



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский технологический университет»
МИРЭА

Подлежит возврату

№ 0010

ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ ПРОЛОГ

ЧАСТЬ 1

Методические указания
по выполнению лабораторных работ
для студентов, обучающихся по направлениям подготовки
09.03.04 «Программная инженерия»
Профиль «Системная и программная инженерия»,
09.03.01 «Информатика и вычислительная техника»
Профиль «Математическое обеспечение вычислительной техники и
автоматизированных систем»

МОСКВА 2015

Составители: В.А. Смольянинова

В методических указаниях излагаются рекомендации и методический материал для выполнения лабораторных работ по дисциплинам «Функциональное и логическое программирование», «Системно-программные основы искусственного интеллекта», «Системы искусственного интеллекта» (в зависимости от учебного плана).

Материал предназначен для студентов дневного, вечернего и заочного отделений и может быть использован для самостоятельной работы.

ЛАБОРАТОРНАЯ РАБОТА 1

ОСНОВНЫЕ ЭЛЕМЕНТЫ ЯЗЫКА ПРОЛОГ

Цель работы: Освоение основных понятий языка Пролог.

Применение фактов и правил для описания отношений на множестве простых объектов.

Подготовка к выполнению работы.

1. Повторите разделы курса:

- основные элементы языка Пролог;
- синтаксис и семантика Пролог- программы.

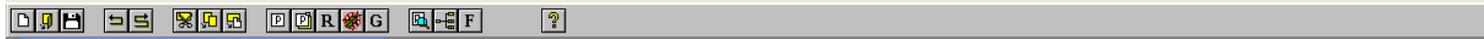
2. Изучите следующие вопросы:

- **структура программы на языке Пролог;**
- **типы доменов;**
- **запись имен объектов и переменных;**
- **запись фактов и правил.**

3. Выполните задания.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

Лабораторные работы выполняются в среде программирования VisualProlog версии 5.2. После запуска системы необходимо открыть учебный проект TestGoal и выполнить его настройки, как показано на рисунках.



C:\Program Files\vip\BIN\WIN...

- Module
- Dialog
- Window
- Menu
- Toolbar
- String
- Icon
- Cursor
- Bitmap
- Help Topics

TestGoal

- TestGoal.pro

New Edit Delete Attribute Code Expert Help

```
TestGoal.pro
1:1 Insert Indent
/*****
Copyright (c) My Company

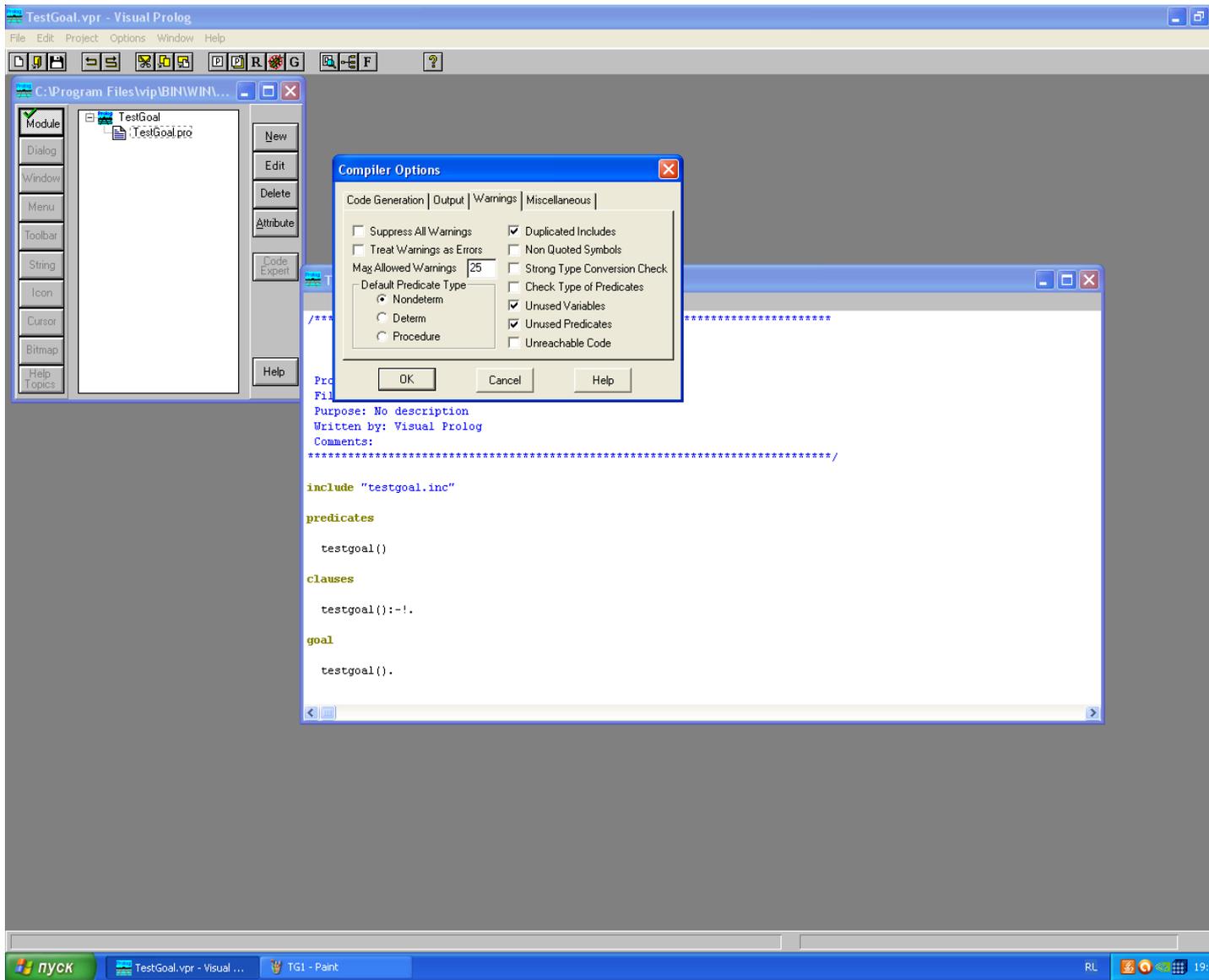
Project: TESTGOAL
FileName: TESTGOAL.PRO
Purpose: No description
Written by: Visual Prolog
Comments:
*****/

include "testgoal.inc"

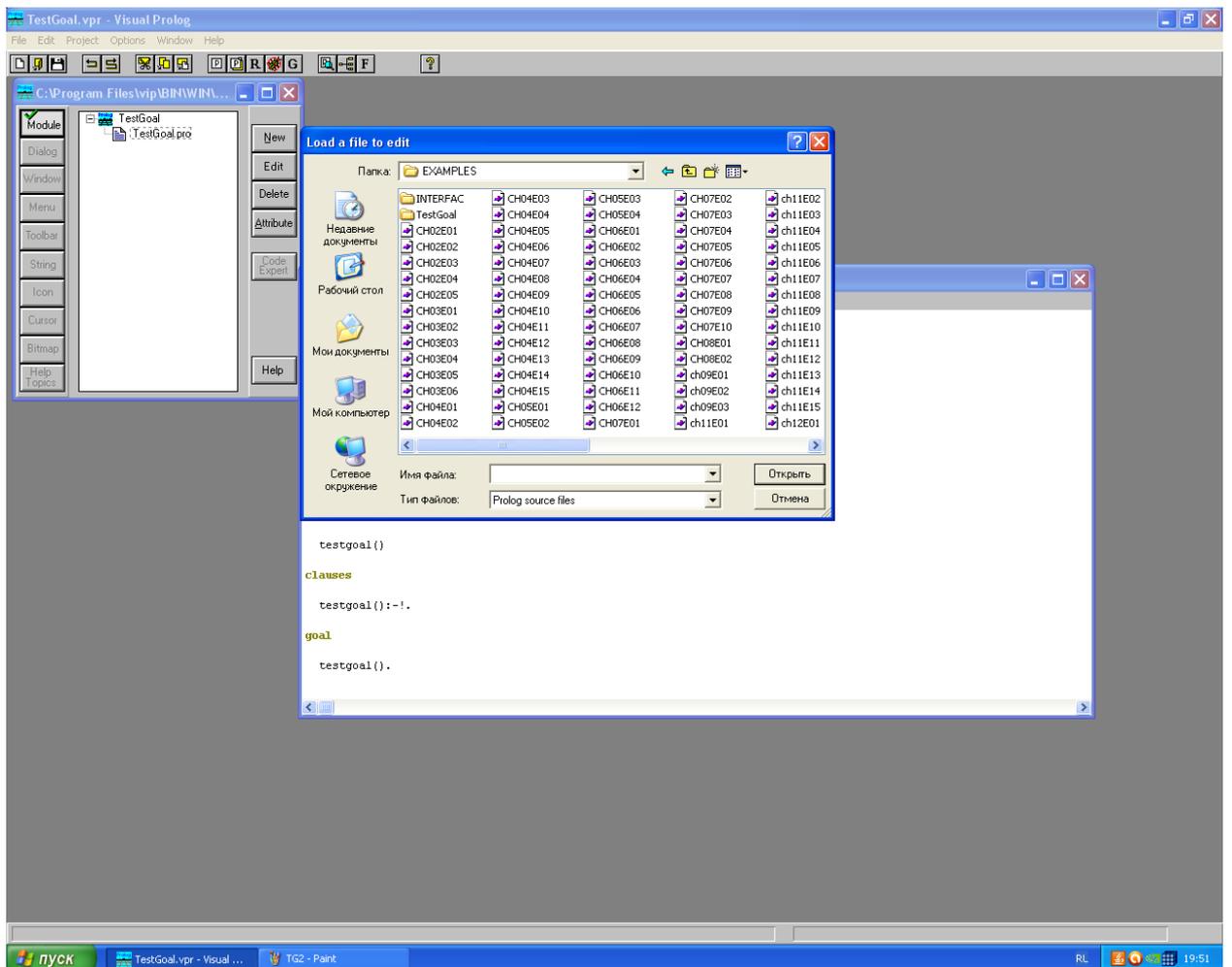
predicates
    testgoal()

clauses
    testgoal():-!.

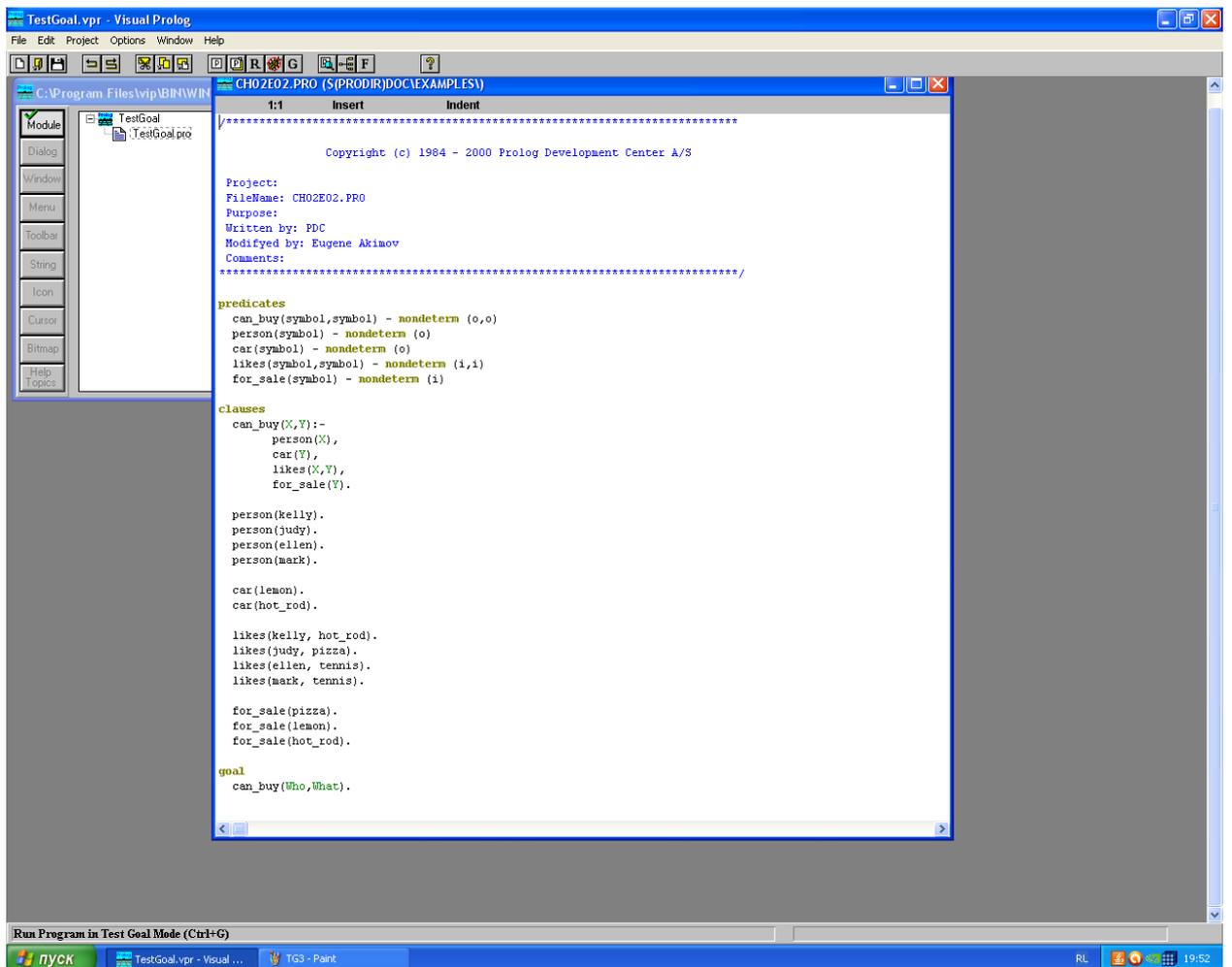
goal
    testgoal().
```



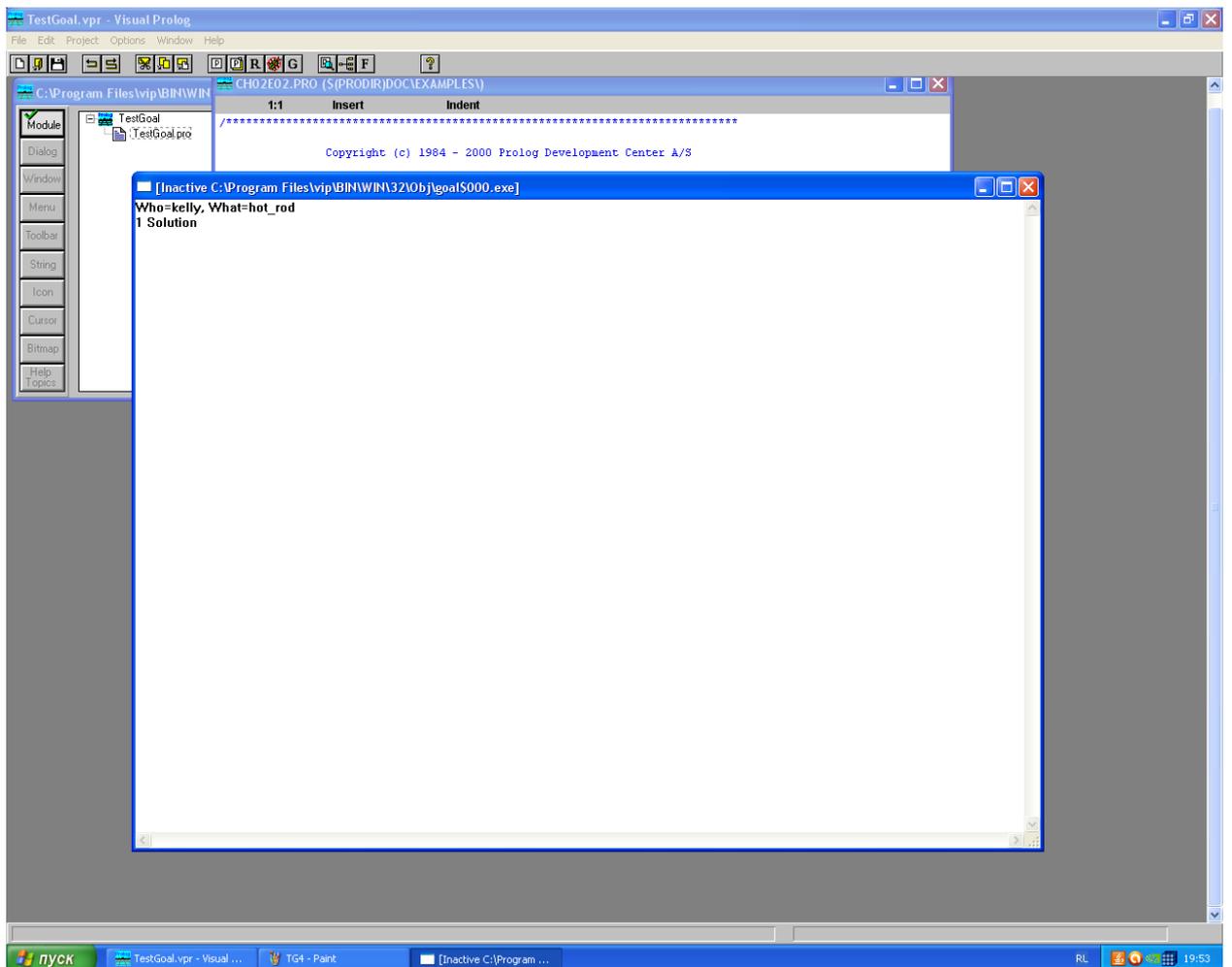
Далее следует открыть примеры для изучения, расположенные в каталоге VIP\DOC\EXAMPLES.



Каждую программу необходимо запустить с разными вариантами целей



и посмотреть, какие ответы выдает система.



Основные понятия языка Пролог

Пролог - это декларативный язык, программы на котором содержат объявления логических взаимосвязей, необходимых для решения задачи. В Прологе рассматриваются отношения между утверждениями и объектами, характерные для логики предикатов.

В программах на Прологе существует три типа предложений (clauses):

факт, правило вывода, цель.

Каждое предложение должно заканчиваться точкой.

Факт - утверждение, истинность которого безусловна. Например,

`likes(mary,apples). /* Мэри любит яблоки */`

или

male(bob) /* Боб - мужчина */

parent(bob,ann). /* Боб - родитель Энн */

Правило - утверждение, зависящее от условий. Например,

child(ann,bob) :-

parent(bob,ann). /* Энн - дитя Боба,

если Боб - родитель Энн */

или

father(X,Y) :-

parent(X,Y),male(X) /* Для всех X и Y

X является отцом Y,если

X является родителем Y и

X - мужчина */

Цель - вопрос пользователя к системе о том, какие утверждения являются истинными.

Для указанных выше примеров на вопрос

child(ann,bob) /* является ли Энн ребенком Боба ?*/

будет выдан ответ

true /* истина */,

а на вопрос

father(X,ann) /* кто является отцом Энн ? */

будет выдан ответ

X = Bob /* отцом Энн является Боб */.

На все поставленные вопросы Пролог пытается ответить с помощью фактов и правил вывода. Он решает задачу, просматривая программу сверху вниз и слева направо. Сначала анализируется цель и ведется поиск такого факта или правила вывода, с помощью которого она может быть достигнута. При нахождении такого

факта после соответствующей подстановки переменных Пролог переходит к анализу следующей цели при условии, что предыдущая достигнута (доказана). Если некоторая цель последняя, доказательство заканчивается. При отсутствии нужных фактов, но наличии правила вывода, которое могло быть применено, цель заменяется условием этого правила с соответствующей подстановкой переменных. Теперь условием выполнения цели становится доказательство условия (правой части) правила вывода. Процесс нахождения соответствия между целью и фактом или правилом называется у н и ф и к а ц и е й. В ходе унификации Пролог ищет все альтернативные решения.

Программа на Прологе включает определенные разделы, не все из которых являются обязательными:

`domains`

`/*(домены) - раздел объявлений*/;`

`database`

`/* описания предикатов динамической базы данных */`

`predicates`

`/* описания предикатов */`

`goal`

`/* целевое утверждение */`

`clauses`

`/* утверждения - факты и правила */`

В программе, по крайней мере, должны быть разделы `predicates` и `clauses`.

Раздел `domains` напоминает объявление данных в традиционных (императивных) языках, например таких, как Паскаль и Си.

Существуют следующие типы доменов:

char (символьный),

integer (целый),

real (вещественный),

string (строковый),

symbol (для цепочки из букв, цифр и символов подчеркивания с первой строчной буквой либо цепочки знаков в двойных кавычках),

file (файловый).

По отношению к именам объектов (идентификаторам) в Прологе используются следующие правила:

1) имя может включать латинские буквы, цифры и символ подчеркивания, причем первым символом не должна быть цифра;

2) имена символьных констант должны начинаться со строчной буквы;

3) в имени можно использовать одновременно и строчные и прописные буквы;

4) имена переменных пишутся с заглавной буквы.

Задание к примерам программ на языке ТурбоПролог.

Наберите программу. Запустите на выполнение. Придумайте и задайте 5-6 вопросов к каждой программе.

Пример 1.Родственные отношения

(Задание: *нарисуйте схему родства*)

domains

s=symbol /* объект s имеет тип symbol */

predicates

parent(s,s)

female(s)

male(s)

mother(s,s)

father(s,s)

ancestor(s,s)

child(s,s)

clauses

parent(pam,bob). /* Пам - родитель Боба */

parent(tom,bob).

parent(tom,liz).

parent(bob,ann).

parent(bob,pat).

parent(pat,jim).

female(pam). /* Пам - женщина */

female(liz).

female(ann).

female(pat).

male(tom). /* Том - мужчина */

male(bob).

male(jim).

child(Y,X):- /* Y - отпрыск X, если X - родитель Y */

parent(X,Y).

mother(X,Y):- /* X - мать Y, если */

parent(X,Y),female(X). /* X - родитель Y и X - женщина */

father(X,Y):- /* X - отец Y, если */

parent(X,Y),male(X). /* X - родитель Y и X - мужчина */

ancestor(X,Z):- /* X - предок Z, если */

parent(X,Z). /* X - родитель Z */

ancestor(X,Z):- /* X - предок Z, если */

parent(X,Y),ancestor(Y,Z). /* X - родитель Y и
Y - предок Z */

Пример 2.Отношение "нравится"

predicates

likes(symbol,symbol)

clauses

likes(ellen, tennis). /* Эллен нравится теннис */

likes(john, football).

likes(tom, baseball).

likes(eric, swimming).

likes(mark, tennis).

Пример 3.Отношение "может купить"

predicates

can_buy(symbol, symbol) /* отношение "может купить" */

person(symbol) /* отношение "субъект" */

car(symbol) /* отношение "марка автомобиля" */

likes(symbol, symbol)

for_sale(symbol) /* отношение "продается" */

clauses

can_buy(X, Y) :-

person(X),

car(Y),

likes(X, Y),

for_sale(Y).

person(kelly).

person(judy).
car(lemon).
car(hot_rod).
likes(kelly, hot_rod).
likes(judy, pizza).
for_sale(pizza).
for_sale(lemon).
for_sale(hot_rod).

Пример 4.

predicates

car(symbol,real,integer,symbol,integer)
truck(symbol,real,integer,symbol,integer)

clauses

car(chrysler,13000,3,red,12000).
car(ford,90000,4,gray,25000).
car(datsun,8000,1,red,30000).
truck(ford,80000,6,blue,8000).
truck(datsun,50000,5,orange,20000).
truck(toyota,25000,5,black,25000).

ЗАДАНИЕ

1. Составьте программу **Родственные отношения**, которая кроме родственных отношений parent (родитель) и ancestor (предок) программа должна содержать одно или несколько из следующих отношений:

brother (брат); sister (сестра); grand-father (дедушка); grand-mother (бабушка); uncle (дядя).

2. Составьте программу, используя отношения likes ("нравится") и can_buy ("может купить").

3. Составьте собственную программу, состоящую из фактов и правил. Проверьте ее работу.

