**Лабораторная работа 02**

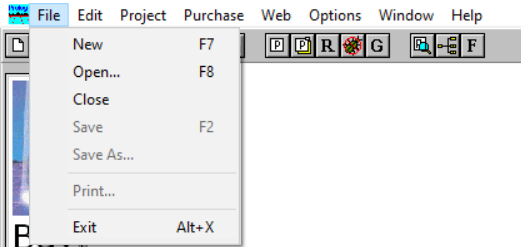
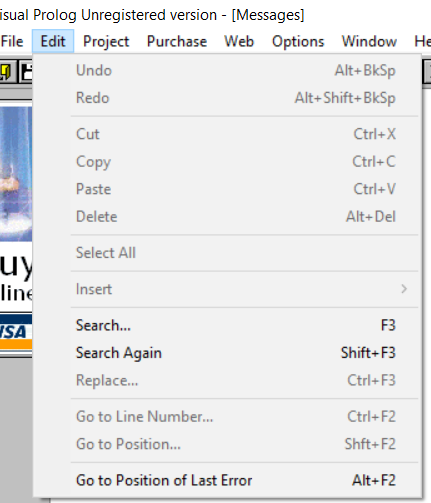
**«Среда Visual Prolog. Простейшая программа.   
Математические операции»**

1. Откройте визуальную среду разработки программ.

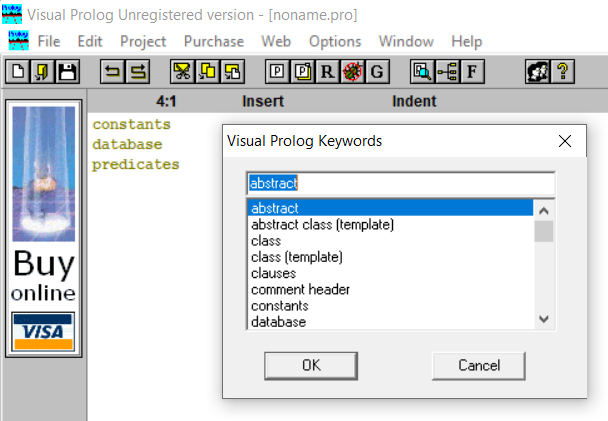
**Пуск** - **Все программы** - **Visual** **Prolog** **5.2** - **Visual** **Prolog** **32**.



1. Ознакомьтесь с интерфейсом программы **Visual** **Prolog**:
2. основными командами пунктов меню;
3. командами сохранения и открытия файла.

1. Используя комбинацию клавиш **Ctrl+Shift+K**, вставьте в окно кода стандартные секции Пролог-программы (доменов, констант, предикатов, фактов и правил, цели).

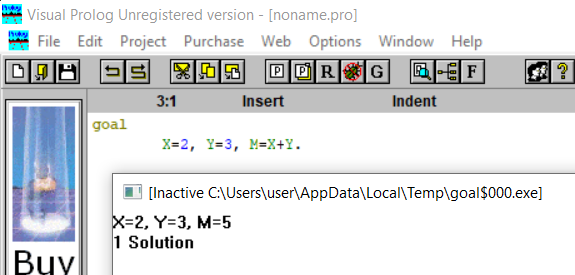


1. Создайте новый файл-модуль и наберите следующий код программы:

goal

X=2, Y=3, M=X+Y.

Осуществите запуск программы на исполнение одним из способов, используя утилиту **Test Goal** (меню **Project** | команда **Test Goal** или комбинация клавиш **CTRL+G** или кнопка  на панели инструментов).

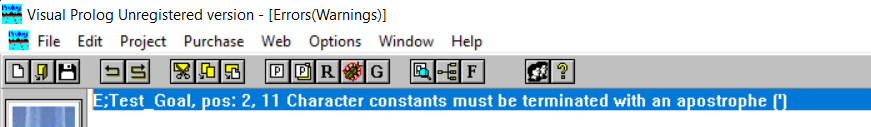


**Обработка ошибок в программе**

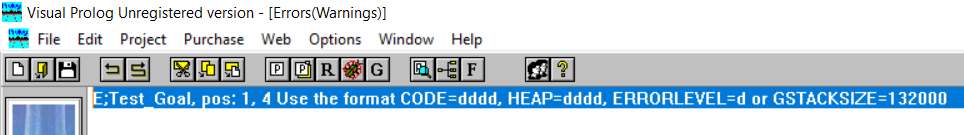
Если в коде программы допущена ошибка, то при запуске программы на исполнение в среде визуальной разработки отобразится окно **Errors** (**Warnings**), которое содержит список обнаруженных ошибок.

Если дважды щелкнуть на строке с указанной ошибкой (окно **Errors (Warnings)**), то курсор переходит в окно кода программы и будет расположен именно на той строке, где допущена ошибка.

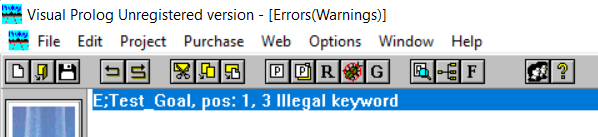
1. Что будет являться операцией сравнения данных
2. ' а' < ' b';



1. "antony" > "antonia";



1. P1 = peter, Р2 = sally, P1>P2?



**Создание проекта программы**

Для создания проекта программы необходимо выполнить следующие действия:

1. создать новый проект (меню **Project** | команда **New Project**);
2. определить имя проекта и установить базовый каталог в окне **Application** **Expert**;
3. определить цель проекта.

На вкладке **Target** окна **Application Expert** установить параметр UI Strategy в значение Textmode.

1. установить требуемые опции компилятора для созданного проекта **Options** | **Project** | **Compiler** **Options** | **Warning:**

* переключатель **Nondeterm** для того, чтобы установить по умолчанию все пользовательские предикаты недерминированными;
* снять флажок **Non** **Quoted** **Symbols**, **Strong** **Type** **Conversion** **Check** и **Check** **Type** **of Predicates** для подавления некоторых возможных предупреждений компилятора;

1. откройте окно редактора кода, запустите и протестируйте программу одним из известных Вам способов;
2. Составьте программу определения четности или нечетности введенного числа с клавиатуры.

% Ввод числа

main :-

write('Insert number: '),

read(Number),

check\_even\_odd(Number).

% Четность числа

check\_even\_odd(Number) :-

(Number mod 2 =:= 0 ->

write(Number), write(' - even number'), nl

;

write(Number), write(' - odd number'), nl

).

1. Измените предыдущую программу так, чтобы исходное число формировалось случайным образом.

% Генерация случайного числа

main :-

random(1, 100, Number),

write('Random number: '), write(Number), nl,

check\_even\_odd(Number).

% Четность числа

check\_even\_odd(Number) :-

(Number mod 2 =:= 0 ->

write(Number), write(' - even number'), nl

;

write(Number), write(' - odd number'), nl

).

**Задания для самостоятельной работы**

Вычислить значения выражений:

Вариант 7.

1. (X+6)2+Y/lg(3.56)

% Ввод чисел

main :-

write('Input X: '),

read(X),

write('Input Y: '),

read(Y),

calculate\_result(X, Y, Result),

write('Result: '), write(Result), nl.

% Значение выражения

calculate\_result(X, Y, Result) :-

Result is ((X + 6) \*\* 2) + Y / log10(3.56).

1. (25+Y+X2)/(6YX\*sin(330o))

% Ввод чисел

main :-

write('Input X: '),

read(X),

write('Input Y: '),

read(Y),

calculate\_result(X, Y, Result),

write('Result: '), write(Result), nl.

% Значение выражения

calculate\_result(X, Y, Result) :-

Result is (25 + Y + (X\*\*2)) / ((6 \* (Y\*\*X)) \* sin(330)).