|  |  |
| --- | --- |
|  | **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**

|  |  |
| --- | --- |
| **Студент:** Керимов А. Ш.  **Группа:** ИУ7-64Б  **Преподаватель:** Толпинская Н. Б. |  |

Москва.

2020 г.

**Цель работы** — познакомиться со структурой, принципами оформления и логикой выполнения программы на Prolog.

**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.

* Исходную базу знаний сформировать с помощью только фактов.
* Исходную базу знаний сформировать, используя правила.
* Разработать свою базу знаний (содержание произвольно).

## Практическая часть

*Листинг 1. База знаний студентов (факты)*

**domains**

surname, name, university, faculty, departament, group = **string**

**predicates**

student(surname, name, university, faculty, departament, group)

**clauses**

student("Kerimov", "Ahmed", "BMSTU", "IU", "IU7", "IU7-64b").

student("Novikov", "Mikael", "BMSTU", "IU", "IU7", "IU7-64b").

student("Yarotskiy", "Danil", "BMSTU", "IU", "IU7", "IU7-61b").

student("Kerimov", "Ahmed", "MSU", "CMC", "SE", "1").

student("Ivanov", "Ivan", "MSU", "CMC", "SE", "1").

student("Ivanov", "Petr", "MSU", "CMC", "SE", "2").

student("Yugay", "Konstantin", "HSE", "CS", "SE", "BPI-175").

student("Volkova", "Anna", "DA RFM", "IR", "IR", "1").

**goal**

student(Surname, Name, "BMSTU", Faculty, Departament, Group).

%student("Kerimov", "Ahmed", University, Faculty, Departament, Group).

%student(Surname, Name, University, Faculty, "SE", Group).

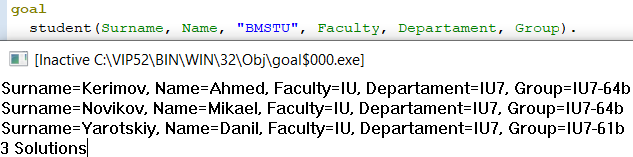
****

Рисунок 1 – Студенты из МГТУ

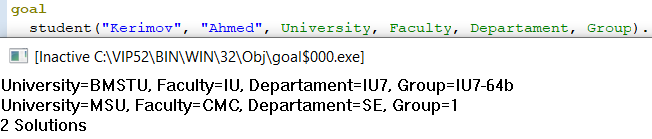
****

Рисунок 2 – Где учится студент



Рисунок 3 – Студенты на «Программной инженерии»

*Листинг 2. База знаний студентов (факты и правила)*

**domains**

surname, name, university, faculty, departament, group = **string**

**predicates**

student(surname, name, university, faculty, departament, group)

in\_one\_group(surname, name, surname, name)

in\_one\_university(surname, name, surname, name)

**clauses**

student("Kerimov", "Ahmed", "BMSTU", "IU", "IU7", "IU7-64b").

student("Novikov", "Mikael", "BMSTU", "IU", "IU7", "IU7-64b").

student("Yarotskiy", "Danil", "BMSTU", "IU", "IU7", "IU7-61b").

student("Kerimov", "Ahmed", "MSU", "CMC", "SE", "1").

student("Ivanov", "Ivan", "MSU", "CMC", "SE", "1").

student("Ivanov", "Petr", "MSU", "CMC", "SE", "2").

student("Yugay", "Konstantin", "HSE", "CS", "SE", "BPI-175").

student("Volkova", "Anna", "DA RFM", "IR", "IR", "1").

in\_one\_group(Surname1, Name1, Surname2, Name2) :-

student(Surname1, Name1, University, Faculty, Departament, Group),

student(Surname2, Name2, University, Faculty, Departament, Group).

in\_one\_university(Surname1, Name1, Surname2, Name2) :-

student(Surname1, Name1, University, \_, \_, \_),

student(Surname2, Name2, University, \_, \_, \_).

**goal**

in\_one\_group("Kerimov", "Ahmed", "Novikov", "Mikael").

%in\_one\_group("Kerimov", "Ahmed", "Yarotskiy", "Danil").

%in\_one\_university("Kerimov", "Ahmed", "Yarotskiy", "Danil").

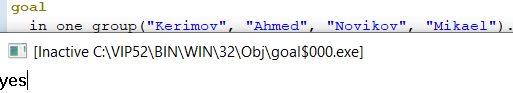


Рисунок 4 - Студенты из одной группы

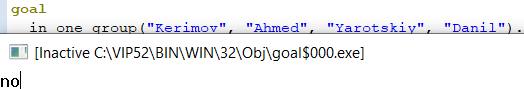


Рисунок 5 - Студенты не из одной группы

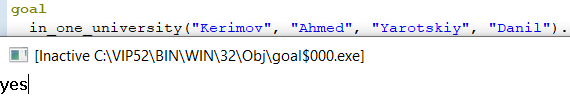


Рисунок 6 - Студенты из одного ВУЗа

*Листинг 3. Генеалогическое дерево*

**domains**

name = **symbol**

**predicates**

parent(name, name)

grandchild(name, name)

has\_child(name)

**clauses**

parent(tom, ann).

parent(tom, mike).

parent(mary, mike).

parent(mike, sue).

parent(mike, john).

parent(mike, nick).

parent(john, jane).

parent(john, jim).

grandchild(X,Z) :- parent(Y,X), parent(Z,Y).

has\_child(X) :- parent(X,Y).

**goal**

grandchild(nick, tom).

%has\_child(john).

%has\_child(jane).

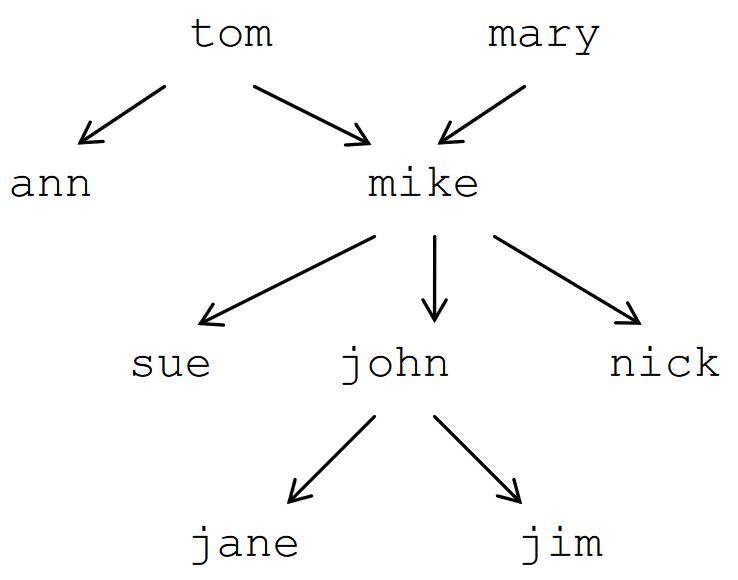


Рисунок 7 - Генеалогическое дерево

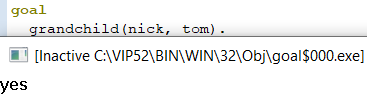


Рисунок 8 – Ник — правнук Тома

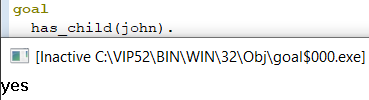


Рисунок 9 - У Джона есть потомство

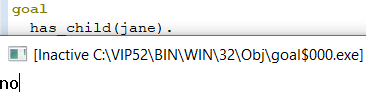


Рисунок 10 - У Джейн нет потомства

## Теоретическая часть

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** — имеет вид:

A :- B1, …, Bn.

**A** называется **заголовком правила**, а **B1, …, Bn**— **телом правила**.

**Факт** — это частный случай правила. Факт — это предложение, в котором отсутствует тело (т. е. тело пустое).

Третьим специфическим 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.

Основным элементом языка является терм. Терм — это:

1. **Константа**:
   * Число (целое, вещественное),
   * Символьный атом (aA, A\_2),
   * 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:**
   * f(t1, t2, ...,tm), где f-функтор (функциональный символ), t1, t2, ..., tm — термы, в том числе и составные.

**Особенность использования переменных**

Именованные переменные уникальны в предикатах одного предложения, анонимные уникальны везде. Анонимные переменные не возвращают значение. Переменной можно обозначить любой объект. При описании переменная может потерять свое значение, но потом его можно вернуть.

**Структура программы**

Программа состоит из разделов

* 1. Domains – описание имен и структур объектов (не обязателен)
  2. Predicates – описание предикатов (названий отношений между объектами)
  3. Clauses – база знаний
  4. Goal – Раздел целевых утверждений

Программа состоит из предложений

* Факт (безусловная истина, формулируется составным термом)- частный случай правил.
* Правила (условная истина, способ порождения новых фактов на основе имеющихся)
  + Пример:

A:- B1, B2,B3. (правило)

A – заголовок, B1, B2,B3.- тело

Вопрос:

* Конъюнктивный (B1, B2, B3)
* Дизъюнктивный (B1; B2; B3).