



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 № 1**

**Дисциплина:** Функциональное и логическое программирование

**Студент:** Платонова Ольга

**Группа:** ИУ7-65Б

**Преподаватели:** Толпинская Н. Б.  
Строганов Ю. В.

Москва, 2021 г.

## Часть 1.

Запустить среду Visual Prolog 5.2. Настроить утилиту TestGoal. Запустить тестовую программу, проанализировать реакцию системы и множество ответов. Разработать свою программу – «Телефонный справочник». Протестировать работу программы.

### Листинг 1.

```
phone.pro
12:38 Insert Indent
predicates
    record(string, string, string).

clauses
    record(pasha, "+7499123", home).
    record(danya, "+7499859", home).
    record(denis, "+7903810", mobile).
    record(dasha, "+7499109", home).
    record(dasha, "+7915918", mobile).

goal
    record(pasha, Number, Phone).

[Inactive C:\VIP52\DOC\EXAMPLES\TestGoal\Obj\goal$000.exe]
Number=+7499123, Phone=home
1 Solution|
```

```
phone.pro
12:38 Insert Indent
predicates
    record(string, string, string).

clauses
    record(pasha, "+7499123", home).
    record(danya, "+7499859", home).
    record(denis, "+7903810", mobile).
    record(dasha, "+7499109", home).
    record(dasha, "+7915918", mobile).

goal
    record(dasha, Number, Phone).

[Inactive C:\VIP52\DOC\EXAMPLES\TestGoal\Obj\goal$000.exe]
Number=+7499109, Phone=home
Number=+7915918, Phone=mobile
2 Solutions|
```

### Вывод 1.

*Что собой представляет программа на Prolog?*

Программа на Prolog представляет собой базу знаний и вопрос. База знаний состоит из предложений – CLAUSES: фактов и правил. Каждое предложение заканчивается точкой. Вопрос является составным термом. Система рассматривает вопрос как цель, к которой надо стремиться.

*Какова структура программы?*

Программа на Prolog состоит из разделов, который начинается со своего заголовка.

- Директивы компилятора – зарезервированные символьные константы.
- CONSTANTS – раздел описания констант.
- DOMAINS – раздел описания доменов.
- DATABASE – раздел описания предикатов внутренней базы данных.
- PREDICATES – раздел описания предикатов.
- CLAUSES – раздел описания предложений базы знаний.
- GOAL – раздел описания внутренней цели.

*Как программа реализуется, как формируются результаты работы программы?*

В процессе выполнения программы – система, используя встроенный алгоритм унификации, пытается обосновать возможность истинности вопрос, строя подстановки и примеры термов. Алгоритм унификации автоматически и многократно запускается системой.

При запуске алгоритма унифицирования в стек помещается равенство  $B = A\theta$ . Затем из стека считывается рабочее поле и обрабатывается 1 равенство. Цикл завершается, если стек пуст или поступило сообщение об отказе. Дополнительно используется переменная «неудача». В результате работы алгоритма унификации результирующая подстановка будет располагаться в результирующей ячейке памяти.

## Часть 2.

Составить программу – базу знаний, с помощью которой можно определить, например, множество студентов, обучающихся в одном ВУЗе. Студент может одновременно обучаться в нескольких ВУЗах. Привести примеры возможных вариантов вопросов и варианты ответов. Описать порядок формирования ответа. Исходную базу знаний сформировать с помощью только фактов.

### Листинг 2.

#### 1. Множество студентов из университета “BMSTU”.

```
students.pro
22:10 Insert Indent
domains
    id = integer.
    name, university = string.

predicates
    student(id, name, university).
    manyUnivers(id, name).

clauses
    student(1, pasha, "BMSTU").
    student(2, danya, "MSU").
    student(3, denis, "HSE").
    student(4, denis, "BMSTU").
    student(5, dasha, "MISIS").
    student(6, dasha, "MEPHI").

    manyUnivers(Id, Name) :- student(Id, Name, U), student(Idx, Name, Ux), Id > Idx.

goal
    student(Id, Name, "BMSTU").
```

[Inactive C:\VIP52\DOC\EXAMPLES\TestGoal\Obj\goal\$000.exe]

**Id=1, Name=pasha**  
**Id=4, Name=denis**  
**2 Solutions**

2. Множество университетов, в которых обучается “dasha”.

```
students.pro
21:9      Insert      Indent
domains
    id = integer.
    name, university = string.

predicates
    student(id, name, university).
    manyUnivers(id, name).

clauses
    student(1, pasha, "BMSTU").
    student(2, danya, "MSU").
    student(3, denis, "HSE").
    student(4, denis, "BMSTU").
    student(5, dasha, "MISIS").
    student(6, dasha, "MEPHI").

    manyUnivers(Id, Name) :- student(Id, Name, U), student(Idx, Name, Ux), Id > Idx.

goal
    %student(Id, Name, "BMSTU").
    student(Id, dasha, University).
```

[Inactive C:\VIP52\DOC\EXAMPLES\TestGoal\Obj\goal\$000.exe]

**Id=5, University=MISIS**  
**Id=6, University=MEPHI**  
**2 Solutions**

3. Информация о студенте номер 2.

```
students.pro
22:9      Insert      Indent
domains
    id = integer.
    name, university = string.

predicates
    student(id, name, university).
    manyUnivers(id, name).

clauses
    student(1, pasha, "BMSTU").
    student(2, danya, "MSU").
    student(3, denis, "HSE").
    student(4, denis, "BMSTU").
    student(5, dasha, "MISIS").
    student(6, dasha, "MEPHI").

    manyUnivers(Id, Name) :- student(Id, Name, U), student(Idx, Name, Ux), Id > Idx.

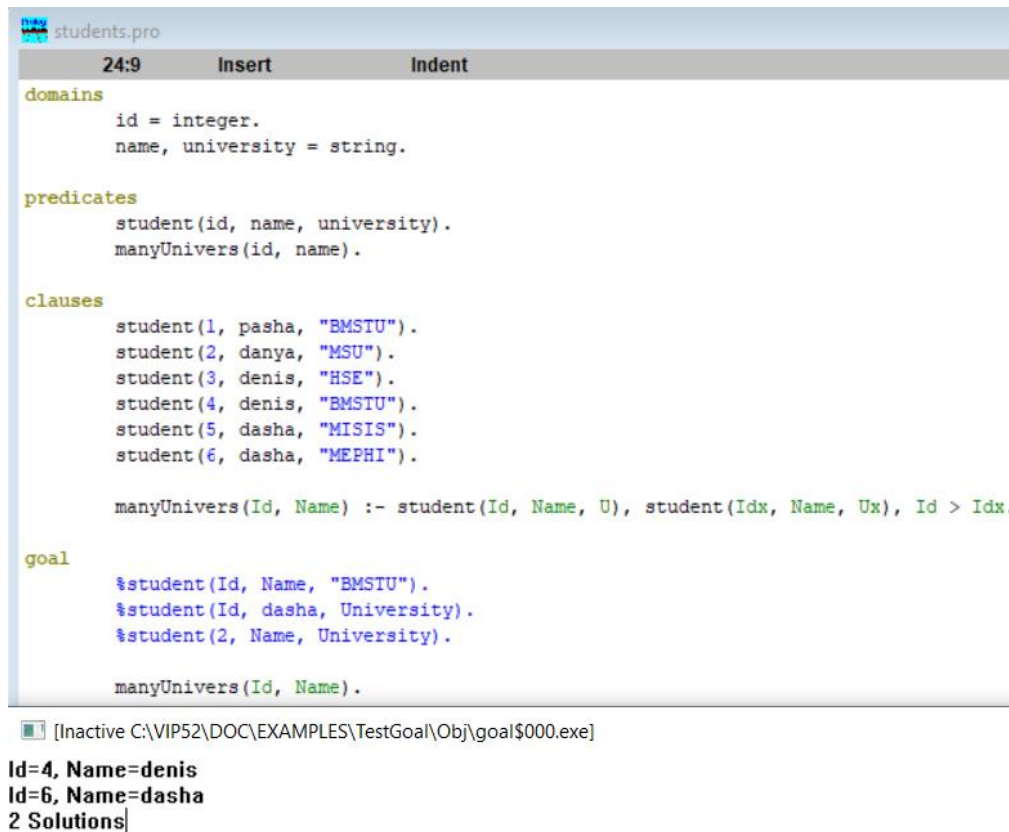
goal
    %student(Id, Name, "BMSTU").
    %student(Id, dasha, University).
    student(2, Name, University).
```

[Inactive C:\VIP52\DOC\EXAMPLES\TestGoal\Obj\goal\$000.exe]

**Name=danya, University=MSU**  
**1 Solution**

4. Список студентов, обучающихся более, чем в 1 ВУЗе.

В данном примере используется правило “manyUniverse”.



```
students.pro
24:9      Insert      Indent
domains
    id = integer.
    name, university = string.

predicates
    student(id, name, university).
    manyUnivers(id, name).

clauses
    student(1, pasha, "BMSTU").
    student(2, danya, "MSU").
    student(3, denis, "HSE").
    student(4, denis, "BMSTU").
    student(5, dasha, "MISIS").
    student(6, dasha, "MEPHI").

    manyUnivers(Id, Name) :- student(Id, Name, U), student(Idx, Name, Ux), Id > Idx.

goal
    %student(Id, Name, "BMSTU").
    %student(Id, dasha, University).
    %student(2, Name, University).

    manyUnivers(Id, Name).
```

[Inactive C:\VIP52\DOC\EXAMPLES\TestGoal\Obj\goal\$000.exe]

**Id=4, Name=denis**  
**Id=6, Name=dasha**  
**2 Solutions**

## Вывод 2.

### *Назначение и использование переменных.*

Факты содержат переменные (Id, Name, University), которые записываются с большой буквы или с `_`. Вообще – переменные предназначены для передачи знаний «во времени и в пространстве». Переменная – способ обобщить, входит в факт с квантором всеобщности (т.е. любой элемент из множества).

*Что собой представляет программа на Prolog, какова ее структура, как она реализуется.*

Программа на Prolog представляет собой базу знаний и вопрос. База знаний состоит из предложений – CLAUSES: фактов и правил. Каждое предложение заканчивается точкой. Вопрос является составным термом. Система рассматривает вопрос как цель, к которой надо стремиться.

Программа на Prolog состоит из разделов, который начинается со своего заголовка.

- Директивы компилятора – зарезервированные символьные константы.
- CONSTANTS – раздел описания констант.
- DOMAINS – раздел описания доменов.
- DATABASE – раздел описания предикатов внутренней базы данных.
- PREDICATES – раздел описания предикатов.
- CLAUSES – раздел описания предложений базы знаний.
- GOAL – раздел описания внутренней цели.

В процессе выполнения программы – система, используя встроенный алгоритм унификации, пытается обосновать возможность истинности вопрос, строя подстановки и примеры термов. Алгоритм унификации автоматически и многократно запускается системой.

При запуске алгоритма унифицирования в стек помещается равенство  $B = A\theta$ . Затем из стека считывается рабочее поле и обрабатывается 1 равенство. Цикл завершается, если стек пуст или поступило сообщение об отказе. Дополнительно используется переменная «неудача». В результате работы алгоритма унификации результирующая подстановка будет располагаться в результирующей ячейке памяти.

### Часть 3.

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

- «Телефонный справочник»: фамилия, телефон, адрес – структура (город, улица, № дома, № квартиры),
- Автомобили: фамилия, марка, цвет, стоимость и др.,
- Вкладчики банков: фамилия, банк, счет, сумма, др.

Владелец может иметь несколько телефонов, автомобилей, вкладов.

Используя правила, обеспечить возможность поиска:

1. А) По № телефона найти: фамилию, марку автомобиля, стоимость автомобиля  
Б) Используя правило а), по № телефона найти марку автомобиля.
2. Используя простой вопрос по фамилии и городу найти: улицу, банки и № телефона.



### Листинг 3.

```
domains
    name, phone, city, street = string.
    model, color = string.
    bank = string.

    house, flat = integer.
    cost, year = integer.
    account, amount, cell = integer.

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

predicates
    phoneRecord(name, phone, address).
    carRecord(name, model, color, cost, year).
    bankRecord(name, bank, account, amount, cell).

    getCarByPhone(phone, name, model, cost).
    getCarModelByPhone(phone, model).
    getStreetBankPhoneByName(name, city, street, bank, phone).

clauses
    phoneRecord("Ivanov", "+7910284", address("Moscow", "Berezovaya", 4, 23)).
    phoneRecord("Petrov", "+7902873", address("Riga", "Grushnevaya", 36, 109)).
    phoneRecord("Sidorov", "+7902873", address("Kazan", "Dryzby", 84, 91)).
    phoneRecord("Fedotov", "+7918276", address("Piter", "Smolenskaya", 62, 41)).

    carRecord("Ivanov", "Volvo", "blue", 891000, 2009).
    carRecord("Ivanov", "Suzuki", "black", 919000, 2017).
    carRecord("Petrov", "BMW", "white", 1000000, 2010).
    carRecord("Fedotov", "Lambargini", "red", 5901000, 2021).

    bankRecord("Petrov", "Sber", 8019, 190276, 346).
    bankRecord("Sidorov", "Tinkoff", 0923, 19872, 901).
    bankRecord("Fedotov", "Raif", 8801, 80238, 19).
    bankRecord("Fedotov", "Russ", 1874, 901238, 1984).

    getCarByPhone(Phone, Name, Model, Cost) :- phoneRecord(Name, Phone, _),
                                                carRecord(Name, Model, _, Cost, _).
    getCarModelByPhone(Phone, Model) :- getCarByPhone(Phone, _, Model, _).

    getStreetBankPhoneByName(Name, City, Street, Bank, Phone) :-
        phoneRecord(Name, Phone, address(City, Street, _, _)),
        bankRecord(Name, Bank, _, _, _).
```


1. По № телефона найти: фамилию, марку автомобиля, стоимость автомобиля.

```
getCarByPhone(Phone, Name, Model, Cost) :- phoneRecord(Name, Phone, _),
                                            carRecord(Name, Model, _, Cost, _).
getCarModelByPhone(Phone, Model) :- getCarByPhone(Phone, _, Model, _).

getStreetBankPhoneByName(Name, City, Street, Bank, Phone) :-
    phoneRecord(Name, Phone, address(City, Street, _, _)),
    bankRecord(Name, Bank, _, _, _).
```

goal

```
getCarByPhone("+7910284", Name, Model, Cost).
```

 [Inactive C:\VIP52\DOC\EXAMPLES\TestGoal\Obj\goal\$000.exe]

**Name=Ivanov, Model=Volvo, Cost=891000**

**Name=Ivanov, Model=Suzuki, Cost=919000**

**2 Solutions**

2. Используя правило а), по № телефона найти марку автомобиля.

```
getCarByPhone(Phone, Name, Model, Cost) :- phoneRecord(Name, Phone, _),
                                         carRecord(Name, Model, _, Cost, _).
getCarModelByPhone(Phone, Model) :- getCarByPhone(Phone, _, Model, _).

getStreetBankPhoneByName(Name, City, Street, Bank, Phone) :-
    phoneRecord(Name, Phone, address(City, Street, _, _)),
    bankRecord(Name, Bank, _, _, _).

goal
%getCarByPhone("+7910284", Name, Model, Cost).
getCarModelByPhone("+7910284", Model).

[Inactive C:\VIP52\DOC\EXAMPLES\TestGoal\Obj\goal$000.exe]
Model=Volvo
Model=Suzuki
2 Solutions
```

3. Используя простой вопрос по фамилии и городу найти: улицу, банки и № телефона.

```
getCarByPhone(Phone, Name, Model, Cost) :- phoneRecord(Name, Phone, _),
                                         carRecord(Name, Model, _, Cost, _).
getCarModelByPhone(Phone, Model) :- getCarByPhone(Phone, _, Model, _).

getStreetBankPhoneByName(Name, City, Street, Bank, Phone) :-
    phoneRecord(Name, Phone, address(City, Street, _, _)),
    bankRecord(Name, Bank, _, _, _).

goal
%getCarByPhone("+7910284", Name, Model, Cost).
%getCarModelByPhone("+7910284", Model).

getStreetBankPhoneByName("Fedotov", City, Street, Bank, Phone).

[Inactive C:\VIP52\DOC\EXAMPLES\TestGoal\Obj\goal$000.exe]
City=Piter, Street=Smolenskaya, Bank=Raiif, Phone=+7918276
City=Piter, Street=Smolenskaya, Bank=Russ, Phone=+7918276
2 Solutions
```

### Вывод 3.

1. Что такое терм?

Терм – основной элемент языка.

а. Константа

- i. Число (целое, вещественное),
- ii. Символьный атом (комбинация букв, цифр и \_),

- iii. Строка (последовательность символов в кавычках).
- b. Переменная
  - i. Именованная (начинается с заглавной буквы или `_`),
  - ii. Анонимная (символ `_`).
- c. Составной терм ( $f(t_1, \dots, t_m)$ ), где  $f$  – функтор,  $t_1, \dots, t_m$  – термы).

## 2. Что такое предикат в матлогике?

Предикат – это высказывание, в которое можно подставлять аргументы 0 (ложь) или 1 (истина).

## 3. Что описывает предикат в Prolog?

Предикат в Prolog – утверждение программы. Описывает отношение сущностей базы знаний.

## 4. Назовите виды предложения в программе и приведите примеры таких предложений вашей программы.

Предложение имеет вид:  $A :- B_1, \dots, B_n$ .  $A$  – заголовок,  $B_1, \dots, B_n$  – тело. Предложения бывают двух видов: факты (частный случай правила, тело пустое) и правила (предложения более общего вида).

Пример факта: `phoneRecord("Ivanov", "+7910284", address("Moscow", "Berezovaya", 4, 23)).`

Пример правила: `getCarModelByPhone(Phone, Model) :- getCarByPhone(Phone, _, Model, _).`

*Какие предложения являются основными, а какие неосновными? Каковы синтаксис и семантика этих предложений?*

Основными называются предложения, которые не содержат переменных. Неосновные в момент фиксации в программе могут содержать переменные.

*5. Каковы назначение, виды и особенности использования переменных в программе на Prolog?*

Вообще – переменные предназначены для передачи значений «во времени и пространстве». Переменные обозначают некоторый неизвестный объект из некоторого множества объектов. Переменные в факты и правила входят только с квантором всеобщности, в вопросы – только с квантором существования. Факты содержат переменные, которые записываются с большой буквы или с \_. Переменная – способ обобщить, входит в факт с квантором всеобщности (т.е. любой элемент из множества).

*Какое предложение БЗ сформулировано в более общей – абстрактной форме: содержащее или не содержащее переменных?*

Содержащее переменные, т.к. чем больше переменных, тем более общая форма.

*6. Что такое подстановка?*

Подстановкой называется множество пар, вида  $\{X_i = t_i\}$ , где  $X_i$  – переменная, а  $t_i$  – терм. Применение подстановки заключается в замене каждого вхождения переменной  $X_i$  на соответствующий терм.

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

Терм В называется примером терма А, если существует такая подстановка  $\Theta$ , что  $B = \Theta A$ . Строится во время работы алгоритма унификации. Пример терма хранится до окончания работы программы.

1. а) `getCarByPhone("+7910284", Name, Model, Cost)`.

№ шага	Сравниваемые термы; результат; подстановка	Дальнейшие действия
1	<code>getCarByPhone("+7910284", Name, Model, Cost).</code> <code>getCarByPhone(phone, Name, Model, Cost).</code>  Связываются: <code>Phone = "+7910284"</code>	<code>getCarByPhone()</code> – удаляется из стека  <code>phoneRecord(Name, Phone, _),</code> <code>carRecord(Name, Model, _, Cost, _)</code> – помещаются в стек.
2	<code>phoneRecord(Name, "+7910284", _).</code> <code>phoneRecord("Ivanov", "+7910284",</code> <code>address("Moscow", "Berezovaya", 4, 23)).</code>  Связываются: <code>Name = "Ivanov"</code>	<code>phoneRecord(Name, Phone, _)</code> – удаляется из стека
3	<code>carRecord("Ivanov", Model, _, Cost, _).</code> <code>carRecord("Ivanov", "Volvo", "blue", 891000, 2009).</code>  Связываются: <code>Model = "Volvo", Cost = 8910000</code>	<code>carRecord(Name, Model, _, Cost, _)</code> – удаляется из стека.  Выводятся «Ivanov», «Volvo», 8910000  <code>carRecord(Name, Model, _, Cost, _)</code> – помещается в стек.
4	<code>carRecord("Ivanov", Model, _, Cost, _).</code> <code>carRecord("Ivanov", "Suzuki", "black", 919000, 2017).</code>  Связываются: <code>Model = "Suzuki", Cost = 919000</code>	<code>carRecord(Name, Model, _, Cost, _)</code> – удаляется из стека.  Выводятся «Ivanov», «Suzuki», 919000  <code>carRecord(Name, Model, _, Cost, _)</code> – помещается в стек.

5	carRecord("Ivanov", Model, _, Cost, _). carRecord("Petrov", "BMW", "white", 1000000, 2010).  Не унифицируемые.	
6	carRecord("Ivanov", Model, _, Cost, _). carRecord("Fedotov", "Lambargini", "red", 5901000, 2021).  Не унифицируемые.	Откат.  Все факты carRecord пройдены. В стек помещается последний вопрос phoneRecord(Name, Phone, _).  Name развязывается.
7	Остальные факты phoneRecord не унифицируемые.	Откат.  Все факты phoneRecord пройдены. Последний вопрос getCarByPhone() – помещается в стек.  Phone развязывается.

1. b) **getCarModelByPhone("+7910284", Model).**

№ шага	Сравниваемые термы; результат; подстановка	Дальнейшие действия
1	getCarModelByPhone("+7910284", Model). getCarModelByPhone(Phone, Model)  Связываются: Phone = "+7910284"	getCarModelByPhone() – удаляется из стека  getCarByPhone(Phone, _, Model, _) - помещается в стек.
2	getCarByPhone("+7910284", Name, Model, Cost). getCarByPhone(phone, Name, Model, Cost).  Связываются: Phone = "+7910284"	phoneRecord(Name, Phone, _) – удаляется из стека

3	<p>phoneRecord(Name, "+7910284", _).</p> <p>phoneRecord("Ivanov", "+7910284", address("Moscow", "Berezovaya", 4, 23)).</p> <p>Связываются: Name = "Ivanov"</p>	<p>phoneRecord(Name, Phone, _) – удаляется из стека</p>
4	<p>carRecord("Ivanov", Model, _, Cost, _).</p> <p>carRecord("Ivanov", "Volvo", "blue", 891000, 2009).</p> <p>Связываются: Model = "Volvo", Cost = 8910000</p>	<p>carRecord(Name, Model, _, Cost, _) – удаляется из стека.</p> <p>Выводится «Volvo»</p> <p>carRecord(Name, Model, _, Cost, _) – помещается в стек.</p>
5	<p>carRecord("Ivanov", Model, _, Cost, _).</p> <p>carRecord("Ivanov", "Suzuki", "black", 919000, 2017).</p> <p>Связываются: Model = "Suzuki", Cost = 919000</p>	<p>carRecord(Name, Model, _, Cost, _) – удаляется из стека.</p> <p>Выводятся «Suzuki»</p> <p>carRecord(Name, Model, _, Cost, _) – помещается в стек.</p>
6	<p>carRecord("Ivanov", Model, _, Cost, _).</p> <p>carRecord("Petrov", "BMW", "white", 1000000, 2010).</p> <p>Не унифицируемые.</p>	
7	<p>carRecord("Ivanov", Model, _, Cost, _).</p> <p>carRecord("Fedotov", "Lambargini", "red", 5901000, 2021).</p> <p>Не унифицируемые.</p>	<p>Откат.</p> <p>Все факты carRecord пройдены. В стек помещается последний вопрос phoneRecord(Name, Phone, _).</p> <p>Name развязывается.</p>
8	<p>Остальные факты phoneRecord не унифицируемые.</p>	<p>Откат.</p>

		<p>Все факты phoneRecord пройдены.</p> <p>Последний вопрос getCarByPhone() – помещается в стек.</p> <p>Phone развязывается.</p>
9		<p>Откат.</p> <p>getCarByPhone – удаляется из стека</p> <p>Последний вопрос getCarModelByPhone (Phone, _, Model, _) - помещается в стек.</p>

## 2. getStreetBankPhoneByName("Fedotov", City, Street, Bank, Phone).

№ шага	Сравниваемые термы; результат; подстановка	Дальнейшие действия
1	<p>getStreetBankPhoneByName("Fedotov", City, Street, Bank, Phone).</p> <p>getStreetBankPhoneByName(Name, City, Street, Bank, Phone).</p> <p>Связываются: Name = "Fedotov"</p>	<p>getStreetBankPhoneByName () – удаляется из стека</p> <p>phoneRecord(Name, Phone, address(City, Street, _, _)),</p> <p>bankRecord(Name, Bank, _, _, _) – помещаются в стек.</p>
2	<p>phoneRecord("Fedotov", Phone, address(City, Street, _, _)),</p> <p>phoneRecord("Fedotov", "+7918276", address("Piter", "Smolenskaya", 62, 41)).</p> <p>Связываются: Phone = "+7918276", City = "Piter", Street = "Smolenskaya"</p>	<p>phoneRecord(Name, Phone, address(City, Street, _, _)) – удаляется из стека.</p>
3	<p>bankRecord("Fedotov", Bank, _, _, _).</p> <p>bankRecord("Fedotov", "Raif", 8801, 80238, 19).</p>	<p>bankRecord(Name, Bank, _, _, _) – удаляется из стека.</p>



	Связываются: Bank = "Raif"	<p>Выводятся "Piter", "Smolenskaya", "Raif", "+7918276"</p> <p>bankRecord(Name, Bank, _, _, _) – помещается в стек.</p>
4	<p>bankRecord("Fedotov", Bank, _, _, _).</p> <p>bankRecord("Fedotov", "Russ", 1874, 901238, 1984).</p> <p>Связываются: Bank = "Russ"</p>	<p>bankRecord(Name, Bank, _, _, _) – удаляется из стека.</p> <p>Выводятся "Piter", "Smolenskaya", "Russ", "+7918276"</p>
5		<p>Откат.</p> <p>Все факты bankRecord пройдены. В стек помещается последний вопрос phoneRecord(Name, Phone, address(City, Street, _, _)).</p>
6	Остальные факты phoneRecord не унифицируемые.	<p>Откат.</p> <p>Все факты phoneRecord пройдены. Последний вопрос getStreetBankPhoneByName() – помещается в стек.</p> <p>Name развязывается.</p>