



**«Московский государственный технический  
университет имени Н.Э. Баумана»  
(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»**

Студент группы ИУ7-63Б: Фурдик Н. О.  
(Фамилия И.О.)

Преподаватель: Толпинская Н. Б., Строганов Ю. В.  
(Фамилия И.О.)

# Оглавление

Задание . . . . .	2
Листинг программы . . . . .	3
Описание порядка поиска ответов . . . . .	4
Ответы на вопросы . . . . .	8
Список литературы . . . . .	11

## Задание

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

- 1) «Телефонный справочник»: Фамилия, №тел, Адрес – структура (Город, Улица, №дома, №кв);
- 2) «Автомобили»: Фамилия владельца, Марка, Цвет, Стоимость, и др;
- 3) «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),  
б) 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 б), 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.

## Листинг программы

Ниже представлен листинг программы:

```
domains

addr = ad(symbol town, symbol street, integer building, integer apartment)

predicates

person(symbol l_name, symbol ph_number, addr)
car(symbol owner_lname, symbol brand, symbol color, integer cost, integer year, symbol town)
bank_client(symbol client_lname, symbol bank_name, symbol type, integer amount_of_money, integer behaviour, symbol town)

get_cars_by_number(symbol ph_num, symbol car_brand, integer car_cost)
get_stuff(symbol lname, symbol town, symbol street, symbol banks, symbol tel_num)

clauses

person("klyuge", "+7912571026", ad("volgograd", "varshavshkoe sh.", 52, 83)).
person("toporova", "+79165072034", ad("saint—petersburg", "varshavshkoe sh.", 100, 1151)).
person("zayceva", "+79138767153", ad("kasimov", "lermontovskiy pr.", 83, 578)).
person("mishkina", "+79165329056", ad("moscow", "antonova", 53, 1294)).
person("klyuge", "+79127675454", ad("moscow", "venevskaya", 85, 947)).
person("nikitin", "+7913336524", ad("kiev", "pushkina", 86, 973)).
person("nikitin", "+79146106984", ad("ryazan", "varshavshkoe sh.", 72, 27)).
person("toporov", "+7913733594", ad("saint—petersburg", "varshavshkoe sh.", 47, 1321)).
person("zaycev", "+7912486657", ad("volgograd", "lermontovskiy pr.", 86, 668)).
person("mishkin", "+79154045900", ad("kiev", "pushkina", 25, 278)).

car("zayceva", "geely", "white", 6409914, 2015, "kasimov").
car("nikitin", "mitsubishi", "green", 42502483, 2011, "ryazan").
car("klyuge", "mercedes", "blue", 19367655, 2013, "moscow").
car("toporov", "geely", "black", 59107666, 2011, "saint—petersburg").
car("toporova", "mitsubishi", "green", 12836628, 2018, "saint—petersburg").
car("mishkin", "mercedes", "black", 33637427, 2015, "kiev").
car("mishkin", "mitsubishi", "green", 3009291, 2011, "kiev").
car("zayceva", "ford", "blue", 36681813, 2012, "kasimov").
car("nikitin", "geely", "black", 58439837, 2011, "ryazan").
car("zaycev", "geely", "blue", 61712885, 2011, "kiev").

bank_client("mishkina", "tinkoff", "payment", 2439027, 8, "moscow").
bank_client("konev", "sberbank", "payment", 63838628, 4, "kasimov").
bank_client("toporova", "tinkoff", "savings", 60815175, 9, "saint—petersburg").
bank_client("klyuge", "tinkoff", "savings", 1768099, 9, "moscow").
bank_client("zayceva", "tinkoff", "savings", 53406609, 0, "kasimov").
bank_client("nikitin", "alphabank", "savings", 14555497, 8, "kiev").
bank_client("klyuge", "tinkoff", "credit", 34218309, —1, "volgograd").
bank_client("toporova", "sberbank", "credit", 37214889, 10, "saint—petersburg").
bank_client("toporova", "sberbank", "savings", 51895925, 10, "saint—petersburg").
bank_client("nikitin", "sberbank", "payment", 3318136, 8, "toporova").

get_cars_by_number(Ph_num, Car_brand, Car_cost) :-person(L_name, Ph_num, ad(Town, _, _, _)),
```

```

car(L_name, Car_brand, _, Car_cost, _, Town).

get_stuff(Lname, Town, Street, Bank, Tel_num) :- person(Lname, Tel_num, ad(Town, Street, _, _)),
    bank_client(Lname, Bank, _, _, Town).

goal

%brand & cost
get_cars_by_number(Ph_number, Car_brand, Car_cost), Ph_number = "+79154045900", nl.

%Ph_number=+79154045900, Car_brand=mercedes, Car_cost=33637427
%Ph_number=+79154045900, Car_brand=mitsubishi, Car_cost=3009291
%2 Solutions

%brand
get_cars_by_number(Ph_number, Car_brand, _), Ph_number = "+79154045900", nl.

%Ph_number=+79154045900, Car_brand=mercedes
%Ph_number=+79154045900, Car_brand=mitsubishi
%2 Solutions

%street, banks, tel_num
get_stuff(Lname, Town, Street, Bank, Tel_num), Lname = "toporova", Town = "saint-petersburg", nl.

%Lname=toporova, Town=saint-petersburg, Street=varshavshkoe sh., Bank=tinkoff, Tel_num=+79165072034
%Lname=toporova, Town=saint-petersburg, Street=varshavshkoe sh., Bank=sberbank, Tel_num=+79165072034
%Lname=toporova, Town=saint-petersburg, Street=varshavshkoe sh., Bank=sberbank, Tel_num=+79165072034
%3 Solutions

```

Листинг 1: Задания 1 и 2

## Описание порядка поиска ответов

Таблица 1: Вывод Ph\_number=+79154045900, Car\_brand=mercedes, Car\_cost=33637427

№ ша- га	Сравниваемые термы; подстановка, если есть	Результат	Дальнейшие действия: прямой ход или откат (к чему приводит?)
1	Подстановка Ph_number = '+79154045900' в запрос get_cars_by_number(Ph_number, Car_brand, Car_cost)	get_cars_by_number('+79154045900', Car_brand, Car_cost)	прямой ход
2	Поиск совпадений и подстановка '+79154045900', Car_brand, Car_cost в соответствующее правило get_cars_by_number(Ph_num, Car_brand, Car_cost)	get_cars_by_number('+79154045900', Car_brand, Car_cost)	прямой ход
3	Подстановка Ph_num = '+79154045900' в person(L_name, Ph_num, ad(Town, _, _, _))	Создается пример person(L_name, '+79154045900', ad(Town, _, _, _))	прямой ход
4	Поиск person по примеру	person('klyuge', '+7912571026', ad('volgograd', 'varshavshkoe sh.', 52, 83))., не подходит.	откат
5	Поиск person по примеру	...	откат
6	Поиск person по примеру	person('mishkin', '+79154045900', ad('kiev', 'pushkina', 25, 278)), L_name = 'mishkin', Town = 'kiev'	прямой ход
7	Подстановка в car(L_name, Car_brand, _, Car_cost, _, Town) L_name = 'mishkin', Car_brand = Car_brand, Car_cost, Town = 'kiev'	Создается пример car('mishkin', Car_brand, _, Car_cost, _, Town, 'kiev')	прямой ход
8	Поиск car по примеру	car('zayceva', 'geely', 'white', 6409914, 2015, 'kasimov'), не подходит	откат
9	Поиск car по примеру	...	откат
10	Поиск car по примеру	car('mishkin', 'mercedes', 'black', 33637427, 2015, 'kiev'), Car_brand = 'mercedes', Car_cost = 33637427	прямой ход
11	Общий пример найден, вывод	Ph_num = '+79154045900', Car_brand = 'mercedes', Car_cost = 33637427	

Таблица 2: Вывод Ph\_number=+79154045900, Car\_brand=mercedes

№ ша-га	Сравниваемые термы; подстановка, если есть	Результат	Дальнейшие действия: прямой ход или откат (к чему приводит?)
1	Подстановка Ph_number = '+79154045900' в запрос get_cars_by_number(Ph_number, Car_brand, _)	get_cars_by_number('+79154045900', Car_brand, _)	прямой ход
2	Поиск совпадений и подстановка '+79154045900', Car_brand в соответствующее правило get_cars_by_number(Ph_num, Car_brand, _)	get_cars_by_number('+79154045900', Car_brand, _)	прямой ход
3	Подстановка Ph_num = '+79154045900' в person(L_name, Ph_num, ad(Town, _, _, _))	Создается пример person(L_name, '+79154045900', ad(Town, _, _, _))	прямой ход
4	Поиск person по примеру	person('klyuge', '+7912571026', ad('volgograd', 'varshavshkoe sh.', 52, 83)), не подходит.	откат
5	Поиск person по примеру	...	откат
6	Поиск person по примеру	person('mishkin', '+79154045900', ad('kiev', 'pushkina', 25, 278)), L_name = 'mishkin', Town = 'kiev'	прямой ход
7	Подстановка в car(L_name, Car_brand, _, _, Town) L_name = 'mishkin', Car_brand = Car_brand, Car_cost, Town = 'kiev'	Создается пример car('mishkin', Car_brand, _, _, Town, 'kiev')	прямой ход
8	Поиск car по примеру	car('zayceva', 'geely', 'white', 6409914, 2015, 'kasimov'), не подходит	откат
9	Поиск car по примеру	...	откат
10	Поиск car по примеру	car('mishkin', 'mercedes', 'black', 33637427, 2015, 'kiev'), Car_brand = 'mercedes'	прямой ход
11	Общий пример найден, вывод	Ph_num = '+79154045900', Car_brand = "mercedes"	

Таблица 3: Вывод Lname=toporova, Town=saint-petersburg, Street=varshavshkoe sh., Bank=tinkoff, Tel\_num=+79165072034

№ ша- га	Сравниваемые термы; подста- новка, если есть	Результат	Дальнейшие действия: прямой ход или откат (к чему приво- дит?)
1	Подстановка Lname = 'toporova' и Town = 'saint-petersburg' в запрос get_stuff(Lname, Town, Street, Bank, Tel_num)	get_stuff('toporova', 'saint- petersburg', Street, Bank, Tel_num)	прямой ход
2	Поиск совпадений и подстанов- ка Lname = 'toporova', Town = 'saint-petersburg' Street = Street, Bank = Bank, Tel_num = Tel_num в соответствен- ное правило get_stuff(Lname, Town, Street, Bank, Tel_num)	('toporova', 'saint-petersburg', Street, Bank, Tel_num)	прямой ход
3	Подстановка Lname = 'toporova', Town = 'saint- petersburg' в person(L_name, Tel_num, ad(Town, Street, _, _))	Создается пример person('toporova', Tel_num, ad('saint-petersburg', Street, _, _))	прямой ход
4	Поиск person по примеру	person('klyuge', '+7912571026', ad('volgograd', 'varshavshkoe sh.', 52, 83)), не подходит	откат
5	Поиск person по примеру	...	откат
6	Поиск person по примеру	person('toporova', '+79165072034', ad('saint- petersburg', 'varshavshkoe sh.', 100, 1151)), L_name = 'toporova', Tel_num = '+79165072034', Town = 'saint-petersburg', Street = 'varshavshkoe sh.'	прямой ход
7	Подстановка в bank_client(Lname, Bank, _, _, _, Town) L_name = 'toporova', Bank = Bank, Town = 'saint-petersburg'	Создается пример bank_client('toporova', Bank, _, _, _, 'saint-petersburg')	прямой ход



8	Поиск bank_client по примеру	bank_client('mishkina', 'tinkoff', 'payment', 2439027, 8, 'moscow'), не подходит	откат
9	Поиск bank_client по примеру	...	откат
10	Поиск bank_client по примеру	bank_client('toporova', 'tinkoff', 'savings', 60815175, 9, 'saint-petersburg'), Bank = 'tinkoff'	прямой ход
11	Общий пример найден, вывод	Lname=toporova, Town=saint-petersburg, Street=varshavshkoe sh., Bank=tinkoff, Tel_num=+79165072034	

## Ответы на вопросы

### 1) Что такое терм?

Терм - основная конструкция Prolog. Термы бывают простыми (константы (атомы и числа) и переменные) и составными. Обработываются слева направо.

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

Определенным на множествах  $M_1, M_2, \dots, M_n$   $n$ -местным предикатом называется предложение, содержащее  $n$  переменных  $x_1, x_2, \dots, x_n$ , превращающееся в высказывание при подстановке вместо этих переменных любых конкретных элементов из множеств  $M_1, M_2, \dots, M_n$  соответственно.

### 3) Что описывает предикат в Prolog?

В Prolog предикат - это утверждение, истинность которого зависит от значения переменных, входящих в него. Синтаксис:

3.1. :- означает if;

3.2. , означает and;

3.3. ; означает or;

3.4. . означает end.

### 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)?**

Виды предложений:

4.1. Факт - предложение, которое устанавливает безусловно-истинное отношение между термами, или утверждает некоторую безусловную истину.

Пример:

```
person("klyuge", "+7912571026", ad("volgograd", "varshavskoe sh.", 52, 83)).
```

4.2. Правило - предложение, которое устанавливает отношение между термами при выполнении заданных условий. Пример:

```
get_cars_by_number(Ph_num, Car_brand, Car_cost) :- person(L_name, Ph_num,  
ad(Town, _, _, _)), (L_name, Car_brand, _, Car_cost, _, Town).
```

4.3. Запрос - предложение, которое устанавливает совпадения с фактами или правилами БЗ. Пример:

```
get_cars_by_number(Ph_number, Car_brand, _), Ph_number = "+79154045900", nl.
```

Факты являются основными предложениями в Prolog.

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 начинаются с заглавной буквы или с символа подчёркивания и могут быть анонимными или именованными (анонимная переменная обозначается нижним подчеркиванием). Говорят, что переменная может быть связана с некоторым значением или оставаться независимой. Именованные переменные уникальны в рамках одного предложения. Анонимная переменная уникальна всегда. Переменные предназначены для передачи значений «во времени и в пространстве».

Предложение БЗ сформулировано в более общей абстрактной форме, если оно содержит переменные.

6) **Что такое подстановка?**

Подстановка - множество пар вида  $x_i = t_i$ , где

6.1.  $x_i$  - переменная;

6.2.  $t_i$  - терм.

Если существует  $A(x_1, x_2, \dots, x_n)$  и подстановка  $\theta = x_1 = t_1, x_2 = t_2, \dots, x_n = t_n$  то применение подстановки заключается в замене  $x_i$  на  $t_i$ .

**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 такая, что  $B = A\theta$ .

Терм  $C$  является общим примером  $A, B$  если существуют подстановки  $\theta_1$  и  $\theta_2$  такие что  $C = A\theta_1$  и  $C = B\theta_2$ .

В процессе выполнения программы — система, используя встроенный алгоритм унификации, пытается обосновать возможность истинности вопроса, строя подстановки и примеры термов (вопроса и формулировки знания), используя базу знаний, и найти такие значения переменных, при которых это удастся, а значит, на поставленный вопрос можно дать ответ «Да». Возможно система «ошибается» в своих обоснованиях и возникает тупиковая ситуация, или, ответив на вопрос, пытается найти другой способ доказательства. Тогда включается механизм отката (отказа от последнего заключения (какого?) и последних действий, сделанных системой) и выполняется ре- конкретизация переменных, конкретизация которых была выполнена на последнем шаге.

# Литература

1. Толпинская Н.Б. - Курс лекций по "Функциональному и Логическому программированию"[Текст], Москва 2019 год.
2. Анатолий Адаменко, Андрей Кучуков. Логическое программирование и Visual Prolog (с CD). — СПб.: БХВ-Петербург, 2003. — 990 с. — ISBN 5-94157-156-9.