

Лабораторная работа №3

тема: " Основы языка логического программирования Пролог"

Цели работы: изучить теоретические основы языка логического программирования Пролог и научиться его использовать для решения задач, связанных с множественным выбором и моделированием логических элементов.

Порядок выполнения:

1. Разработать программу на языке Пролог, реализующую расчет значения функции заданной в табл. 1.
2. Построить таблицу истинности логической функции заданной табл. 2 в Excel.
3. Разработать программу на языке Пролог, реализующую моделирование логической функции.
4. Написать отчет, содержащий тексты программ, реализующих индивидуальные задания.

Задания для самостоятельного выполнения

Таблица 1

$$1. F(x) = \begin{cases} x^2 - 3x + 9, & \text{если } x < 3; \\ x^5 + 7x, & \text{если } 3 \leq x < 10; \\ 25, & \text{если } 10 \leq x \leq 35; \\ x^2, & \text{если } x > 35. \end{cases}$$

$$2. F(x) = \begin{cases} x^2 + 3x - 7, & \text{если } x \geq 50; \\ x^3 + 6, & \text{если } 40 \leq x < 50. \\ 7x^2 + 2x - 21, & \text{если } 20 \leq x < 40 \\ 12, & \text{если } x < 20 \end{cases}$$

$$3. F(x) = \begin{cases} \frac{12x^2}{2x + 51} - 10, & \text{если } x \leq 4; \\ x^2 + 2x + 1, & \text{если } 4 < x \leq 30. \\ \sin(x) + \cos(x), & \text{если } 30 < x \leq 90 \\ 27x^2, & \text{если } x > 90 \end{cases}$$

$$4. F(x) = \begin{cases} 0, & \text{если } x \leq 1; \\ x^3 + 13, & \text{если } 1 < x < 10. \\ x^2 + \frac{x^5}{2x^2 + 17x - 25}, & \text{если } 10 \leq x < 30. \\ 15x, & \text{если } x \geq 30. \end{cases}$$

$$5. F(x) = \begin{cases} 3x + 5, & \text{если } x \leq 7; \\ x^2 + 7, & \text{если } 7 < x \leq 27. \\ 2x^3 + \frac{17x^5 - 27}{3x^2 + 14x - 5}, & \text{если } 27 < x \leq 32 \\ 177, & \text{если } x > 32 \end{cases}$$

$$6. F(x) = \begin{cases} 3x + 127, & \text{если } x \leq 8; \\ 3x^2 - 4, & \text{если } 8 < x \leq 17. \\ \frac{2x^3}{x + 17x - 1} + 7x, & \text{если } 17 < x \leq 29 \\ x^2, & \text{если } x > 29 \end{cases}$$

$$7. F(x) = \begin{cases} x^2, \text{ если } 0 \leq x \leq 3; \\ 2x^2 + 7x, \text{ если } 3 < x \leq 13 \\ 4, \text{ если } x > 13 \text{ или } x < 0. \end{cases}$$

$$8. F(x) = \begin{cases} 7x^2 - 4x + 5, \text{ если } x \leq 2; \\ x^4 - 4x^2 + 5x + 1, \text{ если } 2 < x \leq 20. \\ \frac{x^2}{2x+7} + 2x - 1, \text{ если } 20 < x \leq 30. \\ \frac{27}{x^2}, \text{ если } x > 30. \end{cases}$$

$$9. F(x) = \begin{cases} x^2 - x, \text{ если } 0 \leq x \leq 2; \\ \frac{x^2 + 2x - 12}{27x + 5} + 12, \text{ если } 2 < x \leq 20; \\ x^2 + 5x - 1, \text{ если } x > 20 \text{ или } x < 0. \end{cases}$$

$$10. F(x) = \begin{cases} x^2 - x + 9, \text{ если } x \geq 8; \\ x^4 + 6, \text{ если } 6 \leq x < 8; \\ \frac{2}{x^2}, \text{ если } x < 6 \end{cases}$$

$$11. F(x) = \begin{cases} 4x^2 - 2x + 19, \text{ если } x \geq 7; \\ 2x + 17, \text{ если } 3 < x \leq 7; \\ \frac{27x}{x^2 + 1} + 14x - 7, \text{ если } 2 < x \leq 3; \\ -5, \text{ если } x < 2; \end{cases}$$

$$12. F(x) = \begin{cases} x^2 - 3x - 19, \text{ если } x \leq 3; \\ x^2 + 178, \text{ если } 3 < x \leq 20; \\ \frac{27x + 3}{x^2} + 3, \text{ если } 20 < x \leq 21; \\ x^5 - 27x^2, \text{ если } x > 21. \end{cases}$$

$$13. F(x) = \begin{cases} 3x + 9, \text{ если } x \leq 5; \\ x^3 + 8, \text{ если } 5 < x \leq 15; \\ x^2 + 17, \text{ если } 15 < x \leq 23; \\ \frac{27x + 12}{x^2}, \text{ если } 23 < x \leq 35 \\ \frac{3x^3 + 17x - 8}{x^4 + 27}, \text{ если } 35 < x \leq 36 \\ \frac{25 + x}{17x - 199}, x > 36 \end{cases}$$

$$14. F(x) = \begin{cases} 3x^3 + 9, \text{ если } x \leq 13; \\ x^2 + 15, \text{ если } 13 < x \leq 29; \\ 4x^2 + 16x - 67, \text{ если } 29 < x \leq 32; \\ \frac{27x^3 - 27x + 19}{x^2 + 1}, \text{ если } x > 32 \end{cases}$$

$$15. F(x) = \begin{cases} 4x^2 + 5, \text{ если } x > 4; \\ 5x + 4x^2 - 10, \text{ если } 3 < x \leq 4; \\ \frac{17x^4 - 7x^2 + 1}{2x^3 - 10}, \text{ если } 2 < x \leq 3; \\ 16x^6 - 17x^4 + 17, \text{ если } x \leq 2 \end{cases}$$

Таблица 2

1. $Y = (\overline{X1 \vee X2}) \wedge (\overline{X1} \vee \overline{X3})$
2. $Y = \overline{X1 \vee X2} \wedge X1 \vee \overline{X3}$
3. $Y = (\overline{X1} \wedge X2) \vee (X1 \wedge \overline{X3})$
4. $Y = \overline{X1} \wedge X2 \vee \overline{\overline{X1} \wedge X3}$
5. $Y = X1 \vee \overline{X2} \wedge (X1 \vee \overline{X3})$
6. $Y = (X1 \vee X2 \vee X3) \wedge (X1 \vee X2) \wedge (X2 \vee X3)$
7. $Y = (X1 \vee X2 \vee X3) \wedge ((\overline{X2} \wedge \overline{X3}) \vee X1)$

8. $Y = X1 \wedge X2 \vee (X3 \vee X1) \wedge \overline{X2}$
9. $Y = \overline{X1} \vee \overline{X1} \vee X2 \vee (X2 \wedge (\overline{X1} \vee X2))$
10. $Y = \overline{X1} \vee X2 \vee (\overline{X1} \wedge X2) \vee (\overline{X1} \vee X2)$
11. $Y = X1 \wedge X2 \wedge \overline{X3} \vee X1 \wedge \overline{X2} \wedge X3$
12. $Y = (X1 \vee X2) \wedge (\overline{X1} \vee \overline{X2}) \vee X1 \wedge X2$
13. $Y = \overline{X1} \vee X2 \wedge \overline{X3} \vee X1 \wedge (\overline{X2} \vee X3)$
14. $Y = (X1 \wedge \overline{X2}) \vee X3 \vee (\overline{X1} \vee X2) \wedge \overline{X3}$
15. $Y = \overline{X1} \vee X2 \wedge (\overline{X1} \vee X2) \vee X1 \wedge \overline{X3}$

Пример выполнения работы

Задание. Решить систему уравнений:

$$F(x) = \begin{cases} x^2, & \text{если } x \leq 0; \\ x^3, & \text{если } 0 < x. \end{cases}$$

1. Запустим среду Visual Prolog.
2. Создадим новое приложение, выбрав пункт меню Project\New Project. Перед нами откроется диалоговое окно «Application Expert».
3. В поле Project Name укажем наименование проекта «Арифметика» (рис.1).

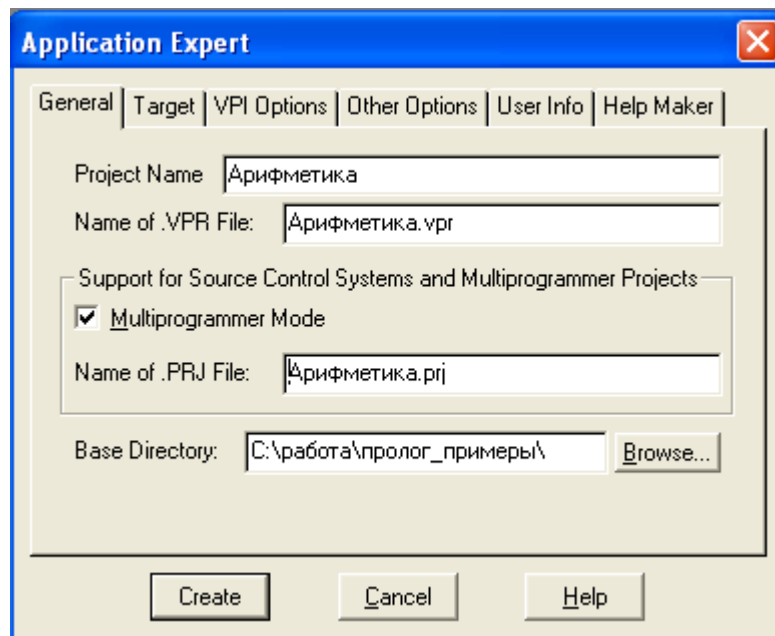


Рис. 1. Окно создания приложения

4. Нажмем кнопку «Create», тем самым откроем окно проекта (рис. 2).

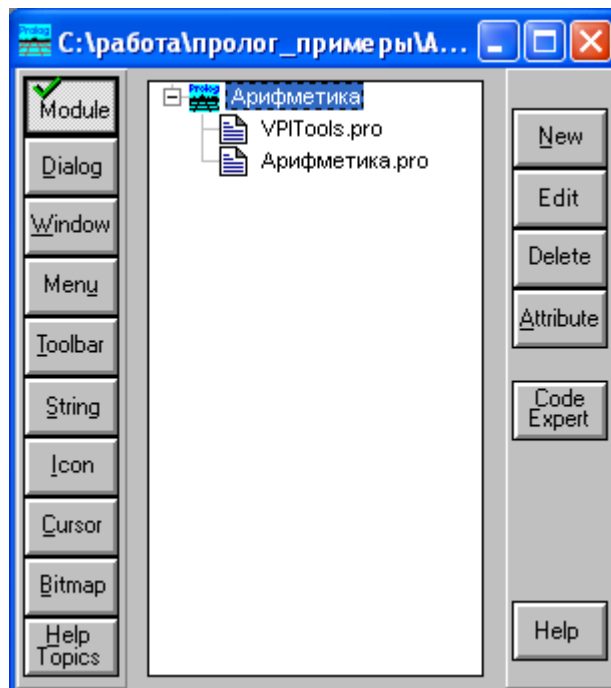


Рис. 2. Окно приложения.

Уже на этом этапе приложение пройдет компиляцию и запуститься.

5. Запустим приложение, нажав либо кнопку F9, либо кнопку на панели инструментов **R**. После чего запустится приложение в виде родительского окна (рис. 3)

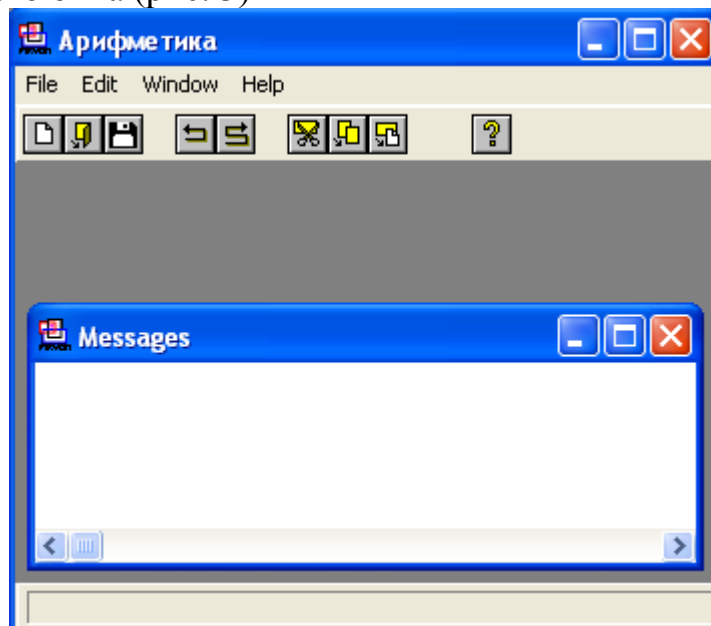


Рис. 3. Приложение «Арифметика»

6. Закроем родительское окно и начнем работу по созданию визуального приложения, решающего поставленную задачу.
7. Обратимся к меню родительского окна Task Menu. Откроем меню на редактирование, нажав на кнопку Edit (рис. 4).

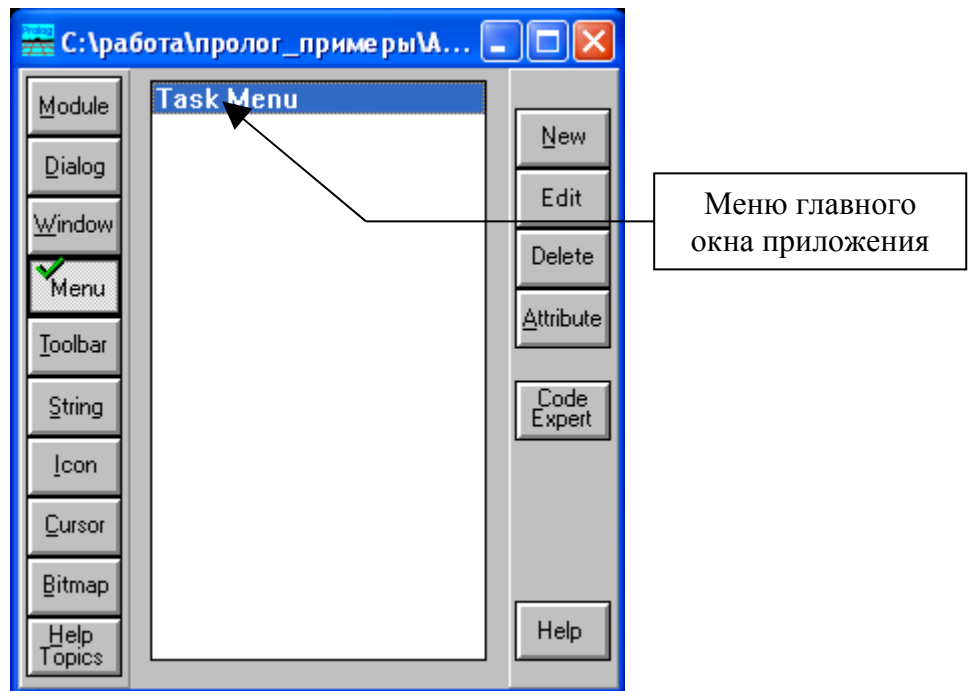


Рис. 4. Меню главного приложения.

8. Добавим новый пункт Тест\Расчет в меню родительского окна. Для этого в редакторе меню нажмем кнопку New и создадим пункт «Тест», затем кликнем по кнопкам Submenu и New, которые позволят нам добавить подпункт «Расчет».
9. Закроем редактор меню, сохранив его.
10. Снова запустим приложение на выполнение и посмотрим результат (рис. 5).

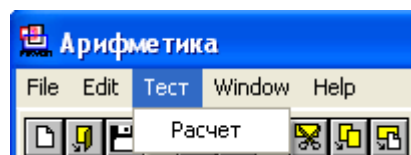


Рис. 5. Новый пункт меню.

11. Остановим приложение.
12. Новый пункт меню будет открывать диалоговое окно, которое можно создать, перейдя на вкладку Window и нажав кнопку New.
13. В открывшемся диалоговом окне «Window Attributes» укажем имя окна «window» и заголовок окна «Моя первая программа», затем нажмем кнопку ОК. На рисунке 6 можно увидеть созданное окно.

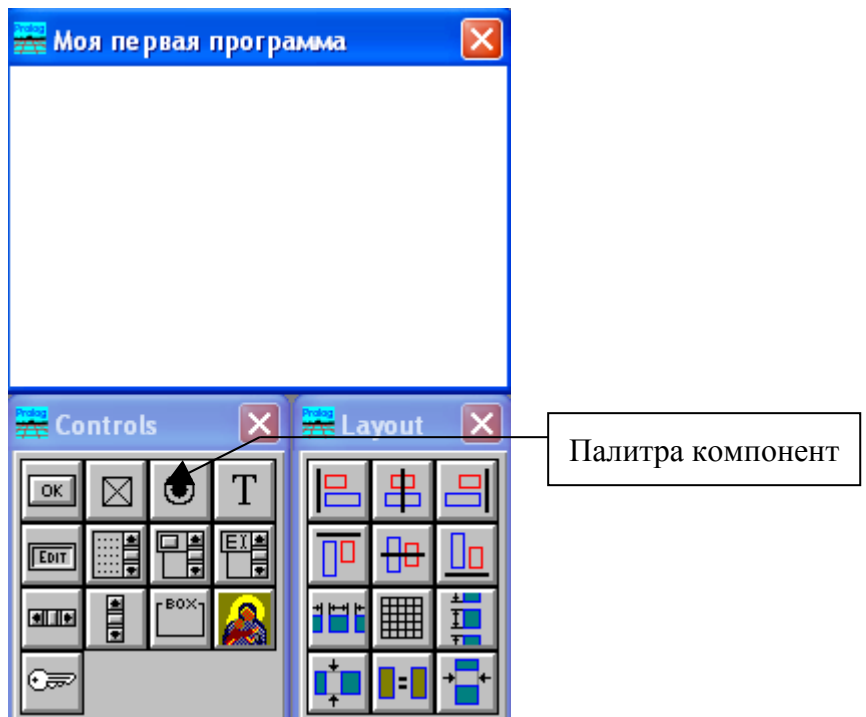


Рис. 6. Редактор окна.

14. Поработаем над меню родительского окна приложения для того, чтобы с помощью пункта меню Тест\Расчет открывалось нами созданное окно window. Для этого нажмем на кнопочку Code Expert.
15. Выберем в качестве редактируемого окна Task Window. Из списка Event Type выберем пункт Menu, а в списке Event or Item выберем строку id_Тест_расчет. Нажмем сначала на кнопку Update Code, затем на Add Clause и Edit Clause. Запишем в правило следующую строку:

$$\text{win_window_Create}(_Win),!$$
16. Закроем редактор кода и перейдем к редактированию диалогового окна «window».
17. Поместим в область окна два объекта типа Static Text, два объекта типа Edit и один объект типа Push Button. Оформим окно в соответствии с рисунком 7.

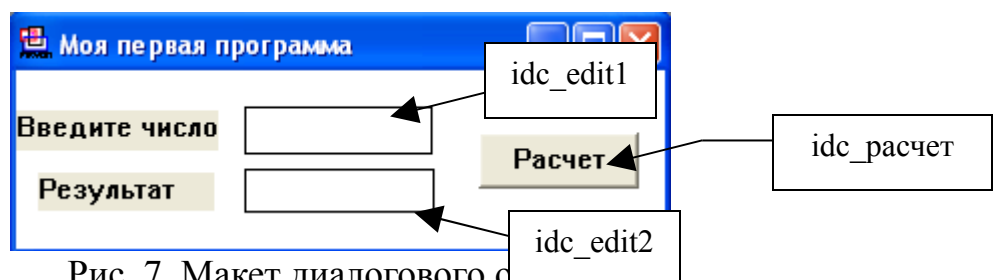


Рис. 7. Макет диалогового окна

18. Обработаем событие нажатия на кнопку «Расчет». Для этого в окне приложения рис. 2 нажмем на кнопку Code Expert.
19. В открывшемся диалоговом окне «Dialog and Window Expert» выберем нами созданное окно «window», установим в строке module значение равное Арифметика.pro. Из списка Event Type выберем пункт Control, а в списке Event or Item выберем строку «rc:idc_расчет». Нажмем сначала на кнопку Update Code, затем на Add Clause и Edit Clause.

20. В открывшемся окне редактирования кода справа от правила `win_window_eh(_Win,e_Control(idc_расчет,_CtrlType,_CtrlWin,_CtlInfo),0):-` запишем :

```
CtrlWindow1=win_GetCtlHandle(_Win, idc_edit1),
CtrlWindow2=win_GetCtlHandle(_Win, idc_edit2),
X=win_gettext(CtrlWindow1),
str_real(X,E),
functions(E,Y),
str_real(S,Y),
win_settext(CtrlWindow2,S),!
```

21. Описание предиката `functions` поместим в раздел `predicates`, который сгенерирован для нашего окна:

```
predicates
...
predicates
win_window_eh : EHANDLER
functions(real,real)
clauses
...

```

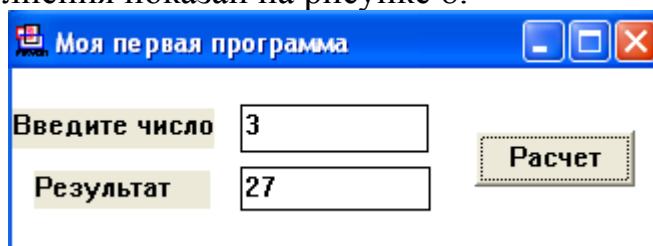
22. В разделе `clauses` запишем следующие правила:

```
...
clauses
functions(X,Y):-Y=X*X,X<=0,!.
functions(X,Y):-Y=X*X*X,X>0,!.

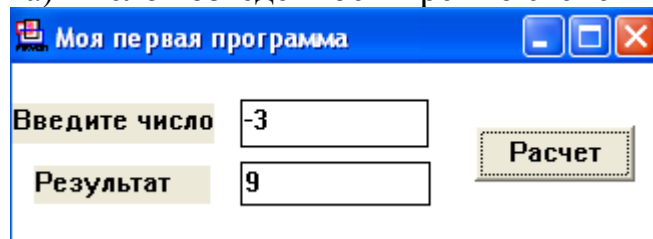
win_window_Create(_Parent):-
...

```

23. Сохраним код. И запустим программу на выполнение. Результат выполнения показан на рисунке 8.



а) Число возведенное в третью степень



б) Число возведенное во вторую степень

Рис. 8. Результат работы программы.