Министерство науки и высшего образования РФ

Севастопольский государственный университет

Кафедра информатики и управления в технических системах

ОТЧЕТ

по лабораторной работе №1

ВЫПОЛНЕНИЕ И ТРАССИРОВКА ЛОГИЧЕСКИХ ПРОГРАММ В СРЕДЕ PIE

по дисциплине «Логическое программирование»

Выполнил:

Студент группы ИВТ/б 22-о

Черняев Н.Г.

Проверил:

Ткаченко К.С.

г. Севастополь 2019

**Цель работы**

Овладеть навыками трассировки логических программ на примере приложения PIE (Prolog Inference Engine - Машина Вывода Пролога), которое включено в комплект системы программирования Visual Prolog. PIE является "классическим" интерпретатором, используя который можно изучать и экспериментировать с Прологом без описания обьектов программы на уровне обьявления предикатов, типов данных, аргументов и т.д.

**Вариант № 23(2,7,1)**

Задание:

Представить структуры данных и структуры программ в виде деревьев И/ИЛИ и рекурсивных схем. Выполнить программу в режиме трассировки.

Задание к варианту 2

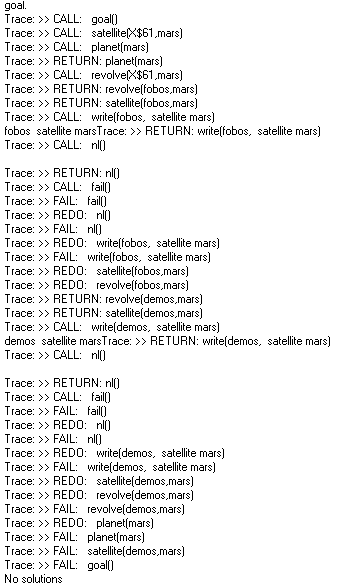
Текст программы 2 планеты.pro:

% domains  
 % name=symbol  
% predicates  
% star(name)  
 % planet(name)  
 % revolve(name,name)  
 % satellite(name,name).  
% clauses  
 star("sun").  
 planet("earth").  
 planet("mars").  
 revolve("earth","sun").  
 revolve("mars","sun").  
 revolve("moon ","earth").  
 revolve("fobos","mars").  
 revolve("demos","mars").  
 satellite(X,Y):-planet(Y),revolve(X,Y).  
goal:-  
 satellite(X,"mars"),  
 write(X,"  satellite mars"),nl,fail.

Результаты работы программы:



Результаты работы программы в режиме трассировки:



Дерево И/ИЛИ:

satellite(X,Y)

planet(Y)

revolve(X,Y)

Рекурсивная схема:

satellite(X,”mars”)

planet(“mars”)

revolve(“fobos”,”mars”)

Задание к варианту 7

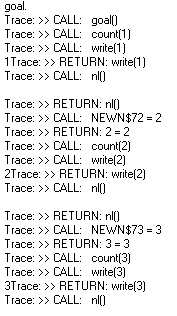
Текст программы 7 count.pro:

% predicates  
  % count(real)  
  
  % clauses  
   count(N) :-   
      write(N), nl,  
      NewN = N+1,  
      count(NewN).  
  
goal:-  
   count(1).

Результаты работы программы:



Результаты работы программы в режиме трассировки:



Дерево И/ИЛИ:

count(N)

write(N)

count(NewN)

NewN = N+1

Рекурсивная схема:

count(1)

write(1)

count(2)

NewN = 1+1

NewN = 2+1

write(2)

count(3)

NewN = 3+1

write(3)

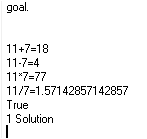
count(4)

Задание к варианту 1

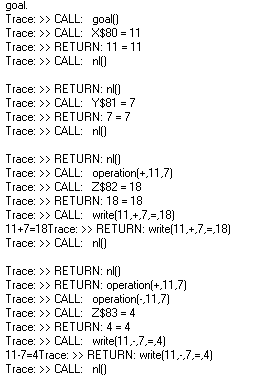
Текст программы 1 четыре арифм опер.pro:

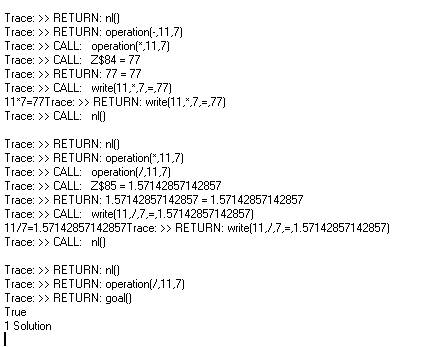
% Реализовать  программу выполнения все четырех арифметических операций.  
 % predicates  
 % operation(symbol, real, real)  
% clauses  
 operation("+",X,Y):-Z=X+Y,  
  write(X,"+",Y,"=",Z),  nl.  
 operation("-",X,Y):-Z=X-Y,  
  write(X,"-",Y,"=",Z),  nl.  
 operation("\*",X,Y):-Z=X\*Y,  
  write(X,"\*",Y,"=",Z),  nl.  
  operation("/",X,Y):-Z=X/Y,  
  write(X,"/",Y,"=",Z),  nl.  
goal:-       X=11, nl,  
             Y=7,  nl,  
             operation("+",X,Y),  
             operation("-",X,Y),  
             operation("\*",X,Y),  
             operation("/",X,Y).

Результаты работы программы:



Результаты работы программы в режиме трассировки:





Дерево И/ИЛИ:

operation("+",X,Y):

Z=X+Y

n1

write(X,"+",Y,"=",Z)

Рекурсивная схема:

operation("+",11,7):

Z=11+7

n1

write(11,"+",7,"=",18)

**Выводы**

В данной лабораторной работе были получены навыки трассировки логических программ на примере приложения PIE (Prolog Inference Engine - Машина Вывода Пролога), которое включено в комплект системы программирования Visual Prolog. PIE является "классическим" интерпретатором, используя который можно изучать и экспериментировать с Прологом без описания обьектов программы на уровне обьявления предикатов, типов данных, аргументов.