



**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 № 12.

По курсу «Функциональное и логическое программирование».

Студент: Власова Е. В.

Группа: ИУ7-64Б.

Преподаватели: Толпинская Н. Б.

Строганов Ю. В.

2020 г.

Теоретическая часть.

1. Программа на Prolog, её структура. Как реализуется программа, как формируются её результаты работы.

Программа на Prolog представляет собой базу знаний и вопрос. Основным элементом языка является терм.

Терм – это:

- Константа;
- Переменная;
- Составной терм.

Программа на Prolog не является последовательностью действий, - она представляет собой набор фактов и правил, которые формируют базу знаний о предметной области. Факты представляют собой составные термы, с помощью которых фиксируется наличие истинностных отношений между объектами предметной области — аргументами терма. Правила являются обобщенной формулировкой условия истинности знания — 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> :- <тело правила>.

Заголовок правила — это утверждение базы знаний (предикат), синтаксически это составной терм. Тело правила может представлять собой один терм или быть последовательностью термов (конъюнкцией или дизъюнкцией). В лабораторных работах будем использовать только конъюнкцию (термы в теле разделяются запятыми).

Утверждения программы — это предикаты. Предикаты могут не содержать переменных (основные) или содержать переменные (не основные). В процессе выполнения программы — система пытается

найти, используя базу знаний, такие значения переменных, при которых на поставленный вопрос можно дать ответ «Да».

Программа на Prolog состоит из разделов. Каждый раздел начинается со своего заголовка. Структура программы:

- ^ 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 не обязательно должны быть все разделы.

Практическая часть.

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
    SURNAME = string
    UNIVERSITY = string
predicates
    studying(SURNAME, UNIVERSITY)
clauses
    studying("petrov", "BMSTU").
    studying("petrov", "MAI").
    studying("petrov", "MFTI").
    studying("ivanov", "BMSTU").
    studying("ivanova", UNIVERSITY) :- studying("ivanov",
UNIVERSITY).
    studying("gyreev", "MAI").
```

```
studying("kozlov", UNIVERSITY) :- studying("gyreev",
UNIVERSITY).
studying("evelnov", "MGIMO").
studying("evelnova", UNIVERSITY) :- studying("evelnov",
UNIVERSITY).
goal
SURNAME = "kozlov",
write("SURNAME and UNIVERSITY: "), nl ,
studying(SURNAME, UNIVERSITY).
```


Примеры работы программы:

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. Система должна быть настроена в режим получения всех возможных вариантов ответа «Да» на поставленный вопрос.

1) Студент учится в нескольких университетах:

Система выводит всевозможные ответы «Да» на заданный вопрос.


```
goal
SURNAME = "petrov",
write("SURNAME and UNIVERSITY: "), nl ,
studying(SURNAME, UNIVERSITY).
```

 [Inactive D:\ph\BIN\WIN\32\Obj\goal\$000.exe]

```
SURNAME and UNIVERSITY:
SURNAME=petrov, UNIVERSITY=BMSTU
SURNAME=petrov, UNIVERSITY=MAI
SURNAME=petrov, UNIVERSITY=MFTI
3 Solutions
```

2) Все студенты, обучающиеся в определенном университете:

```
goal
UNIVERSITY = "MAI",
write("SURNAME and UNIVERSITY: "), nl ,
studying(SURNAME, UNIVERSITY).
```


 [Inactive D:\ph\BIN\WIN\32\Obj\goal\$000.exe]

SURNAME and UNIVERSITY:
UNIVERSITY=MAI, SURNAME=petrov
UNIVERSITY=MAI, SURNAME=gyreev
UNIVERSITY=MAI, SURNAME=kozlov
3 Solutions

3) Студента нет в базе:

В данном случае система не найдёт вариант ответить «Да».

```
goal
SURNAME = "petrova",
write("SURNAME and UNIVERSITY: "), nl ,
studying(SURNAME, UNIVERSITY).
```


 [Inactive D:\ph\BIN\WIN\32\Obj\goal\$000.exe]

SURNAME and UNIVERSITY:
No Solution

4) Университета нет в базе:

В данном случае система также не найдёт вариант ответить «Да».

```
goal
UNIVERSITY = "MAIgh",
write("SURNAME and UNIVERSITY: "), nl ,
studying(SURNAME, UNIVERSITY).
```

 [Inactive D:\ph\BIN\WIN\32\Obj\goal\$000.exe]

SURNAME and UNIVERSITY:
No Solution