



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 № 11

Студент: Зейналов З. Г.

Группа: ИУ7-61Б

Оценка (баллы) _____

Преподаватель: Толпинская Н. Б.

Москва.
2020 г.

Введение

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

Задачи: 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.

Ход работы:

Задание на лабораторную работу: запустить среду Visual Prolog5.2. Настроить утилиту TestGoal (способ настройки см. в дополнительных материалах к лаб. раб.). Запустить тестовую программу, проанализировать реакцию системы и множество ответов. Разработать свою программу – «Телефонный справочник». Протестировать работу программы.

Листинг 1- Тестовая программа

```
predicates  
  
likes(symbol,symbol)  
  
clauses  
  
likes(ellen,tennis).  
  
likes(john,football).  
  
likes(tom,baseball).  
  
likes(eric,swimming).  
  
likes(mark,tennis) .  
  
likes(bill,Activity):-likes (tom, Activity) .  
  
goal  
  
likes(bill, baseball).
```

Утилита Test Goal ответит «yes».

В случае запроса likes(bill, tennis), получим ответ «no».

Листинг 2- код программы-справочника.

```

predicates

    people(symbol, symbol, symbol, symbol).

    getAddress(symbol, symbol, symbol, symbol).

clauses

    people("Ivan", "Ivanov", "912341", "Moscow").

    people("Petr", "Petrov", "212321", "Moscow").

    people("igor", "Igorov", "212142", "Moscow").

    getAddress(Name, Lastname, Phone, Address) :-

        people(Name, Lastname, Phone, Address).

goal

    getAddress("Ivan", Lastname, Phone, Address).

```

```

predicates
    people(symbol, symbol, symbol, symbol).
clauses
    people("Ivan", "Ivanov", "912341", "Moscow").
    people("Petr", "Petrov", "212321", "Moscow").
    people("igor", "Igorov", "212142", "Moscow").
    people("Ivan", "Petrov", "912341", "Moscow").
goal
people("Ivan", Lastname, Phone, Address).

```

[Inactive C:\VIP52\BIN\WIN\32\Obj\goal\$000.exe]

Lastname=Ivanov, Phone=912341, Address=Moscow
Lastname=Petrov, Phone=912341, Address=Moscow
2 Solutions

Рисунок 1 - Результат работы программы 1

```

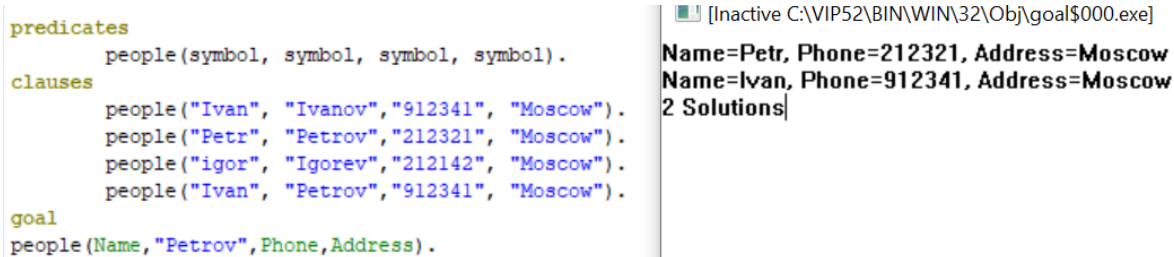
predicates
    people(symbol, symbol, symbol, symbol).
clauses
    people("Ivan", "Ivanov", "912341", "Moscow").
    people("Petr", "Petrov", "212321", "Moscow").
    people("igor", "Igorov", "212142", "Moscow").
    people("Ivan", "Petrov", "912341", "Moscow").
goal
people("Georgiy", Lastname, Phone, Address).

```

[Inactive C:\VIP52\BIN\WIN\32\Obj\g

No Solution

Рисунок 2 - Результат работы программы 2



```

predicates
    people(symbol, symbol, symbol, symbol).
clauses
    people("Ivan", "Ivanov", "912341", "Moscow").
    people("Petr", "Petrov", "212321", "Moscow").
    people("igor", "Igorev", "212142", "Moscow").
    people("Ivan", "Petrov", "912341", "Moscow").
goal
    people(Name, "Petrov", Phone, Address).

```

[Inactive C:\VIP52\BIN\WIN\32\Obj\goal\$000.exe]
Name=Petr, Phone=212321, Address=Moscow
Name=Ivan, Phone=912341, Address=Moscow
2 Solutions

Рисунок 3 - Результат работы программы 3

Из рисунка 1 мы получаем 2 решения, так как в базе знаний два “Ivan”. На рисунке 2 решения не найдены так как нет факта, содержащего “Georgiy”. И наконец на 3 рисунке мы выводим соответствия по фамилии “Petrov”.

Программа на Prolog представляет собой базу знаний и вопрос.

База знаний состоит из предложений - CLAUSES (отдельных знаний или утверждений): фактов и правил. Каждое предложение заканчивается точкой. Предложения бывают двух видов: факты и правила.

Предложение более общего вида – правило. Факт – это частный случай правила. Факт – это предложение, в котором отсутствует тело.

Факт формулируется из составных термов и записывается как $f(t_1, t_2, \dots, t_m)$. Правило называют условной истиной, а факт, не содержащий тела – безусловной истиной.

Правило имеет вид:

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

A называется заголовком правила, а B_1, \dots, B_n – телом правила. Причем, A, B_1, \dots, B_n – это термы; символ “:-” это специальный символ-разделитель.

Ещё одним видом предложений Prolog можно считать вопросы. Вопрос состоит только из тела – составного терма (или нескольких составных термов). Вопросы используются для выяснения выполнимости некоторого отношения между описанными в программе объектами. Система рассматривает вопрос как цель, к которой (к истинности которой) надо стремиться. Ответ на вопрос может оказаться логически положительным или отрицательным, в зависимости от того, может ли быть достигнута соответствующая цель.

Программа на Prolog может содержать вопрос в программе (так называемая внутренняя цель GOAL). Если программа содержит внутреннюю цель, то после

запуска программы на выполнение система проверяет достижимость заданной цели, исходя из базы знаний.

Ответ на поставленный вопрос система дает в логической форме – «Да» или «нет». Цель системы состоит в том, чтобы на поставленный вопрос найти возможность, исходя из базы знаний, ответить «Да». Вариантов ответить «Да» на поставленный вопрос может быть несколько.

В процессе выполнения программы переменные могут связываться с различными объектами – конкретизироваться. Это относится только к именованным переменным. Анонимные переменные не могут быть связаны со значением

Поиск содержательного ответа на поставленный вопрос, с помощью имеющейся базы знаний, фактически заключается в поиске нужного знания, но какое знание понадобится – заранее неизвестно. Этот поиск осуществляется формально с помощью механизма унификации, встроенного в систему и не доступного программисту. Упрощенно, процесс унификации можно представить как формальный процесс сравнения (сопоставления) терма вопроса с очередным термом знания. При этом, знания по умолчанию просматриваются сверху вниз, хотя такой порядок и не очевиден. В процессе сравнения для переменных «подбираются», исходя из базы знаний, значения (для именованных переменных). И эти подобранные для переменных значения возвращаются в качестве побочного эффекта ответа на поставленный вопрос.

Основным элементом языка является терм. Терм – это:

1. Константа:

- Число (целое, вещественное),
- Символьный атом (комбинация символов латинского алфавита, цифр и символа подчеркивания, начинающаяся со строчной буквы: `aA`, `ab_2`), используется для обозначения конкретного объекта предметной области или для обозначения конкретного отношения,
- Строка: последовательность символов, заключенных в кавычки,

2. Переменная:

- Именованная – обозначается комбинацией символов латинского алфавита, цифр и символа подчеркивания, начинающейся с прописной буквы или символа подчеркивания (`X`, `A21`, `_X`),
- Анонимная - обозначается символом подчеркивания (`_`),

3. Составной терм:

• Это средство организации группы отдельных элементов знаний в единый объект, синтаксически представляется: $f(t_1, t_2, \dots, t_m)$, где f - функтор (функциональный символ), t_1, t_2, \dots, t_m – термы, в том числе и составные (их называют аргументами). Аргументом или параметром составного терма может быть константа, переменная или составной объект. Число аргументов предиката называется его арностью или местностью. Составные термы с одинаковыми функторами, но разной арности, обозначают разные отношения.