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» (ИУ7)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.04 Программная инженерия**

О Т Ч Е Т

по лабораторной работе № 1 1 / 1 2 / 1 3

Дисциплина: Функциональное и логическое программирование

Студент ИУ7-62Б (Группа) _____ (Подпись, дата) Е.В. Брянская (И.О. Фамилия)

Преподаватель _____ (Подпись, дата) Н.Б.Толпинская, Ю.В.Строганов (И.О. Фамилия)

Москва, 2021

Л/р 11

Задание

Составить программу, с помощью которой можно определить, например, множество студентов, обучающихся в одном вузе.

```
DOMAINS
        name, surname, subj, unv = symbol.

PREDICATES
        student(name, surname, unv).
        teacher(name, surname, subj, unv).

CLAUSES
        student("Ann", "Richy", "unv_1").
        student("Boby", "Adamson", "unv_3").
        student("Galya", "Backer", "unv_2").
        student("Gigi", "Dyson", "unv_1").

        student("Tanya", "Park", Unv):-student("Ann", "Richy", Unv).

        teacher("Kristina", "Nikiforova", "physics", "unv_2").
        teacher("Mila", "Kolovanova", "english", "unv_2").
        teacher("Anton", "Popov", "drawing", "unv_4").

GOAL
        %student("Boby", "Adamson", "unv_3").

        %student("Galya", Surname, "unv_2").

        %teacher(_, _, _, "unv_2").

        %teacher(Name, Surname, Subj, "unv_2").

        %teacher(Name, Surname, Subj, "unv_100").
```

1	student("Boby", "Adamson", "unv_3").	yes
2	student("Galya", Surname, "unv_2").	Surname=Backer 1 Solution
3	teacher(_, _, _, "unv_2").	yes
4	teacher(Name, Surname, Subj, "unv_2").	Name=Kristina, Surname=Nikiforova, Subj=physics Name=Mila, Surname=Kolovanova, Subj=english 2 Solutions
5	teacher(Name, Surname, Subj, "unv_100").	No Solution

Л/р 12

Задание

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 (не менее 3-х). Описать порядок формирования вариантов ответа.

```
DOMAINS
        name, surname, subj, unv = symbol.

PREDICATES
        student(name, surname, unv).
        teacher(name, surname, subj, unv).

CLAUSES
        student("Ann", "Richy", "unv_1").
        student("Boby", "Adamson", "unv_3").
        student("Boby", "Dicson", "unv_1").
        student("Galya", "Backer", "unv_2").
        student("Gigi", "Dyson", "unv_1").
        student("Gigi", "Dyson", "unv_2").
        student("Gigi", "Dyson", "unv_3").

        student("Tanya", "Park", Unv):-student("Ann", "Richy", Unv).

        teacher("Kristina", "Nikiforova", "physics", "unv_2").
        teacher("Mila", "Kolovanova", "english", "unv_2").
        teacher("Anton", "Popov", "drawing", "unv_4").

GOAL
        %student(Name, Surname, "unv_1").

        %student("Boby", Surname, "unv_2").

        %teacher(_, _, "english", "unv_2").

        %teacher(Name, Surname, Subj, "unv_4").

        %teacher("Mila", "Kolovanova", "english", "unv_2").
```

1	student(Name, Surname, "unv_1").	Система сравнивает весь вопрос с первым фактом. В вопросе две переменные, система свяжет их с соответствующими значениями знания (при условии, что природа аргументов одинаковая) и ответит на поставленный вопрос, если ответ да, то в качестве побочного эффекта на экран будут выведены значения переменных. Чтобы продолжить поиск система отменяет это решение (то есть переменная теряет своё значение), и далее продолжается поиск решения.	**Name=Ann, Surname=Richy** **Name=Gigi, Surname=Dyson** **Name=Tanya, Surname=Park** **3 Solutions**

2	student("Boby", Surname, "unv_2").	Аналогично, только в вопросе только одна переменная и две константы.	No Solution
3	teacher(_, _, "english", "unv_2").	Аналогично, только разница в том, что не столько важно значение анонимных переменных, сколько важно количество аргументов	yes
4	teacher(Name, Surname, Subj, "unv_4").	Аналогично вопросу №1/2	Name=Anton, Surname=Popov, Subj=drawing 1 Solution
5	teacher("Mila", "Kolovanova", "english", "unv_2").	В вопросе нет переменных, поэтому проверяется только совпадение констант.	yes

Вопросы

1. **Что представляет из себя программа на Prolog?**

Программа на Prolog состоит из базы знаний, которую составляют факты и правила.

2. **Структура программы на Prolog?**

Программа на Prolog состоит из разделов, которые начинаются с заголовка.

Разделы:

- 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 (вопроса)

В программе могут быть не все разделы.

3. **Как реализуется программа на Prolog?**

Описывается база знаний (факты и правила), задаётся вопрос.

4. **Как формируются результаты работы программы?**

Система пытается найти среди базы знаний такие значения переменных, чтобы ответить «Да» на поставленный вопрос.

Л/р 13

Задание

Составить программу, то есть 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 т.д.
- «Вкладчики банков»: Фамилия, Банк, счет, сумма, др.

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

Используя правила, обеспечить возможность поиска:

1)
    a. По № телефона найти: Фамилию, Марку автомобиля, Стоимость автомобиля (может быть несколько).
    b. 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 a и для b, 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.

Вопросы

1. **Что такое терм?**

Основной элемент – терм.

Терм:

1) Константа
    a. Число (целое, вещественное)
    b. Символьный атом (комбинация символов латинского алфавита, цифр и _, **начинающаяся со строчной буквы**)

c. Строка (последовательность символов, заключенных в “ “)

2) Переменная
   a. Именованная (комбинация символов латинского алфавита, цифр, **начинающаяся с прописной буквы или с _**)
   b. Анонимная (обозначается символом _)
3) Составной терм (средство организации группы отдельных элементов знаний в единый объект). Состоит из функтора (имя отношения) и аргументов, представляющих из себя термы.

2. Что такое предикат в матлогике (математике)?

Предикат в матлогике – это логическая функция, возвращающая либо истину, либо ложь.

3. Что описывает предикат в Prolog?

Предикат в Prolog – отношение, определяемое процедурой, утверждение базы знаний.

Процедура – множество предложений базы знаний, которые определяют одно значение, заголовки которых имеют одинаковые функторы, одинаковое количество аргументов одной природы.

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)?

Факты – частный случай правила, с пустым телом.

Пример: phone_book("Birukova", "+123456", address("Moscow", "Zhukovsky Street", 12, 145)).

Правила – состоят из заголовка (содержится знание) и тела (содержит условие истинности)

Пример: car_by_phone(Phone, Surname, Brand, Price) :-

phone_book(Surname, Phone, _),

car(Surname, Brand, _, Price, _).

Вопрос – состоят только из тела, с их помощью пользователь может задать вопрос системе, и та, используя базу знаний даёт соответствующий ответ.

Пример: brand_by_phone("+123456", Brand).

Предложения, не содержащие переменные, называются основными, и, наоборот, предложения, использующие переменные, называются неосновными.

Синтаксис:

<заголовок> :- <тело правила>.

Семантика основных предложений заключается в том, что формируется безусловная истина, неосновных – условная.

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

Переменные нужны для обобщения, в программе нужны как способ передачи значения во времени и пространстве.

Виды переменных:

- Именованная (комбинация символов латинского алфавита, цифр, **начинающаяся с прописной буквы или с _**) (позволяет передавать значения в пространстве и во времени)
- Анонимная (обозначается символом _) (значение неважно)

Чем больше переменных, тем более общая формулировка знания. Следовательно, предложение, содержащее переменные сформулировано в более абстрактной форме.

6. **Что такое подстановка?**

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

Пусть Θ = $\{x_1 = t_1, \ldots x_n = tn\}$ - подстановка, тогда результат применения подстановки к терму обозначается как AΘ.

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 Θ, что B = A Θ, где A Θ – результат применения подстановки к терму. Строятся в процессе работы алгоритма унификации. Система строит примеры в процессе поиска среди базы знаний такие значения переменных, чтобы ответить «Да» на поставленный вопрос, и хранит их до окончания работы программы.

```
DOMAINS
        surname, phone = symbol.

        city, street = symbol.
        home, flat = integer.

        brand, color = symbol.
        price = real.
        years = integer.

        bank, account = symbol.
        sum = real.

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

PREDICATES
        phone_book(surname, phone, address).

        car(surname, brand, color, price, years).

        investor(surname, bank, account, sum).

        car_by_phone(phone, surname, brand, price).

        brand_by_phone(phone, brand).

        address_by_surname_city(surname, city, street, bank, phone).

CLAUSES
        phone_book("Birukova", "+123456", address("Moscow", "Zhukovsky Street", 12, 145)).
        phone_book("Nikiforov", "+987456", address("Zhukovsky", "Gagarin Street", 64, 32)).
        phone_book("Nikiforov", "+111111", address("Zhukovsky", "Gagarin Street", 64, 32)).
        phone_book("Mironova", "+909090", address("Zhukovsky", "Sunny Street", 2, 89)).
        phone_book("Filin", "+444000", address("Ramenskoe", "Central", 1, 1)).
        phone_book("Mironova", "+333333", address("Ramenskoe", "New Street", 77, 77)).

        car("Birukova", "BMW", black, 4500000, 2).
        car("Birukova", "Ford", white, 6200000, 1).
        car("Filin", "Honda", grey, 2300000, 4).

        investor("Birukova", "New_1", deposit, 1000000).
        investor("Legneva", "Old_bank", special, 2370000).
        investor("Mironova", "Old_bank", deposit, 5000).

        car_by_phone(Phone, Surname, Brand, Price) :-
                phone_book(Surname, Phone, _),
                car(Surname, Brand, _, Price, _).

        brand_by_phone(Phone, Brand) :-
                car_by_phone(Phone, _, Brand, _).

        address_by_surname_city(Surname, City, Street, Bank, Phone) :-
                phone_book(Surname, Phone, address(City, Street, _, _)),
                investor(Surname, Bank, _, _).

GOAL
        %car_by_phone("+123456", Surname, Brand, Price).
        %car_by_phone("+000000", Surname, Brand, Price).
        %car_by_phone("+444000", Surname, Brand, Price).

        %brand_by_phone("+123456", Brand).
        %brand_by_phone("+444000", Brand).
        %brand_by_phone("+******", Brand).

        %address_by_surname_city("Mironova", "Zhukovsky", Street, Bank, Phone).
        %address_by_surname_city("Legneva", "Ramenskoe", Street, Bank, Phone).
```

Результаты работы программы

car_by_phone("+123456", Surname, Brand, Price).	Surname=Birukova, Brand=BMW, Price=4500000 Surname=Birukova, Brand=Ford, Price=6200000 2 Solutions
car_by_phone("+000000", Surname, Brand, Price).	No Solution
car_by_phone("+444000", Surname, Brand, Price).	Surname=Filin, Brand=Honda, Price=2300000 1 Solution
brand_by_phone("+123456", Brand).	Brand=BMW Brand=Ford 2 Solutions
brand_by_phone("+444000", Brand).	Brand=Honda 1 Solution
brand_by_phone("+******", Brand).	No Solution
address_by_surname_city("Mironova ", "Zhukovsky", Street, Bank, Phone).	Street=Sunny Street, Bank=Old_bank, Phone=+909090 1 Solution
address_by_surname_city("Legneva", "Ramenskoe", Street, Bank, Phone).	No Solution

1. **car_by_phone("+123456", Surnamet, Brandt, Pricet).**

№	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
0.		Начальное состояние резольвенты: car_by_phone("+123456", Surnamet, Brandt, Pricet)
1.	car_by_phone("+123456", Surnamet, Brandt, Pricet) = phone_book("Birukova", "+123456", address("Moscow", "Zhukovsky Street", 12, 145))	неудача (не совпали функторы)
2.	* * *	* * *
3.	car_by_phone("+123456", Surnamet, Brandt, Pricet) = car("Birukova", "BMW", black, 4500000, 2)	неудача (не совпали функторы)
4.	* * *	* * *
5.	car_by_phone("+123456", Surnamet, Brandt, Pricet) = investor("Birukova", "New_1", deposit, 1000000)	неудача (не совпали функторы)
6.	* * *	* * *
7.	car_by_phone("+123456", Surnamet, Brandt, Pricet) = car_by_phone(Phone, Surname, Brand, Price) **Подстановка:** {Phone = "+123456", Surname = Surnamet, Brand = Brandt, Price = Pricet}	удача **Изменение резольвенты** Новое состояние резольвенты: phone_book(Surnamet, "+123456", _) car(Surnamet, Brandt, _, Pricet, _)

8.	phone_book(Surnamet, "+123456", _) = phone_book("Birukova", "+123456", address("Moscow", "Zhukovsky Street", 12, 145)) **Подстановка:** {Phone = "+123456", Surname = Surnamet, Brand = Brandt, Price = Pricet, Surnamet = “Birukova”}	удача **Изменение резольвенты** Новое состояние резольвенты: car("Birukova", Brandt, _, Pricet, _)
9.	car("Birukova", Brandt, _, Pricet, _) = car("Birukova", "BMW", black, 4500000, 2) **Подстановка:** {Phone = "+123456", Surname = Surnamet, Brand = Brandt, Price = Pricet, Surnamet = “Birukova”, Brandt = “BMW”, Pricet = 4500000}	удача **Изменение резольвенты** Резольвента пустая Выводится: Surnamet =“Birukova”, Brandt ="BMW", Pricet =4500000 Откат
10.	* * *	* * *

2. brand_by_phone("+123456", Brandt).

№	**Сравниваемые термы; результат; подстановка, если есть**	**Дальнейшие действия: прямой ход или откат (к чему приводит?)**
0.		Начальное состояние резольвенты: brand_by_phone("+123456", Brandt)
1.	brand_by_phone("+123456", Brandt) = phone_book("Birukova", "+123456", address("Moscow", "Zhukovsky Street", 12, 145))	неудача (не совпали функторы)
2.	* * *	* * *
3.	brand_by_phone("+123456", Brandt) = car("Birukova", "BMW", black, 4500000, 2)	неудача (не совпали функторы)
4.	* * *	* * *
5.	brand_by_phone("+123456", Brandt) = investor("Birukova", "New_1", deposit, 1000000)	неудача (не совпали функторы)
6.	* * *	* * *
7.	brand_by_phone("+123456", Brandt) = car_by_phone(Phone, Surname, Brand, Price)	неудача (не совпали функторы)
8.	* * *	* * *
9.	brand_by_phone("+123456", Brandt) = brand_by_phone(Phone, Brand) **Подстановка:** {Phone = “+123456”, Brand = Brandt}	удача **Изменение резольвенты** Новое состояние резольвенты: car_by_phone(“+132456”, _, Brandt, _)
10.	car_by_phone(“+132456”, _, Brandt, _) и phone_book("Birukova", "+123456", address("Moscow", "Zhukovsky Street", 12, 145))	неудача (не совпали функторы)

11.	* * *	* * *
12.	car_by_phone(“+132456”, _, Brandt, _) = car_by_phone(Phone, Surname, Brand, Price) **Подстановка:** {Phone = "+123456", Brand = Brandt, Phone = “+123456”, Brand = Brandt}	удача **Изменение резольвенты** Новое состояние резольвенты: phone_book(Surname, "+123456", _) car(Surname, Brandt, _, Price, _)
13.	phone_book(Surname, "+123456", _) = phone_book("Birukova", "+123456", address("Moscow", "Zhukovsky Street", 12, 145)) **Подстановка:** {Phone = "+123456", Brand = Brandt, Phone = “+123456”, Brand = Brandt , Surname = “Birukova”}	удача **Изменение резольвенты** Новое состояние резольвенты: car("Birukova", Brandt, _, Price, _)
14.	car("Birukova", Brandt, _, Price, _) = phone_book("Birukova", "+123456", address("Moscow", "Zhukovsky Street", 12, 145))	неудача (не совпали функторы)
15.	* * *	* * *
16.	car("Birukova", Brandt, _, Price, _) = car("Birukova", "BMW", black, 4500000, 2) **Подстановка:** {Phone = "+123456", Brand = Brandt, Phone = “+123456”, Brand = Brandt, Surname = “Birukova”, Brandt = “BMW”, Price = 4500000}	удача **Изменение резольвенты** Резольвента пустая Выводится: Brandt ="BMW" Откат
17.	* * *	* * *

3. address_by_surname_city("Mironova", "Zhukovsky", Streett, Bankt, Phonet).

№	**Сравниваемые термы; результат; подстановка, если есть**	**Дальнейшие действия: прямой ход или откат (к чему приводит?)**
0.		Начальное состояние резольвенты: address_by_surname_city("Mironova", "Zhukovsky", Streett, Bankt, Phonet)
1.	address_by_surname_city("Mironova", "Zhukovsky", Streett, Bankt, Phonet) = phone_book("Birukova", "+123456", address("Moscow", "Zhukovsky Street", 12, 145))	неудача (не совпали функторы)
2.	* * *	* * *
3.	address_by_surname_city("Mironova", "Zhukovsky", Streett, Bankt, Phonet) = address_by_surname_city(Surname, City, Street, Bank, Phone) **Подстановка:** {Surname = “Mironova”, City = “Zhukovsky”, Street=Streett, Bank = Bankt, Phone = Phonet}	удача **Изменение резольвенты** Новое состояние резольвенты: phone_book(“Mironova”, Phonet, address(“Zhukovsky”, Streett, _, _)), investor(“Mironova”, Bankt, _, _)
4.	phone_book(“Mironova”, Phonet, address(“Zhukovsky”, Streett, _, _)) =	неудача (не совпали функторы)

	phone_book("Birukova", "+123456", address("Moscow", "Zhukovsky Street", 12, 145))	
5.	* * *	* * *
6.	phone_book(“Mironova”, Phonet, address(“Zhukovsky”, Streett, _, _)) = phone_book("Mironova", "+909090", address("Zhukovsky", "Sunny Street", 2, 89)) **Подстановка:** {Surname = “Mironova”, City = “Zhukovsky”, Street=Streett, Bank = Bankt, Phone = Phonet , Phonet = “+909090”, Streett = "Sunny Street"}	удача **Изменение резольвенты** Новое состояние резольвенты: investor(“Mironova”, Bankt, _, _)
7.	investor(“Mironova”, Bankt, _, _) = phone_book("Birukova", "+123456", address("Moscow", "Zhukovsky Street", 12, 145))	неудача (не совпали функторы)
8.	* * *	* * *
9.	investor(“Mironova”, Bankt, _, _) = investor("Mironova", "Old_bank", deposit, 5000) **Подстановка:** {Surname = “Mironova”, City = “Zhukovsky”, Street=Streett, Bank = Bankt, Phone = Phonet , Phonet = “+909090”, Streett = "Sunny Street", Bankt=”Old_bank”}	удача **Изменение резольвенты** Резольвента пустая Выводится: Streett = "Sunny Street", Brankt ="Old_bank", Phonet = “+909090” Откат
10.	* * *	* * *