(Surname, Model, _, Price) убирается из стека Выводится: "Ivanov", "Kia", 1550000 automobile(Surname, Model, _, Price) заносится в стек Model и Price развязываются
4	automobile("Ivanov", Model, _, Price) и automobile("Ivanov", "Lada", "grey", 1000000) Связываются через Model = "Lada", Price = 1000000	automobile(Surname, Model, _, Price) убирается из стека Выводится: "Ivanov", "Lada", 1000000 automobile(Surname, Model, _, Price) заносится в стек Model и Price развязываются

5	<p>automobile("Ivanov", Model, _, Price)</p> <p>и</p> <p>automobile("Vasechkin", "Land Rover", "black", 5000000)</p> <p>не унифицируется</p>	
6	<p>automobile("Ivanov", Model, _, Price)</p> <p>и</p> <p>automobile("Ivanov", "Kia", "red", 1550000).</p> <p>не унифицируется</p>	
7	<p>... (все остальные факты automobile тоже не унифицируются)</p>	<p>Откат.</p> <p>Все факты automobile пройдены, в стек заносится последний вопрос</p> <p>phonebook(Surname, Telephone, _)</p> <p>Surname развязывается</p>
8	<p>phonebook(Surname, "8111111111", _)</p> <p>и</p> <p>phonebook("Ivanov", "8111111111", address("Moscow", "Street1", 1))</p> <p>Связываются через Surname = "Ivanov"</p>	<p>Откат.</p> <p>Все факты phonebook пройдены, в стек заносится последний вопрос</p> <p>find_by_number(Telephone, Surname, Model, Price)</p> <p>Telephone развязывается</p> <p>Система завершает работу.</p>

find_by_number("8666666666", Surname, Model, Price).

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
1	<p>find_by_number("8666666666", Surname, Model, Price)</p> <p>и</p> <p>find_by_number(Telephone, Surname, Model, Price)</p>	<p>find_by_number убирается из стека</p> <p>phonebook(Surname, Telephone, _) помещается в стек</p> <p>automobile(Surname, Model, _, Price) помещается в стек</p>

	Связываются через Telephone = "8111111111"	
2	<p>phonebook(Surname, "8111111111", _)</p> <p>и</p> <p>phonebook("Ivanov", "8111111111", address("Moscow", "Street1", 1))</p> <p>Не унифицируются</p>	
3	... (факты phonebook не унифицируются)	
4	<p>phonebook(Surname, "8666666666", _)</p> <p>и</p> <p>phonebook("Lolaev", "8666666666", address("Ekaterinburg", "Street6", 6)).</p> <p>Связываются через Surname = "Lolaev"</p>	<p>phonebook(Surname, Telephone, _)</p> <p>убирается из стека</p>
5	<p>automobile("Lolaev", Model, _, Price)</p> <p>и</p> <p>automobile("Vasechkin", "Land Rover", "black", 5000000)</p> <p>не унифицируется</p>	
6	... (факты automobile не унифицируются)	
7		<p>Откат.</p> <p>Все факты automobile пройдены, в стек заносится последний вопрос phonebook(Surname, Telephone, _)</p> <p>Surname развязывается</p>
8		<p>Откат.</p> <p>Все факты phonebook пройдены, в</p>

		<p>стек заносится последний вопрос</p> <p>find_by_number</p> <p>Telephone развязывается</p> <p>Система завершает работу.</p>
--	--	--

find_by_surname_city("Ivanov", "Moscow" , Street, Name_of_bank, Telephone).

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
1	<p>find_by_surname_city("Ivanov", "Moscow" , Street, Name_of_bank, Telephone).</p> <p>и</p> <p>find_by_surname_city(Surname, City, Street, Name_of_bank, Telephone</p> <p>Связываются через Surname = "Ivanov" и City = "Moscow"</p>	<p>find_by_surname_city убирается из стека</p> <p>phonebook(Surname, Telephone, address(City, Street, _)) помещается в стек</p> <p>bank(Surname, Name_of_bank, _, _) помещается в стек</p>
2	<p>phonebook("Ivanov", Telephone, address("Moscow", Street, _))</p> <p>и</p> <p>phonebook("Ivanov", "8111111111", address("Moscow", "Street1", 1))</p> <p>Связываются через Telephone = "8111111111" и Street = "Street1"</p>	<p>phonebook убирается из стека</p>
3	<p>bank("Ivanov", Name_of_bank, _, _)</p> <p>и</p> <p>bank("Lolaev", "HomeCreditBank", 100, 100)</p> <p>Не унифицируются.</p>	
4	<p>bank("Ivanov", Name_of_bank, _, _)</p>	<p>bank убирается из стека</p>

	и bank("Ivanov", "Sberbank", 200, 200) Связываются через Name_of_bank = "Sberbank"	Выводится: "Street1", "Sberbank", "81111111111" Bank заносится в стек Name_of_bank развязывается
5	... (факты bank не унифицируются)	
6		Откат. Все факты bank пройдены, в стек заносится последний вопрос phonebook(Surname, Telephone, address(City, Street, _)) Telephone и Street развязывается
7	... (факты phonebook не унифицируются)	
8		Откат. Все факты phonebook пройдены, в стек заносится последний вопрос find_by_surname_city Surname и City развязывается Система завершает работу.

