

	<p><b>Министерство образования и науки Российской Федерации</b> <b>Федеральное государственное бюджетное образовательное</b> <b>учреждение</b> <b>высшего образования</b> <b>«Московский государственный технический университет</b> <b>имени Н.Э. Баумана</b> <b>(национальный исследовательский университет)»</b> <b>(МГТУ им. Н.Э. Баумана)</b></p>
---	--

---

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Отчет по лабораторной работе № 11-13

По курсу: "Функциональное и логическое  
программирование"

Студент:

Турсунов Жасурбек Рустамович

Группа: ИУ7-66Б

Преподаватели:

Толпинская Наталья Борисовна

Строганов Юрий Владимирович

Москва, 2021 г.

# Содержание

Введение	2
Задание 1	4
Задание 2	6
Задание 3	7
Ответы на вопросы	14

**Цель работы:** Познакомится со средой Visual Prolog, познакомится со структурой программы: способом запуска и формой вывода результатов.

**Задачи работы:** изучить принципы работы в среде VisualProlog, возможность получения однократного и многократного результата, изучить базовые конструкции языка Prolog, структуру программы Prolog, форму ввода исходных данных и вывода результатов работы программы.

## Введение

Программа на Prolog представляет собой базу знаний и вопрос. База знаний содержит истинные значения, используя которые программа выдает ответ на вопрос. В процессе поиска ответа программа рассматривает все возможные альтернативные решения и находит все возможные значения переменных, при которых на поставленный вопрос можно ответить “ДА”.

Основным элементом языка является терм: константа, переменная или составной терм. Составной терм является предикатом. Программа представляет набор фактов и правил.

Факты представляют собой составные термы, с помощью которых фиксируется наличие истинности между объектами предметной области – аргументами терма.

Правила являются обобщённый формулировкой условий истинности отношения между объектами предметной области (аргументами терма), которое записано в заголовке правила. Условие истинности этого отношения является телом правила. Заголовок правила отделяется от тела правила символом «:-», правило завершается символом «.». Заголовок правила – это предикат. Тело правила может быть представлено последовательностью предикатов.

Основной элемент языка-терм.

Терминология:

### 1. Простой

- Константа(с маленькой буквы);
  - Символ;

- Число.
- Переменная (с большой буквы);
  - Именованная;
  - Анонимная.
- Составной (пример  $f(t_1, t_2, \dots, t_n)$ ). Где F-функтор (имя отношения между объектами)

### Особенность использования переменных

Именованные переменные уникальны в предикатах одного предложения, анонимные уникальны везде. Анонимные переменные не возвращают значение. Переменной можно обозначить любой объект. При описании переменная может потерять свое значение, но потом его можно вернуть. **Структура программы**  
Программа состоит из разделов:

1. Domains - отношение имен и структур объектов (не обязателен);
2. Predicates – описание предикатов (названий отношений между объектами);
3. Clauses – база знаний;
4. Goal – Раздел целевых утверждений.

Программа состоит из предложений:

1. Факт (безусловная истина, формулируется составным термом)- частный случай правил;
2. Правила (условная истина, способ порождения новых фактов на основе имеющихся).

Вопрос:

1. Конъюнктивный (B1, B2, B3);
2. Дизъюнктивный (B1; B2; B3).

# Задание 1

Разработать свою программу - "Телефонный справочник". Протестировать работу программы.

```
1 domains
2     NAME=symbol
3     NUM=string
4     AREA=integer
5     CITY=string
6     STREET=string
7     HOUSE=integer
8
9 predicates
10    likes(symbol,symbol)
11    abonent(NAME,NUM)
12    abonname(NAME,NUM)
13    abonnum(NAME,NUM)
14    house(NAME,AREA,CITY,STREET,HOUSE)
15    housesAddr(NAME,CITY,STREET,HOUSE)
16    housesAREA(NAME,AREA)
17    mult(NAME,NUM,CITY)
18
19 clauses
20    likes(ellen,tennis).
21    likes(john,football).
22    likes(tom,baseball).
23    likes(eric,swimming).
24    likes(mark,tennis).
25    likes(bill,Activity):-likes (tom, Activity).
26
27    abonent(alex,"1111111").
28    abonent(alex,"1112121").
29    abonent(ivan,"2222222").
30    abonent(petr,"3333333").
31    abonent(semen,"4444444").
32    abonent(evgen,"555555").
33    abonent(dima,"6666666").
34    abonent(semen,"777777").
```

```

35     abonent(oleg,"888888").
36     abonent(roman,"9999999").
37
38     house(alex,2500,"Moscow","Baker",140).
39     house(alex,560,"London","Baker",221).
40     house(alex,70,"NY","Baker",14).
41     house(ivan,2500,"Moscow","Lyubanka",10).
42     house(semen,56,"London","Tsentrantly",14).
43     house(dima,700,"NY","Tsemntr",221).
44
45
46     abonname(NAME,NUM):-abonent(NAME,NUM).
47     abonnum(NAME,NUM):-abonent(NAME,NUM).
48     housesAddr(NAME,CITY,STREET,HOUSE):-house(NAME,_,CITY,STREET,HOUSE).
49     housesAREA(NAME,AREA):-house(NAME,AREA,_,_,_).
50     mult(NAME,NUM,CITY):-house(NAME,_,CITY,_,_),abonent(NAME,NUM).
51
52
53
54 goal
55     abonname(alex,NUM).
56     %housesAddr(alex,CITY,STREET,HOUSE).
57     %housesAREA(alex,AREA).
58     %mult(alex,NUM,ADDRESS).
59     %house(alex,_,ADDRESS),abonent(alex,NUM).
60

```

#### Результат работы программы

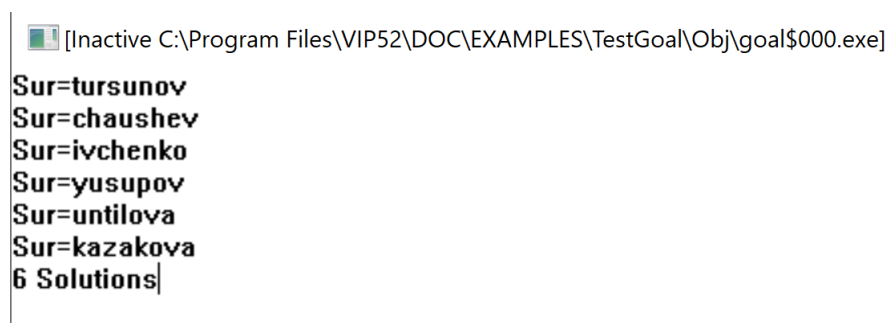
Вопрос	Результат
abonname(alex,NUM).	NUM=1111111/ NUM=1112121/ 2 Solutions
housesAddr(alex,CITY, STREET,HOUSE).	CITY=Moscow, STREET=Baker, HOUSE=140
housesAREA(alex,AREA).	AREA=2500, AREA=560, AREA=70, 3 Solutions
mult(alex,NUM,ADDRESS).	NUM=1111111, ADDRESS=Moscow

## Задание 2

Разработать свою программу - "Базу знаний с помощью которой, например , множество студентов обучающихся в одной группе. Привести примеры возможных вариантов вопросов и варианты ответов(не менее 3)

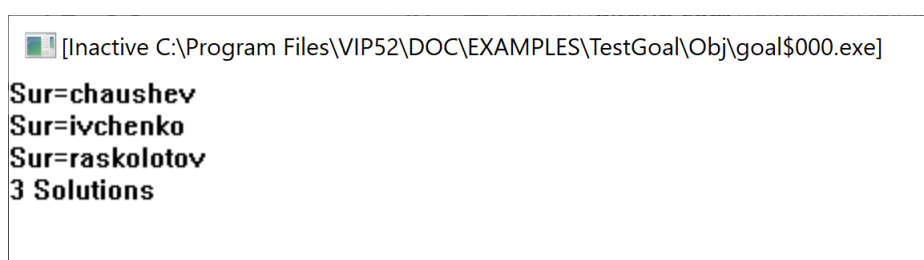
```
1 predicates
2     student(symbol, symbol, symbol, symbol, real, symbol).
3     group(symbol, symbol).
4     district(symbol, symbol).
5     grantL(symbol).
6 clauses
7     student(tursunov, jasur, iu766, zelenograd, 4.5, no).
8     student(chaushev, alexandr, iu766, izmaylovo, 4.75, no).
9     student(ivchenko, artem, iu766, izmaylovo, 3.75, yes).
10    student(yusupov, felix, iu766, otradnoye, 4, no).
11    student(untilova, arina, iu766, otradnoye, 5.0, no).
12    student(kazakova, eliza, iu766, balashiha, 4.25, no).
13    student(raskolotov, dima, iu764, izmaylovo, 5.0, no).
14    student(savinov, egor, iu764, otradnoye, 3.25, yes).
15
16    group(Sur, Group) :- student(Sur, _, Group, _, _, _).
17    district(Surname, District) :- student(Surname, _, _, District, _, _).
18    grantL(Surname) :- student(Surname, _, _, _, Mean, _), Mean>=4.
19    grantL(Surname) :- student(Surname, _, _, _, _, Info), Info=yes.
20 goal
21     %group (Sur, iu766).
22     %district (Sur, izmaylovo).
23     grantL(Sur).
```

## Примеры работы:



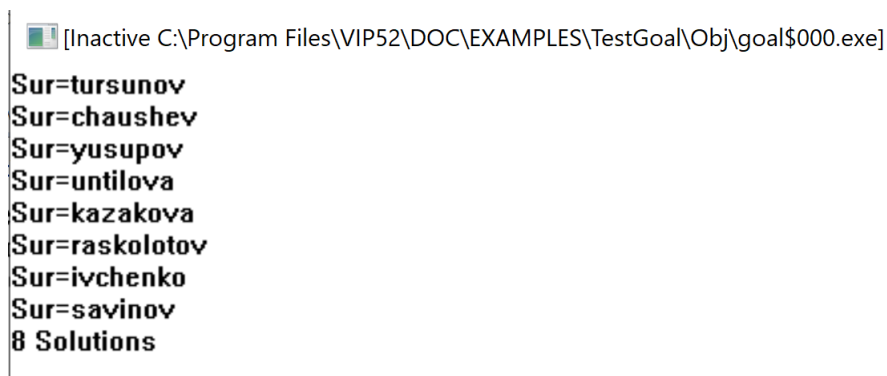
[Inactive C:\Program Files\VIP52\DOC\EXAMPLES\TestGoal\Obj\goal\$000.exe]  
**Sur=tursunov**  
**Sur=chaushev**  
**Sur=ivchenko**  
**Sur=yusupov**  
**Sur=untilova**  
**Sur=kazakova**  
**6 Solutions**

Рис. 1: Вопрос: group (Sur, iu766).



[Inactive C:\Program Files\VIP52\DOC\EXAMPLES\TestGoal\Obj\goal\$000.exe]  
**Sur=chaushev**  
**Sur=ivchenko**  
**Sur=raskolotov**  
**3 Solutions**

Рис. 2: Вопрос: district (Sur, izmaylovo).



[Inactive C:\Program Files\VIP52\DOC\EXAMPLES\TestGoal\Obj\goal\$000.exe]  
**Sur=tursunov**  
**Sur=chaushev**  
**Sur=yusupov**  
**Sur=untilova**  
**Sur=kazakova**  
**Sur=raskolotov**  
**Sur=ivchenko**  
**Sur=savinov**  
**8 Solutions**

Рис. 3: Вопрос: grantL(Sur).

## Задание 3

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»: , 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.
  - По № телефона найти: Фамилию, Марку автомобиля, Стоимость автомобиля (может быть несколько);
  - 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.

Для задания 1 и задания 2:

Для одного из вариантов ответов, и для а) и для в), описать словесно порядок поиска ответа на вопрос, указав, как выбираются знания, и, при этом, для каждого этапа унификации, выписать подстановку – наибольший общий унификатор, и соответствующие примеры термов.

```

1 domains
2     surname, number, city, street, brand, model, color, bank, account = symbol.
3     price, money = integer.
4     address_t = address(city, street, integer, integer).
5
6 predicates
7     person(surname, number, address_t).
8     car(surname, brand, model, color, price).
9     deposit(surname, address_t, bank, account, money).
10    all_by_phone(number, surname, brand, price).
11    brand_by_phone(number, brand).
12    by_surname_city(surname, city, street, bank, number).
13
14 clauses
15    person("Ivanov", "000-000", address("Example", "street", 0, 0)).
16    person("Ivanov", "111-111", address("City-17", "Gordon street", 0, 0)).
17    person("Petrov", "001-917", address("St. Petersburg", "Lenina", 24, 42)).

```

```

18     person("Sidorov", "555-555", address("Los Angeles", "Apple street", 0, 1)).
19     person("A", "123-456", address("B", "C avenue", 13, 14)).
20     person("Another", "123-321", address("One", "Good street", 3, 12)).
21     person("One", "999-666", address("More", "Pioneer street", 3, 4)).
22     person("Not", "987-654", address("Enough", "Bad Fantasy avenue", 9, 9)).
23
24     car("Ivanov", "Bugatti", "La Voiture Noire", "Black", 1178000).
25     car("Ivanov", "Aston Martin", "Valkyrie", "Grey", 230000).
26     car("Petrov", "Lada", "Kalina", "White", 200).
27     car("Sidorov", "Ford", "Focus", "Red", 400).
28
29     deposit("Ivanov", address("Example", "street", 0, 0),
30           "Sberbank", "0-0-0-0", 999999999).
31     deposit("Ivanov", address("Example", "street", 0, 0),
32           "VTB", "0-0-0-1", 1).
33     deposit("Ivanov", address("City-17", "Gordon street", 0, 0),
34           "Tinkoff", "0-1-0-1", 987654).
35     deposit("Petrov", address("St. Petersburg", "Lenina", 24, 42),
36           "Alfa", "1-2-3-4", 999999999).
37     deposit("Sidorov", address("Los Angeles", "Apple street", 0, 1),
38           "Mavrodi", "6-9-6-9", 1).
39
40
41     by_surname_city(Surname, City, Street, Bank, Number) :-
42         person(Surname, Number, address(City, Street, _, _)),
43         deposit(Surname, address(City, Street, _, _), Bank, _, _).
44     person("Another", "123-321", address("One", "Good street", 3, 12)).
45     person("One", "999-666", address("More", "Pioneer street", 3, 4)).
46     person("Not", "987-654", address("Enough", "Bad Fantasy avenue", 9, 9)).
47
48     car("Ivanov", "Bugatti", "La Voiture Noire", "Black", 1178000).
49     car("Ivanov", "Aston Martin", "Valkyrie", "Grey", 230000).
50     car("Petrov", "Lada", "Kalina", "White", 200).
51     car("Sidorov", "Ford", "Focus", "Red", 400).
52
53     deposit("Ivanov", address("Example", "street", 0, 0),
54           "Sberbank", "0-0-0-0", 999999999).
55     deposit("Ivanov", address("Example", "street", 0, 0),
56           "VTB", "0-0-0-1", 1).

```

```

57     deposit("Ivanov", address("City-17", "Gordon street", 0, 0),
58           "Tinkoff", "0-1-0-1", 987654).
59     deposit("Petrov", address("St. Petersburg", "Lenina", 24, 42),
60           "Alfa", "1-2-3-4", 999999999).
61     deposit("Sidorov", address("Los Angeles", "Apple street", 0, 1),
62           "Mavrodi", "6-9-6-9", 1).
63     deposit("Another", address("One", "Good street", 3, 12),
64           "Bankname", "10-20-30-40", 40302010).
65
66     all_by_phone(Number, Surname, Brand, Price) :- person(Surname, Number, _),
67           car(Surname, Brand, _, _, Price).
68     brand_by_phone(Number, Brand) :- all_by_phone(Number, _, Brand, _).
69     by_surname_city(Surname, City, Street, Bank, Number) :-
70           person(Surname, Number, address(City, Street, _, _)),
71           deposit(Surname, address(City, Street, _, _), Bank, _, _).
72
73 goal
74     %all_by_phone("000-000", Surname, Brand, Price).
75     %brand_by_phone("000-000", Brand).
76     %by_surname_city("Ivanov", "Example", Street, Bank, Number).

```

### Примеры работы:

Surname=Ivanov, Brand=Bugatti, Price=11780000

Surname=Ivanov, Brand=Aston Martin, Price=2300000

2 Solutions

Ответ на вопрос all by phone("000-000 Surname, Brand, Price).

Brand=Bugatti

Brand=Aston Martin

2 Solutions

Ответ на вопрос brand by phone("000-000 Brand).

Street=street, Bank=Sberbank, Number=000-000

Street=street, Bank=VTB, Number=000-000

2 Solutions

Ответ на вопрос by surname city("Ivanov "Example Street, Bank, Number).

Описание порядка поиска ответа.

1) all by phone("000-000 Surname, Brand, Price).

№ шага	Сравниваемые термины; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
1	all_by_phone("000-000", Surname, Brand, Price). и all_by_phone(Number, Surname, Brand, Price) :- person(Surname, Number, _), car(Surname, Brand, _, _, Price).  Number = "000-000"	прямой ход. Нужно вычислить person и car
2	person(Surname,"000-000", _) и person("Ivanov", "000-000", address("Example", "street", 0, 0)).  Surname="Ivanov"	person <u>вычислен</u> , вычисляем car
3	car("Ivanov", Brand, _, _, Price) и car("Ivanov", "Bugatti", "La Voiture Noire", "Black", 1178000)  Brand="Bugati", Price=1178000	car вычислен, получен результат 1. Ищем остальные результаты
4	car("Ivanov", Brand, _, _, Price) и car("Ivanov", "Aston Martin", "Valkyrie", "Grey", 230000).  Brand="Aston Martin", Price=230000	car вычислен, получен результат 2. Ищем остальные результаты
5	car("Ivanov", Brand, _, _, Price) и car("Petrov", "Lada", "Kalina", "White", 200)  Примеры не совпадают	car вычислен, новых результатов нет, ищем остальные
6	car("Ivanov", Brand, _, _, Price) и car("Sidorov", "Ford", "Focus", "Red", 400)  Примеры не совпадают	car вычислен, новых результатов нет, ищем остальные  все примеры с car для Surname="Ivanov" разобраны, ищем другие Surname
7	person(Surname,"000-000", _) и person("Ivanov", "111-111", address("City-17", "Gordon street", 0, 0)).  Примеры не совпадают	person вычислен, новых результатов нет, ищем остальные
8	person(Surname,"000-000", _) и person("Petrov", "001-917", address("St. Petersburg", "Lenina", 24, 42)).  Примеры не совпадают	person вычислен, новых результатов нет, ищем остальные
9	person(Surname,"000-000", _) и person("Sidorov", "555- 555", address("Los Angeles", "Apple street", 0, 1)).  Примеры не совпадают	person вычислен, новых результатов нет, ищем остальные

10	<code>person(Surname,"000-000",_) и person("A", "123-456", address("B", "C avenue", 13, 14)).</code>  Примеры не совпадают	person вычислен, новых результатов нет, ищем остальные
11	<code>person(Surname,"000-000",_) и person("Another", "123-321", address("One", "Good street", 3, 12)).</code>  Примеры не совпадают	person вычислен, новых результатов нет, ищем остальные
12	<code>person(Surname,"000-000",_) и person("One", "999-666", address("More", "Pioneer street", 3, 4)).</code>  Примеры не совпадают	person вычислен, новых результатов нет, ищем остальные
13	<code>person(Surname,"000-000",_) и person("Not", "987-654", address("Enough", "Bad Fantasy avenue", 9, 9)).</code>  Примеры не совпадают	person вычислен, новых результатов нет. Все примеры с person разобраны. <code>all_by_phone</code> вычислен, получено 2 результата

## 2) brand by phone("000-000 Brand).

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
1	<code>brand_by_phone("000-000", Brand) и</code> <code>brand_by_phone(Number, Brand) :- all_by_phone(Number, _, Brand, _).</code>  Number = "000-000"	прямой ход. Нужно вычислить <code>all_by_phone</code>
2	<code>all_by_phone("000-000", _, Brand, _).</code> и <code>all_by_phone(Number, Surname, Brand, Price) :- person(Surname, Number, _), car(Surname, Brand, _, _, Price).</code>  Number = "000-000"	прямой ход. Нужно вычислить person и car
3	<code>person(Surname,"000-000",_) и person("Ivanov", "000-000", address("Example", "street", 0, 0)).</code>  Surname="Ivanov"	person вычислен, вычисляем car
4	<code>car("Ivanov", Brand, _, _, Price) и car("Ivanov", "Bugatti", "La Voiture Noire", "Black", 1178000)</code>  Brand="Bugati", Price=1178000	car вычислен, получен результат 1. Ищем остальные результаты
5	<code>car("Ivanov", Brand, _, _, Price) и car("Ivanov", "Aston Martin", "Valkyrie", "Grey", 230000).</code>  Brand="Aston Martin", Price=230000	car вычислен, получен результат 2. Ищем остальные результаты
6	<code>car("Ivanov", Brand, _, _, Price) и car("Petrov", "Lada", "Kalina", "White", 200)</code> Примеры не совпадают	car вычислен, новых результатов нет, ищем остальные

7	<p><code>car("Ivanov", Brand, _, Price)</code> и <code>car("Sidorov", "Ford", "Focus", "Red", 400)</code></p> <p>Примеры не совпадают</p>	<p>car вычислен, новых результатов нет, ищем остальные</p> <p>все примеры с car для Surname="Ivanov" разобраны, ищем другие Surname</p>
8	<p><code>person(Surname,"000-000", _) и person("Ivanov", "111-111", address("City-17", "Gordon street", 0, 0)).</code></p> <p>Примеры не совпадают</p>	<p>person вычислен, новых результатов нет, ищем остальные</p>
9	<p><code>person(Surname,"000-000", _) и person("Petrov", "001-917", address("St. Petersburg", "Lenina", 24, 42)).</code></p> <p>Примеры не совпадают</p>	<p>person вычислен, новых результатов нет, ищем остальные</p>
10	<p><code>person(Surname,"000-000", _) и person("Sidorov", "555-555", address("Los Angeles", "Apple street", 0, 1)).</code></p> <p>Примеры не совпадают</p>	<p>person вычислен, новых результатов нет, ищем остальные</p>
11	<p><code>person(Surname,"000-000", _) и person("A", "123-456", address("B", "C avenue", 13, 14)).</code>Примеры не совпадают</p>	<p>person вычислен, новых результатов нет, ищем остальные</p>
12	<p><code>person(Surname,"000-000", _) и person("Another", "123-321", address("One", "Good street", 3, 12)).</code>Примеры не совпадают</p>	<p>person вычислен, новых результатов нет, ищем остальные</p>
12	<p><code>person(Surname,"000-000", _) и person("One", "999-666", address("More", "Pioneer street", 3, 4)).</code>Примеры не совпадают</p>	<p>person вычислен, новых результатов нет, ищем остальные</p>
13	<p><code>person(Surname,"000-000", _) и person("Not", "987-654", address("Enough", "Bad Fantasy avenue", 9, 9)).</code></p> <p>Примеры не совпадают</p>	<p>person вычислен, новых результатов нет. Все примеры с person разобраны. <code>all_by_phone</code> вычислен, <code>brand_by_phone</code> вычислен получено 2 результата</p>

### 3) by surname city("Ivanov "Example Street, Bank, Number).

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
1	<p><code>by_surname_city("Ivanov", "Example", Street, Bank, Number)</code> и <code>by_surname_city(Surname, City, Street, Bank, Number) :- person(Surname, Number, address(City, Street, _, _)), deposit(Surname, address(City, Street, _, _), Bank, _, _)</code></p> <p>Surname="Ivanov" City="Example"</p>	<p>Прямой ход. Необходимо вычислить person и deposit</p>
2	<p><code>person("Ivanov", Number, address("Example", Street, _, _)) и person("Ivanov", "000-000", address("Example", "street", 0, 0))</code></p> <p>Number="000-000" Street="street"</p>	<p>Person вычислен, вычисляем deposit</p>

3	<code>deposit("Ivanov", address("Example", "street", _, _), Bank, _, _) и deposit("Ivanov", address("Example", "street", 0, 0), "Sberbank", "0-0-0-0", 999999999). Bank="Sberbank"</code>	deposit вычислен, получен результат 1, ищем остальные результаты
4	<code>deposit("Ivanov", address("Example", "street", _, _), Bank, _, _) и deposit("Ivanov", address("Example", "street", 0, 0), "VTB", "0-0-0-1", 1). Bank="VTB"</code>	deposit вычислен, получен результат 2, ищем остальные результаты
5	<code>deposit("Ivanov", address("Example", "street", _, _), Bank, _, _) и deposit("Ivanov", address("City-17", "Gordon street", 0, 0), "Tinkoff", "0-1-0-1", 987654) Примеры не совпадают</code>	deposit вычислен, новых результатов нет, ищем остальные
6	<code>deposit("Ivanov", address("Example", "street", _, _), Bank, _, _) и deposit("Petrov", address("St. Petersburg", "Lenina", 24, 42), "Alfa", "1-2-3-4", 999999999) Примеры не совпадают</code>	deposit вычислен, новых результатов нет, ищем остальные
7	<code>deposit("Ivanov", address("Example", "street", _, _), Bank, _, _) и deposit("Sidorov", address("Los Angeles", "Apple street", 0, 1), "Mavrodi", "6-9-6-9", 1) Примеры не совпадают</code>	deposit вычислен, новых результатов нет, ищем остальные
8	<code>deposit("Ivanov", address("Example", "street", _, _), Bank, _, _) и deposit("Another", address("One", "Good street", 3, 12), "Bankname", "10-20-30-40", 40302010) Примеры не совпадают</code>	deposit вычислен, новых результатов нет, все примеры для deposit при Number="000-000" и Street="street" рассмотрены, ищем новые Number и Street
9	<code>person("Ivanov", Number, address("Example", Street, _, _)) и person("Ivanov", "111-111", address("City-17", "Gordon street", 0, 0)) Примеры не совпадают</code>	person вычислен, новых результатов нет, ищем остальные
10	<code>person("Ivanov", Number, address("Example", Street, _, _)) и person("Petrov", "001-917", address("St. Petersburg", "Lenina", 24, 42)) Примеры не совпадают</code>	person вычислен, новых результатов нет, ищем остальные
11	<code>person("Ivanov", Number, address("Example", Street, _, _)) и person("Sidorov", "555-555", address("Los Angeles", "Apple street", 0, 1)) Примеры не совпадают</code>	person вычислен, новых результатов нет, ищем остальные
12	<code>person("Ivanov", Number, address("Example", Street, _, _)) и person("A", "123-456", address("B", "C avenue", 13, 14)) Примеры не совпадают</code>	person вычислен, новых результатов нет, ищем остальные
13	<code>person("Ivanov", Number, address("Example", Street, _, _)) и person("Another", "123-321", address("One", "Good street", 3, 12)) Примеры не совпадают</code>	person вычислен, новых результатов нет, ищем остальные
14	<code>person("Ivanov", Number, address("Example", Street, _, _)) и person("One", "999-666", address("More", "Pioneer street", 3, 4)) Примеры не совпадают</code>	person вычислен, новых результатов нет, ищем остальные
15	<code>person("Ivanov", Number, address("Example", Street, _, _)) и person("Not", "987-654", address("Enough", "Bad Fantasy avenue", 9, 9)) Примеры не совпадают</code>	person вычислен, новых результатов нет, все примеры с person разобраны. by_surname_city вычислен получено 2 результата

## Ответы на вопросы:

### 1) Что такое терм?

Терм – основной элемент языка Prolog. Может быть константой (число, символ-

ный атом, строка), переменной (именованной или анонимной) или составным термом (функтуатор(терм1, терм2, ...)).

## 2) Что такое предикат в матлогике (математике)?

Предикат (n -местный, или n-арный) — это функция с множеством значений 0, 1 (или ложь, истина), определённая на множестве  $M = M_1 \times M_2 \times \dots \times M_n$ . Таким образом, каждый набор элементов множества M характеризуется либо как «истинный», либо как «ложный».

## 3) Что описывает предикат в Prolog?

В Prolog предикат описывает общий вид фактов и правил.

## 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)?

Предложение может быть фактом или правилом. Предложения бывают основные (не содержащие переменные) и неосновные (содержащие).

Пример:

1) `person("Ivanov "000-000 address("Example "street 0, 0)).` - основное `all_by_phone(Number, Surname, Brand, Price) :- person(Surname, Number, _), car(Surname, Brand, _, Price).` – неосновное (правило)

`all_by_phone("000-000 Surname, Brand, Price).` – неосновное (запрос) синтаксис основных: `имя_предиката(значение1, значение2, ...)`. синтаксис неосновных: `имя_предиката(зн...`

## 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?

Переменная обозначает данные, одинаковые внутри предложения. Переменные должны начинаться с большой буквы или символа подчеркивания и состоять только из символов латинского алфавита и символов подчеркивания. Переменные бывают именованные и анонимные (их имя – один символ «\_»). Предложение, сформулированное с помощью переменных, является более абстрактным, так как не требует заранее зафиксированного значения, а лишь описывает, какие зна-



чения должны совпадать. Соответственно, по запросам с разными значениями одни и те же предложения могут давать разные ответы, если были описаны с переменными.

**6) Что такое подстановка?**

Подстановка – набор термов  $t_1, t_2, \dots, t_N$ , подставляемых в терм  $A(X_1, X_2, \dots, X_N)$ . Применение подстановки заключается в замене каждого вхождения переменной  $x_i$  на соответствующий терм. Пусть  $\delta = x_1 = t_1, x_2 = t_2, \dots, x_n = t_n$  – подстановка. Обозначение ее результата:  $A\delta$ .

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

Терм  $B$  называется примером терма  $A$ , если существует такая подстановка  $\delta$ , что  $B=A\delta$ . Вероятно, система строит примеры при вычислении ответа на вопрос, и может их хранить, если в данном правиле часто идет обращение к данному примеру.