

ЧЕБОКСАРСКИЙ КООПЕРАТИВНЫЙ ИНСТИТУТ
РОССИЙСКОГО УНИВЕРСИТЕТА КООПЕРАЦИИ

Кафедра информационных систем в экономике

Интеллектуальные информационные системы

Методические указания к выполнению лабораторных работ и самостоятельной работы студентов специальности 080801 «Прикладная информатика (в экономике)» очной формы обучения

Чебоксары
2007

Составители: **И.В. Путевская,**
А.В. Речнов.

Рецензенты: зав. кафедрой информатики и информационных технологий Волжского филиала МАДИ (ГТУ), канд. техн. наук **С.В. Ковалев;**
доцент кафедры информатики и математического моделирования ФГОУ ВПО «Чувашский государственный университет им. И.Н. Ульянова», канд. физ.-мат. наук **С.В. Сейфуллина.**

Ответственный редактор: канд. физ.-мат. наук **В.П. Филиппов.**

Утверждено Методическим советом Чебоксарского кооперативного института Российского университета кооперации

Интеллектуальные информационные системы: Методические указания к выполнению лабораторных работ и самостоятельной работы студентов специальности 080801 «Прикладная информатика (в экономике)» очной формы обучения. / Сост. И.В. Путевская, А.В. Речнов. – Чебоксары: ЧКИ РУК, 2007. – с.

В методических указаниях описывается порядок выполнения лабораторных работ и приводятся варианты контрольных заданий по основным разделам курса «Интеллектуальные информационные системы».

Для студентов очной формы обучения специальности 080801 «Прикладная информатика (в экономике)».

Содержание

<u>Введение.....</u>	<u>4</u>
<u>Лабораторная работа №1.</u>	
<u>Основы использования нейронных сетей.....</u>	<u>5</u>
<u>Лабораторная работа №2.</u>	
<u>Классификационный анализ с использованием нейронных сетей.....</u>	<u>8</u>
<u>Лабораторная работа №3.</u>	
<u>Основы языка логического программирования Visual Prolog.....</u>	<u>9</u>
<u>Лабораторная работа №4.</u>	
<u>Использование рекурсивных вычислений в Пролог.....</u>	<u>16</u>
<u>Лабораторная работа №5.</u>	
<u>Обработка строковых данных в Visual Prolog.....</u>	<u>20</u>
<u>Лабораторная работа №6.</u>	
<u>Обработка файлов в Visual Prolog.....</u>	<u>24</u>
<u>Рекомендуемая литература</u>	<u>28</u>

Введение

Основной целью дисциплины «Интеллектуальные информационные системы» является подготовка студентов к созданию и применению интеллектуальных информационных систем для решения экономических задач.

Предметом дисциплины являются методы и средства разработки, классификация, архитектура, использование интеллектуальных систем, применяемых в экономике.

Дисциплина «Интеллектуальные информационные системы» изучается на базе дисциплин «Информатика», «Базы данных», «Разработка и стандартизация программных средств и информационных технологий» во взаимосвязи с экономическими дисциплинами, создающими предметную основу для обучения студентов формализованному описанию задач создания интеллектуальных информационных технологий.

До начала лабораторных занятий студенты получают допуск к работе у ведущего преподавателя. Для этого необходимо подготовить в соответствии с индивидуальным заданием план работы и составить алгоритм решения.

На выполнение каждой лабораторной работы отводится не более трех занятий.

После успешной сдачи очередной работы студент оформляет отчет по следующей схеме:

- 1) титульный лист;
- 2) цель работы;
- 3) постановка задачи;
- 4) текст программы;
- 5) результат.

Защита работы проводится студентом по правильно оформленному отчету.

Студент, выполнивший весь объем лабораторных работ, допускается к сдаче зачета по данному курсу.

Лабораторная работа №1. Основы использования нейронных сетей

Цели работы: изучить теоретические основы нейронных сетей, научиться создавать нейронные сети, реализующие простейшие арифметические операции.

Порядок выполнения:

1. Изучить теоретические основы нейронных сетей.
2. Создать нейронную сеть и обучить ее так, чтобы она выполняла расчеты одновременно по обеим формулам, указанным в таблице 1 (вариант задания соответствует номеру в общем журнале группы).
Например, вариант 1: введя $7+1+4$, получим ответ 12, а введя $7+1-4$ получим ответ 4.
Для создания обучающей выборки использовать интервал значений x, y, z от 0 до 10 включительно.
Погрешность расчетов должна быть не более 1-2%.
3. Продемонстрировать работу преподавателю.
4. Выполнить отчет. В отчете пошагово описать последовательность выполненных операций с обоснованием логики выбора той или иной альтернативы на каждом этапе.

Таблица 1

Задания для самостоятельного выполнения

	а)	б)
1.	$x+y+z$	$x+y-z$
2.	$x+y+z$	$x-y+z$
3.	$x+y+z$	$x*y+z$
4.	$x+y+z$	$x+y*z$
5.	$x+y+z$	$x*y-z$
6.	$x+y+z$	$x-y*z$
7.	$x-y-z$	$x+y-z$
8.	$x-y-z$	$x-y+z$
9.	$x-y-z$	$x*y+z$
10.	$x-y-z$	$x+y*z$
.		
11.	$x-y-z$	$x*y-z$
12.	$x-y-z$	$x-y*z$
.		
13.	$-x+y+z$	$x+y-z$
.		

14	$-x+y+z$	$x-y+z$
15	$-x+y+z$	$x*y+z$
16	$-x+y+z$	$x+y*z$

Продолжение таблицы 1

17	$-x+y+z$	$X*y-z$
18	$-x+y+z$	$x-y*z$
19	$x+y+z$	$-x+y-z$
20	$x+y+z$	$-x-y+z$
21	$x+y+z$	$-x*y+z$
22	$x+y+z$	$-x+y*z$
23	$x+y+z$	$-x*y-z$
24	$x+y+z$	$-x-y*z$
25	$-x+y+z$	$x+y-z$

Пример выполнения работы

Задание. Создать нейронную сеть и обучить ее так, чтобы она выполняла расчет по формуле $x+y$. Для создания обучающей выборки использовать интервал значений x, y от 0 до 2 включительно.

Порядок выполнения работы:

1. Используя MS Excel или Open office org создадим обучающую выборку рис. 1.

0	p	0	0
0	p	0.5	0.5
0	p	1	1
0	p	1.5	1.5
0	p	2	2
0.5	p	0	0.5
0.5	p	0.5	1
0.5	p	1	1.5
0.5	p	1.5	2
0.5	p	2	2.5
1	p	0	1
1	p	0.5	1.5
1	p	1	2
1	p	1.5	2.5
1	p	2	3
1.5	p	0	1.5
1.5	p	0.5	2
1.5	p	1	2.5
1.5	p	1.5	3
1.5	p	2	3.5
2	p	0	2
2	p	0.5	2.5
2	p	1	3
2	p	1.5	3.5
2	p	2	4

Рис. 1. Обучающая выборка

- Создадим набор данных для создания нейронной сети, выбрав пункт меню. File\New\Data set (Количество входных переменных 3, количество выходных переменных 1).
- Определим с помощью пункта контекстного меню «Definition» для второй переменной (VAR2) ее единственное значение «р» (рис. 2)

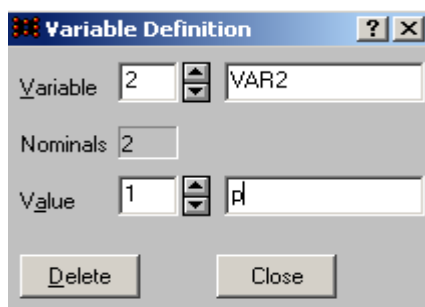


Рис. 2. Определение значения переменной VAR2

- Увеличим количество строк набора данных до 25, выбрав пункт меню Edit\Cases\Add...
- Экспортируем данные, созданные в п.1, используя буфер обмена.
- Создадим оптимальную нейронную сеть, используя пункт меню File\New\Intelligent Problem Solver.

7. Произведем обучение созданной нейронной сети, используя один из пунктов меню Train\Multilayer Perceptrons.
8. Добавим еще одну строку набора, заполним значения переменных VAR1-VAR3 (рис. 3).

	VAR1	VAR2	VAR3	VAR4
01	0.01	p	1.11	?
02	0	p	0	0
03	0	p	0.5	0.5
04	0	p	1	1
05	0	p	1.5	1.5
06	0	p	2	2
07	0.5	p	0	0.5

Рис. 3. Добавление дополнительной строки

9. Используя пункт меню Run\Single Case, проверим работоспособность созданной нейронной сети (рис. 4)

	VAR1	VAR3
Input	0.01	1.11

Outputs Shown: Variables

	VAR4
Output	1.12
Target	?
Error	?

Рис. 4. Проверка на примере работоспособности нейронной сети

Лабораторная работа №2.

Классификационный анализ с использованием нейронных сетей

Цели работы: изучить теоретические основы нейронных сетей, научиться использовать нейронные сети для решения классификационных задач.

Порядок выполнения:

1. Изучить теоретические основы нейронных сетей.
2. Создать нейронную сеть и обучить ее, используя значения экономических показателей прошлого года в качестве обучающей выборки \\student_murk_nw\common\!Задания\ПК4 Интеллектуальные информационные системы\Задания к лабораторной работе2.doc (вариант задания соответствует номеру в общем журнале группы).

3. Выполнить классификацию экономических показателей, определить какие из потребсоюзов окажутся рентабельными, а какие нерентабельными.
4. Продемонстрировать работу преподавателю.
5. Выполнить отчет. В отчете пошагово описать последовательность выполненных операций с обоснованием логики выбора той или иной альтернативы на каждом этапе.

Лабораторная работа №3.

Основы языка логического программирования Visual Prolog

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

Порядок выполнения:

1. Изучить теоретические основы языка логического программирования Visual Prolog.
2. Разработать программу на языке Visual Prolog, реализующую расчет значения функции заданной в табл. 2.
3. Разработать программу на языке Visual Prolog, реализующую моделирование логической функции заданной в табл. 3.
4. Для проверки корректности функционирования разработанной программы, построить в MS Excel таблицу истинности для заданной логической функции.
5. Написать отчет, содержащий тексты программ, реализующих индивидуальные задания.

Таблица 2

Задания для самостоятельного выполнения

$1. F(x) = \begin{cases} x^2 - 3x + 9, \text{ если } x < 3; \\ x^5 + 7x, \text{ если } 3 \leq x < 10; \\ 25, \text{ если } 10 \leq x \leq 35; \\ x^2, \text{ если } x > 35. \end{cases}$	$2. F(x) = \begin{cases} x^2 + 3x - 7, \text{ а́ннè } x \geq 50; \\ x^3 + 6, \text{ а́ннè } 40 \leq x < 50; \\ 7x^2 + 2x - 21, \text{ а́ннè } 20 \leq x < 40; \\ 12, \text{ а́ннè } x < 20. \end{cases}$
$3. F(x) = \begin{cases} \frac{12x^2}{2x+51} - 10, \text{ а́ннè } x \leq 4; \\ x^2 + 2x + 1, \text{ а́ннè } 4 < x \leq 30; \\ \sin(x) + \cos(x), \text{ а́ннè } 30 < x \leq 90; \\ 27x^2, \text{ а́ннè } x > 90. \end{cases}$	$4. F(x) = \begin{cases} x^2, \text{ а́ннè } 0 \leq x \leq 3; \\ 2x^2 + 7x, \text{ а́ннè } 3 < x \leq 13; \\ 4, \text{ а́ннè } x > 13 \text{ èèè } x < 0. \end{cases}$
$5. F(x) = \begin{cases} 3x + 5, \text{ а́ннè } x \leq 7; \\ x^2 + 7, \text{ а́ннè } 7 < x \leq 27; \\ 2x^3 + \frac{17x^5 - 27}{3x^2 + 14x - 5}, \text{ а́ннè } 27 < x \leq 32; \\ 177, \text{ а́ннè } x > 32. \end{cases}$	$6. F(x) = \begin{cases} 3x + 127, \text{ а́ннè } x \leq 8; \\ 3x^2 - 4, \text{ а́ннè } 8 < x \leq 17; \\ \frac{2x^3}{x + 17x - 1} + 7x, \text{ а́ннè } 17 < x \leq 29; \\ x^2, \text{ а́ннè } x > 29. \end{cases}$
$7. F(x) = \begin{cases} x^2 - x, \text{ а́ннè } 0 \leq x \leq 2; \\ \frac{x^2 + 2x - 12}{27x + 5} + 12, \text{ а́ннè } 2 < x \leq 20; \\ x^2 + 5x - 1, \text{ а́ннè } x > 20 \text{ èèè } x < 0. \end{cases}$	$8. F(x) = \begin{cases} x^2 - x + 9, \text{ а́ннè } x \geq 8; \\ x^4 + 6, \text{ а́ннè } 6 \leq x < 8; \\ \frac{2}{x^2}, \text{ а́ннè } x < 6. \end{cases}$
$9. F(x) = \begin{cases} 4x^2 - 2x + 19, \text{ а́ннè } x \geq 7; \\ 2x + 17, \text{ а́ннè } 3 < x \leq 7; \\ \frac{27x}{x^2 + 1} + 14x - 7, \text{ а́ннè } 2 < x \leq 3; \\ -5, \text{ а́ннè } x < 2. \end{cases}$	$10. F(x) = \begin{cases} x^2 - 3x - 19, \text{ если } x \leq 3; \\ x^2 + 178, \text{ если } 3 < x \leq 20; \\ \frac{27x + 3}{x^2} + 3, \text{ если } 20 < x \leq 21; \\ x^5 - 27x^2, \text{ если } x > 21. \end{cases}$

Продолжение таблицы 2

$11. F(x) = \begin{cases} 0, \text{ \u0430\u043d\u0435\u0435 } x \leq 1; \\ x^3 + 13, \text{ \u0430\u043d\u0435\u0435 } 1 < x < 10; \\ x^2 + \frac{x^5}{2x^2 + 17x - 25}, \text{ \u0430\u043d\u0435\u0435 } 10 \leq x < 30; \\ 15x, \text{ \u0430\u043d\u0435\u0435 } x \geq 30. \end{cases}$	$12. F(x) = \begin{cases} 7x^2 - 4x + 5, \text{ \u0430\u043d\u0435\u0435 } x \leq 2; \\ x^4 - 4x^2 + 5x + 1, \text{ \u0430\u043d\u0435\u0435 } 2 < x \leq 20; \\ \frac{x^2}{2x + 7} + 2x - 1, \text{ \u0430\u043d\u0435\u0435 } 20 < x \leq 30; \\ \frac{27}{x^2}, \text{ \u0430\u043d\u0435\u0435 } x > 30. \end{cases}$
$13. F(x) = \begin{cases} 4x^2 + 5, \text{ \u0430\u043d\u0435\u0435 } x > 4; \\ 5x + 4x^2 - 10, \text{ \u0430\u043d\u0435\u0435 } 3 < x \leq 4; \\ \frac{17x^4 - 7x^2 + 1}{2x^3 - 10}, \text{ \u0430\u043d\u0435\u0435 } 2 < x \leq 3; \\ 16x^6 - 17x^4 + 17, \text{ \u0430\u043d\u0435\u0435 } x \leq 2. \end{cases}$	$14. F(x) = \begin{cases} 3x^3 + 9, \text{ \u0430\u043d\u0435\u0435 } x \leq 13; \\ x^2 + 15, \text{ \u0430\u043d\u0435\u0435 } 13 < x \leq 29; \\ 4x^2 + 16x - 67, \text{ \u0430\u043d\u0435\u0435 } 29 < x \leq 32; \\ \frac{27x^3 - 27x + 19}{x^2 + 1}, \text{ \u0430\u043d\u0435\u0435 } x > 32. \end{cases}$
$15. F(x) = \begin{cases} 3x + 9, \text{ \u0430\u043d\u0435\u0435 } x \leq 5; \\ x^3 + 8, \text{ \u0430\u043d\u0435\u0435 } 5 < x \leq 15; \\ x^2 + 17, \text{ \u0430\u043d\u0435\u0435 } 15 < x \leq 23; \\ \frac{27x + 12}{x^2}, \text{ \u0430\u043d\u0435\u0435 } 23 < x \leq 35; \\ \frac{3x^3 + 17x - 8}{x^4 + 27}, \text{ \u0430\u043d\u0435\u0435 } 35 < x \leq 36; \\ \frac{25}{x} \\ 17x - 199, x > 36. \end{cases}$	

Таблица 3

1.	$Y = (\overline{X1} \vee \overline{X2}) \wedge (\overline{X1} \vee \overline{X3})$
2.	$Y = \overline{X1} \vee \overline{X2} \wedge X1 \vee \overline{X3}$
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.	$Y = \overline{X1} \vee \overline{\overline{X1} \vee X2} \vee (X2 \wedge (\overline{X1} \vee \overline{X2}))$
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:

$$F(x): \begin{cases} x^2, \text{ а́ннè } x < 0; \\ x^3, \text{ а́ннè } 0 < x. \end{cases}$$

Порядок выполнения работы:

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. 5).

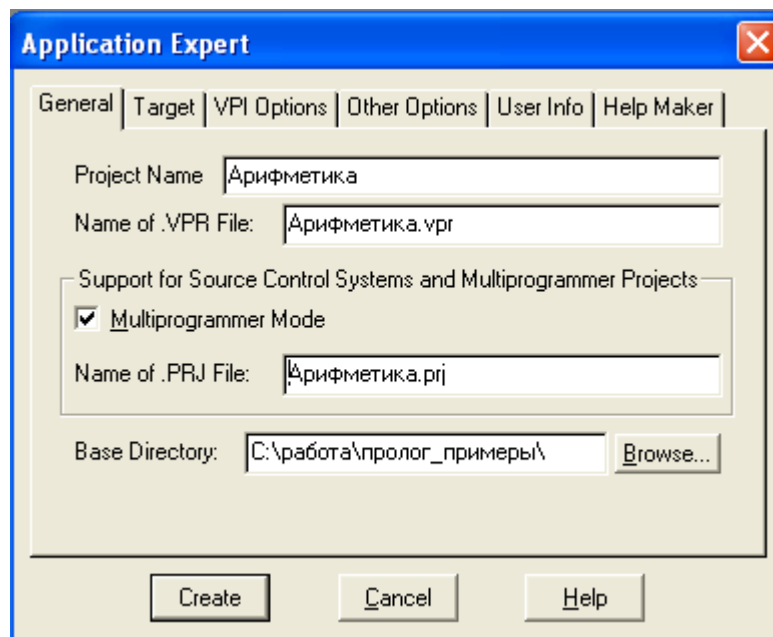


Рис. 5. Окно создания приложения

4. Создадим новый проект, нажав кнопку «Create».
5. Запустим приложение, нажав либо кнопку «F9», либо кнопку на панели инструментов **R**. После чего запустится приложение в виде «родительского» окна (рис. 6)

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, используя кнопку «Edit» (рис. 7).

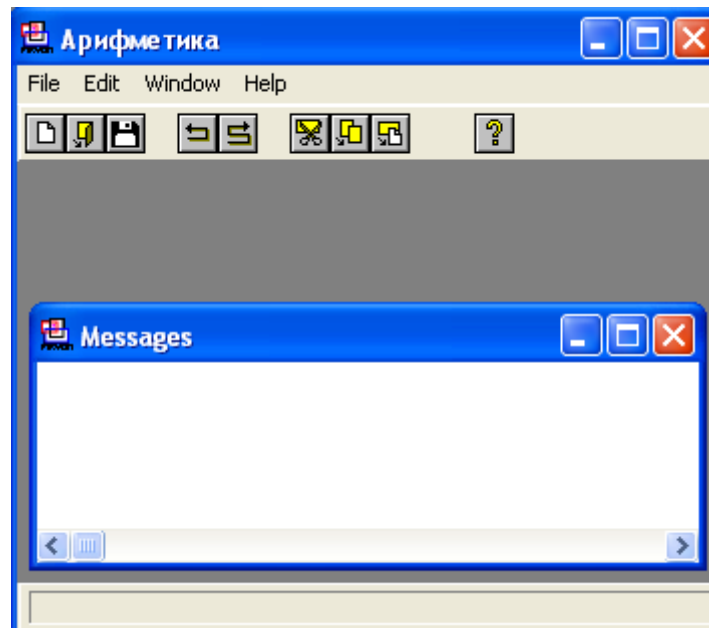


Рис. 6. Приложение «Арифметика»

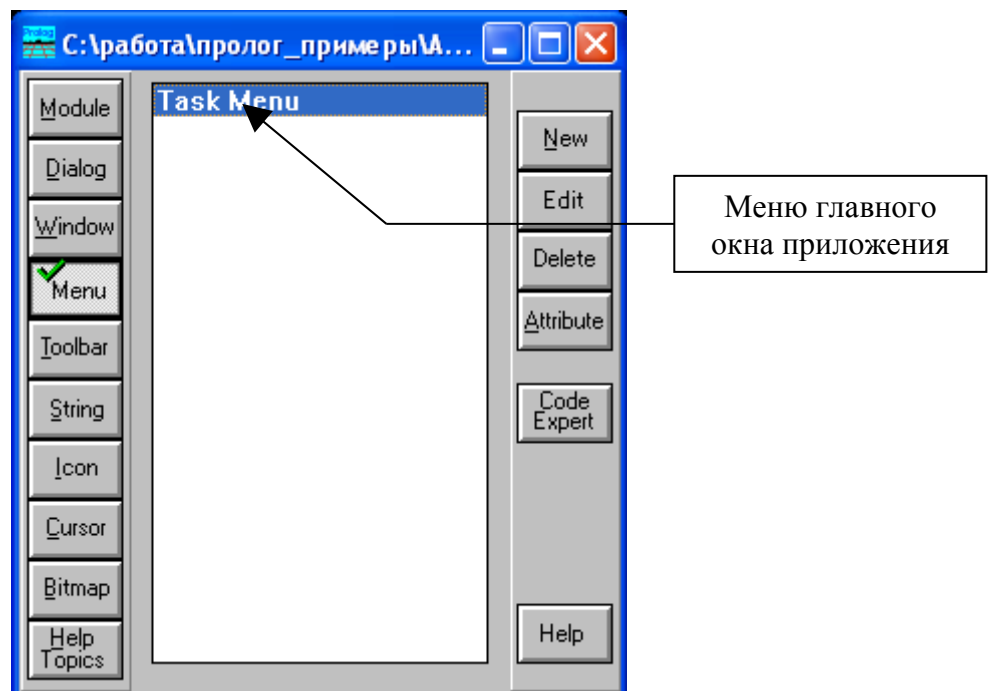


Рис. 7. Меню главного приложения.

8. Добавим новый пункт Тест\Расчет в меню «родительского» окна. Для этого в редакторе меню нажмем кнопку «New» и создадим пункт

«Тест», затем используя кнопки «Submenu» и «New» для добавления подпункта «Расчет».

9. Закроем редактор меню, сохранив его.
10. Снова запустим приложение на выполнение. Результат работы приложения изображен на рис. 8.

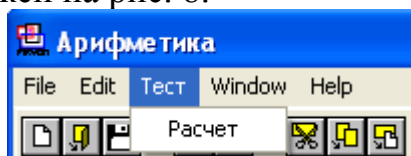


Рис. 8. Новый пункт меню.

11. Завершим работу приложения.
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» рис. 9.

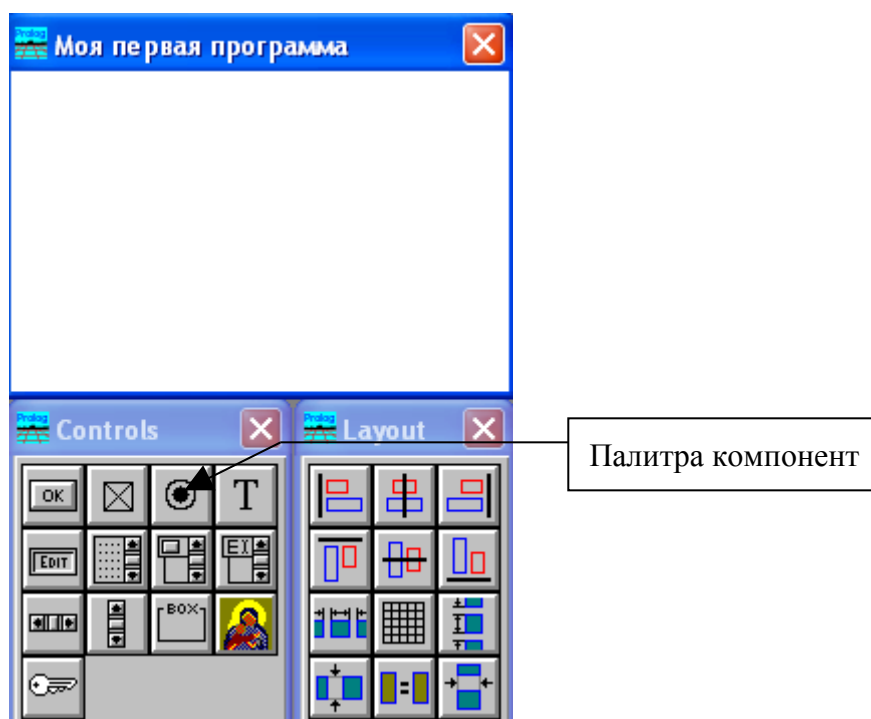


Рис. 9. Редактор окна.

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 кнопку «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:

```
task_win_eh(_Win,e_Menu(id_Тест_Расчет,_ShiftCtlAlt),0):-!,
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 рис. 10.

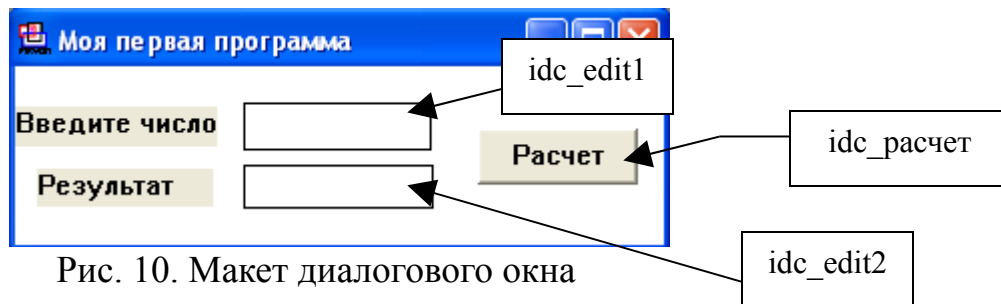


Рис. 10. Макет диалогового окна

18. Обрабатываем событие нажатия на кнопку «Расчет». Для этого в окне приложения рис. 5 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 «Default Code», а затем на кнопки «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):-**

**functions(X,Y):-Y=X*X*X,X>0,!.
win_window_Create(_Parent):-**

win_window_Create(_Parent):-

...

23. Сохраним код. И запустим программу на выполнение. Результат выполнения показан на рис. 11.

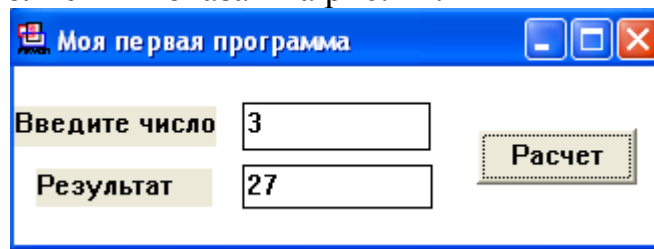


Рис. 11. Результат работы программы

Лабораторная работа №4.

Использование рекурсивных вычислений в Пролог

Цели работы: изучить теоретические основы использования рекурсивных вычислений в языке логического программирования Visual Prolog и научиться их использовать для решения задач, связанных с циклическими алгоритмами.

Порядок выполнения:

1. Изучить теоретические основы использования рекурсивных вычислений в языке логического программирования Visual Prolog.
2. Разработать программу на языке логического программирования Visual Prolog, реализующую расчет значения рядов, заданных в табл. 4.
3. Написать отчет, содержащий текст программы, реализующей индивидуальное задание.

Задания для самостоятельного выполнения

№	а	б
1	$\sum_{i=1}^n \frac{1}{(i+1)^3}$	$\prod_{n=2}^{32} \left(1 - \frac{1}{n^2}\right)$
2	$\sum_{n=1}^{12} \frac{1}{\sqrt{n(n+1)}}$	$\prod_{n=2}^{33} \frac{n^3 - 1}{n^3 + 1}$
3	$\sum_{n=1}^{11} n^2 e^{-\sqrt{n}}$	$\prod_{n=1}^{34} \operatorname{ch} \frac{1}{2^n}$
4	$\sum_{n=1}^{11} \frac{1}{\ln^2 \sin \frac{1}{n}}$	$\prod_{n=1}^{36} \cos \frac{1+n}{3^{n-1}}$
5	$\sum_{n=1}^{11} (n^{n^2-1} - 1)$	$\prod_{n=1}^{37} \cos \frac{n^3 - 1}{n^3 + 1}$
6	$\sum_{n=1}^{13} e^{\sqrt[3]{n}}$	$\prod_{n=1}^{38} \left(1 + \frac{1}{4^n}\right)$
7	$\sum_{n=1}^{11} \sin^3 \frac{1}{n}$	$\prod_{n=1}^{39} \left(1 + \frac{1}{n(n+2)}\right)$
8	$\sum_{n=1}^{11} \frac{n^{\ln n}}{(\ln n)^n}$	$\prod_{n=1}^{40} \left[1 + \left(\frac{1}{2}\right)^{2n}\right]$
9	$\sum_{n=1}^{11} \left[\frac{1}{2n} + \frac{(-1)^n}{3^n} \right]$	$\prod_{n=1}^{41} \sqrt{\frac{n+1}{n+2}}$
10	$\sum_{n=1}^{20} \frac{\cos \frac{2\pi n}{3}}{2^n}$	$\prod_{n=1}^{42} \sqrt[n]{1 + \frac{1}{n}}$
11	$\sum_{n=1}^{21} \frac{1}{2^{n/2+1}}$	$\prod_{n=1}^{43} \frac{n}{\sqrt[3]{n^3 + 2}}$
12	$\sum_{n=1}^{11} \frac{(-1)^n}{\sqrt{n}}$	$\prod_{n=1}^{50} \left(1 + \frac{1}{3^{n+1}}\right)$
13	$\sum_{n=1}^{23} 2^n \sin \frac{1}{3^n}$	$\prod_{n=1}^{51} \left(1 + \frac{n+1}{n(n+2)}\right)$

14	$\sum_{n=1}^{25} \frac{\cos n}{n}$	$\prod_{n=1}^{53} \left(1 + \frac{n+1}{n^2+1}\right)$
15	$\sum_{n=1}^{28} \frac{1}{(n+1)^2-1}$	$\prod_{n=1}^{54} \frac{n^3+1}{n^3}$

Пример выполнения работы

Задание. Вычислить сумму ряда $\frac{1}{n^2}$.

Порядок выполнения работы:

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 «Рекурсия».
4. Нажмем кнопку «Create», тем самым откроем окно проекта.
5. Добавим новый пункт «Тест\Рекурсия» в меню «родительского» окна.
6. Новый пункт меню будет открывать диалоговое окно, которое создадим, перейдя на вкладку «Window» и нажав кнопку «New».
7. Назовем окно «window».
8. Сформируем следующую структуру окна, поместив новый для нас компонент «ListBox» (рис. 12).



Рис. 12. Макет диалогового окна.

9. Добавим в обработчик пункта меню «Тест\Рекурсия» следующую строку:

```

task_win_eh(_Win,e_Menu(id_Тест_Рекурсия,_ShiftCtlAlt),0):-!,
win_window_Create(_Win),!.

```

10. Обработчик события кнопки «Расчет» выглядит следующим образом:

```

win_window_eh(_Win,e_Control(idc_расчет,_CtrlType,_CtrlWin,_CtlInfo),0):-!,

```

```

CtrlWindow1=win_GetCtlHandle(_Win, IDC_EDIT1),
CtrlWindow2=win_GetCtlHandle(_Win, IDC_EDIT2),
CtrlWindow3=win_GetCtlHandle(_Win, IDC_LIST),
lbox_clear(CtrlWindow3),
S=win_gettext(CtrlWindow1),
str_real(S,X),
ряд(X,Y, CtrlWindow3),
str_real(F,Y),
win_settext(CtrlWindow2,F),
!.

```

11. Описание предиката, предназначенного для вычисления суммы ряд поместим в раздел «predicates», который сгенерирован для нашего окна:

ряд(real,real,WINDOW)

12. В разделе «clauses» запишем следующие правила:

```
ряд(X,Y,C):-X=1,Y=X,str_real(S,Y),lbox_Add(C,S),!
```

```
ряд(X,Y,C):-T=X-1,
```

```
ряд(T,Z,C),W=1/(X*X),Y=Z+W,str_real(S,W),lbox_Add(C,S).
```

13. Сохраним код. И запустим программу на выполнение.

Результат выполнения показан на рис. 13.

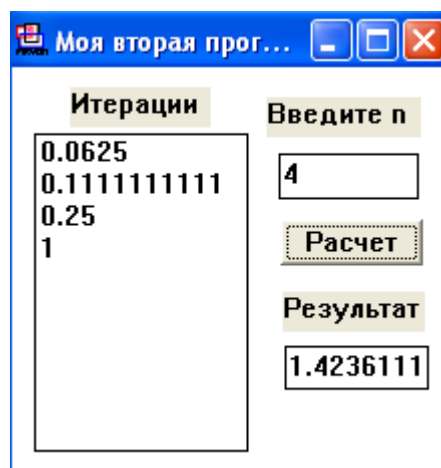


Рис. 13. Результат работы программы.

Лабораторная работа №5. Обработка строковых данных в Visual Prolog

Цели работы: изучить теоретические основы использования рекурсивных вычислений в языке логического программирования Visual Prolog и научиться их использовать для решения задач, связанных с обработкой строковых данных.

Порядок выполнения:

1. Изучить теоретические основы использования языка логического программирования Visual Prolog для обработки строковых данных.
2. Разработать программу на языке логического программирования Пролог, реализующую индивидуальное задание.
3. Написать отчет, содержащий текст программы, реализующей индивидуальное задание.

Задания для самостоятельного выполнения

1. Подсчитать количество слов в заданной строке.
2. Подсчитать количество букв "u" в последнем слове заданной строки.
3. Найти количество слов в заданной строке, начинающихся с буквы "b".
4. Найти количество слов в заданной строке, у которых первый и последний символы совпадают между собой.
5. Найти длину самого короткого слова в заданной строке.
6. В заданной строке замените "a" на букву "e", если "a" стоит на четной позиции в слове, и заменить букву "b" на сочетание "ak", если буква "b" стоит на нечетной позиции в слове.
7. В заданной строке найти и вывести все слова, отличные от последнего слова.
8. Отредактировать заданное предложение текста, удаляя из него все слова с нечетными номерами.
9. Написать программу для подсчета суммы мест, на которых в заданной строке текста стоит заданная буква.
10. Составить список слов заданного текста, начинающихся с буквы "A", с указанием числа повторений каждого слова в тексте.
11. Составить программу для удаления из слов в заданном тексте всех букв, стоящих на нечетных позициях после последней буквы "a" в слове.
12. Составить программу для перевода заданных арабских чисел в диапазоне от 1 до 1000 в римские.

13. Автоморфными называются числа, которые содержатся в последних разрядах их квадрата. Например: $52=25$, $252=625$. Составить программу для проверки, имеются ли в заданной строке автоморфные числа.
14. Подсчитать, сколько букв надо исправить в слове X, чтобы получилось слово Y (X, Y - слова одинаковой длины).
15. Дана фраза. Определить, имеются ли в ней слова – палиндромы (одинаково читаются слева направо и справа – налево).

Пример выполнения работы

Задание. Создать программу, которая по заданной строке и символу подсчитывает количество вхождений этого символа в данную строку.

Порядок выполнения работы:

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 «Строки».
4. Нажмем кнопку «Create», тем самым откроем окно проекта.
5. Добавим новый пункт «Тест\Строки» в меню «родительского» окна.
6. Новый пункт меню будет открывать диалоговое окно, которое создадим, перейдя на вкладку «Window» и нажав кнопку «New».
7. Назовем окно «window».
8. Сформируем следующую структуру окна, поместив новый компонент «List Edit» (рис. 14).

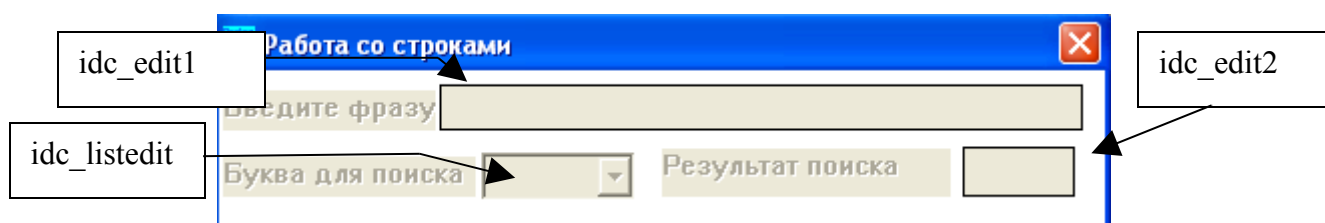


Рис. 14. Макет диалогового окна

9. После того как компонента «List Edit» размещена на окне, ее необходимо «растянуть» так, как показано на рис. 15.

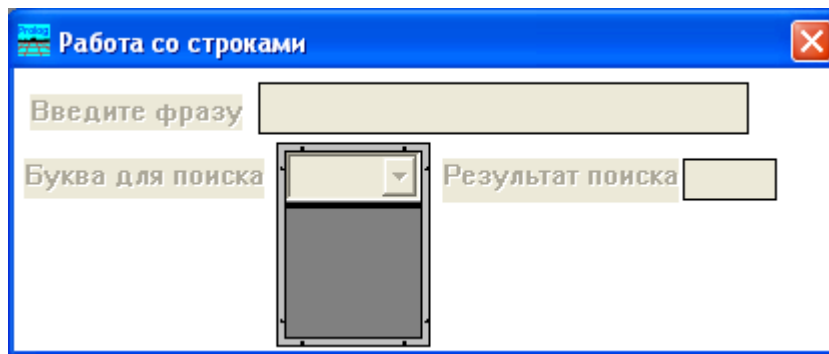


Рис. 15. Формирование выпадающего списка

10. Выберем пункт «Control Attributes», используя контекстное меню объекта «`idc_edit2`».
11. В открывшемся диалоговом окне «EditText Attributes» нажмем кнопку «Control Flags» и установим свойство «Disabled», тем самым запретим редактирование в данном текстовом окне.
12. Закроем диалоговые окна свойств, нажав кнопку «Ok» два раза.
13. Обработаем событие открытия окна.
14. В разделе описания предикатов для нашего окна «window» опишем новый предикат «`char_count`». Для этого в окне приложения рис. 5 нажмем на кнопку «Code Expert». В открывшемся диалоговом окне «Dialog and Window Expert» выберем нами созданное окно «window», установим в строке «module» значение равное «Строка.pro» и нажмем кнопку «Edit Code».
15. В открывшемся редакторе кода перейдем к описанию предикатов окна «window» и внесем туда изменения :

```

predicates
  win_window_eh : EHANDLER
  char_count(string,char,integer)

```

16. Ниже опишем правила в разделе «clauses»:


```

char_count("",_,0):-!. /* Любой символ не встречается
      в пустой строке ни разу*/
char_count(S,C,N):-
  frontchar(S,C,S1),!,
  /* символ C оказался первым символом
  строки S, в S1 - оставшиеся
  символы строки S */
  char_count(S1,C,N1),
  /* N1 - количество вхождений
  символа C в строку S1 */
  N=N1+1.
  /* N - количество вхождений
  символа C в строку S получается

```

```

из количества вхождений символа C
в строку S1 добавлением единицы */
char_count(S,C,N):-
    frontchar(S,_,S1),
    /* первым символом строки S
    оказался символ, отличный
    от исходного символа C, в S1 -
    оставшиеся символы строки S */
    char_count(S1,C,N).
/* в этом случае количество
вхождений символа C в строку S
совпадает с количеством
вхождений символа C
в строку S1 */

```

24. Обработаем событие изменения компоненты «`idc_listedit`». Для этого в окне приложения рис. 5 нажмем на кнопку «Code Expert».
25. Из списка «Event Type» выберем пункт «Control», а в списке «Event or Item» выберем строку «selchange». Нажмем сначала на кнопку «Update Code», затем на «Add Clause» и «Edit Clause».
26. В открывшемся окне редактирования кода справа от правила `win_window_eh(_Win,e_Control(idc_listedit,_CtrlType,_CtrlWin,selchange),0):-!`, запишем:

```

C_edit1=win_getCtlHandle(_Win,idc_edit1),
T=win_gettext(C_edit1),
C_list=win_getCtlHandle(_Win,idc_listedit),
I=lbox_GetSelIndex(C_list),
S=lbox_GetItem(C_list,I),
str_char(S,R),
char_count(T,R,Y),
C_edit2=win_getCtlHandle(_Win,idc_edit2),
str_int(K,Y),
win_settext(C_edit2,K),!.

```

27. Добавим в правило, которое выполняет открытие нашего окна `window`, следующую инициализацию компонента «`idc_listedit`»:

```

win_window_eh(_Win,e_Create(_,0):-!,
...
C1=win_getCtlHandle(_Win,idc_listedit),
LIST=[а,б,в,г,д,е,ж,з,и,й,к,л,м,н,о,п,р,с,т,ф,у,х,ш,щ,э],
lbox_add(C1,LIST),!.

```

28. Пример выполнения программы изображен на рис. 16.

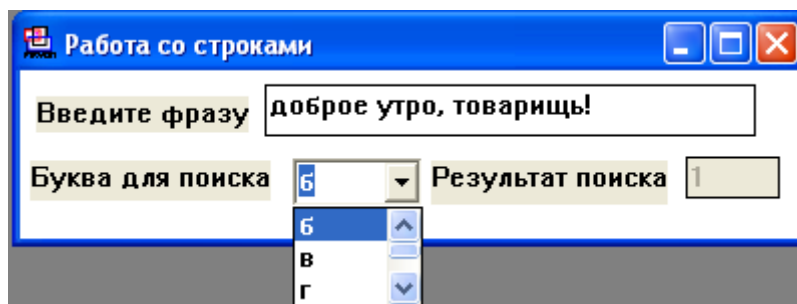


Рис. 16. Результат выполнения программы

Лабораторная работа №6. Обработка файлов в Visual Prolog

Цели работы: изучить теоретические основы работы с файлами в языке логического программирования Visual Prolog и научиться их использовать для решения задач, связанных с обработкой текстовых файлов.

Порядок выполнения:

1. Изучить теоретические основы работы с файлами в языке логического программирования Visual Prolog.
2. Разработать программу на языке логического программирования Visual Prolog, реализующую индивидуальное задание.
3. Написать отчет, содержащий текст программы, реализующей индивидуальное задание.

Задания для самостоятельного выполнения

1. Разработать программу, которая обеспечивает выбор заданного пользователем файла и удаление в каждой его строке лишних пробелов между словами.
2. Разработать программу, которая обеспечивает выбор заданного пользователем файла и удаление в каждой его строке самого короткого слова.
3. Разработать программу, которая обеспечивает выбор заданного пользователем файла и добавление в начало каждой его строки среднего арифметического всех целых чисел, встречающихся в данной строке.
4. Разработать программу, которая обеспечивает выбор заданного пользователем файла и добавление в начало каждой его строки среднего арифметического длин слов, встречающихся в данной строке.
5. Разработать программу, которая обеспечивает выбор заданного пользователем файла и вставку в начало каждой его строки слова,

составленного из всех встречающихся в строке символов латинского алфавита.

6. Разработать программу, которая обеспечивает выбор заданного пользователем файла, удаление всех чисел, встречающихся в выбранном файле и вставку в начало каждой строки количества удаленных в ней чисел.
7. Разработать программу, которая обеспечивает выбор пользователем двух файлов и их объединение в третий, заданный пользователем файл, со вставкой признака конца строки после каждого предложения, заканчивающегося знаком «.».
8. Разработать программу, которая обеспечивает выбор заданного пользователем файла и производит в нем замену последовательности определенного количества пробелов, заданного пользователем, на символ табуляции.
9. Разработать программу, которая обеспечивает выбор заданного пользователем файла и производит удаление в каждой его строке слов, начинающихся с символов "au".
10. Разработать программу, которая обеспечивает выбор заданного пользователем файла и производит удаление в каждой его строке слов, заканчивающихся символами "ai".
11. Разработать программу, которая обеспечивает выбор заданного пользователем файла и удаление в каждой его строке самого длинного слова.
12. Разработать программу, которая обеспечивает выбор заданного пользователем файла и удаление в каждой его строке слов, встречающихся в строке только один раз

Пример выполнения работы

Задание. Разработать программу, позволяющую редактировать текстовые файлы.

Порядок выполнения работы:

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 «Работа с файлами».
4. Нажмем кнопку «Create», тем самым откроем окно проекта.
5. Добавим новый пункт «Тест\Работа» с файлами в меню «родительского» окна.

6. Новый пункт меню будет открывать диалоговое окно, которое создадим, перейдя на вкладку «Window» и нажав кнопку «New».
7. Назовем окно «editor».
8. Добавим в обработчик пункта меню «Тест\Работа с файлами» следующую строку:

```
task_win_eh(_Win,e_Menu(id_Тест_работа_с_файлами,_ShiftCtlAlt),0):-!,
```

```
FILENAME=dlg_GetOpenFileName("*.txt",["*.txt","*.txt"],"Title"),
FILENAME<>"",
retractall(edit_window(_)),
assert(edit_window(FILENAME)),
win_editor_Create(_Win),!
```

9. Изменим предикат создающий окно «editor» следующим образом:

```
win_editor_Create(_Parent):-
CRCT = win_GetClientRect(Parent),
CRCT = rct(L,T,R,B),
L1 = L + 50,
T1 = T + 50,
R1 = R - 50,
B1 = B - 50,
edit_window(FileName),
existfile(FileName),
file_str(FileName,Text),
Font = font_Create(ff_Fixed,[],10),
ReadOnly = b_false, Indent = b_true, InitPos = 1,
edit_Create(win_editor_WinType,rct(L1,T1,R1,B1),FileName,
win_editor_Menu,Parent,win_editor_Flags,Font,ReadOnly,
Indent,Text,InitPos,win_editor_eh).
```

10. При завершении работы с файлом изменим обработчик события «e_Destroy»:

```
win_editor_eh(_Win,e_Destroy,0):-!,
Text = edit_GetText(_Win),
Rezult=dlg_Ask("Сохранить файл?","Да","Нет"),
Rezult=0,
edit_window(FileName),
file_str(FileName,Text),
retract(edit_window(FileName)),!
```

11. Перейдем в начало файла (редактор программного кода) и перед первым разделом «predicates» включим раздел «FACTS» (рис. 17)

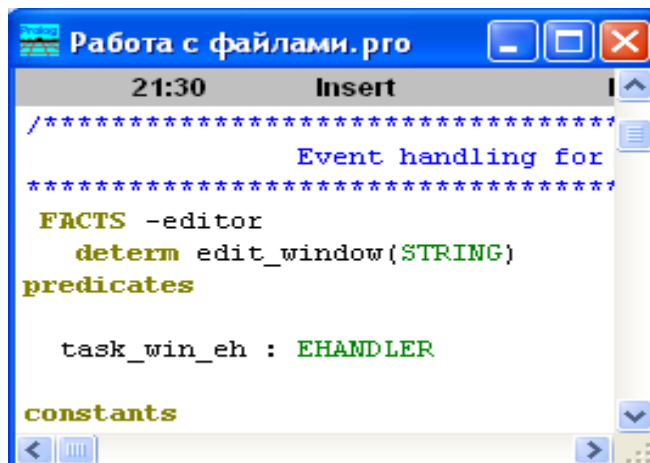


Рис.17. База знаний

12. Запустим программу. Результат работы программы показан на рис. 18.

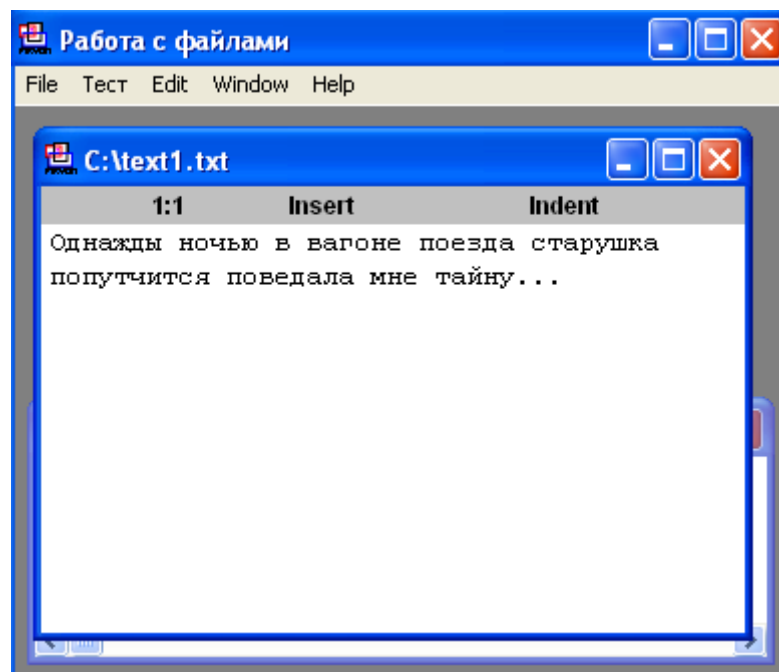


Рис. 18. Окно приложения «Работа с файлами»

Рекомендуемая литература

1. Бухаров М.Н. Интеллектуальные информационные системы: Учебное пособие по специальности 351400 "Прикладная информатика в экономике" / М. Н. Бухаров, Моск. ун-т потреб. кооперации. - М.: МУПК, 2003. - 36 с.
2. Гаврилова Т.А. Базы знаний интеллектуальных систем: Учебное пособие для вузов / Т. А. Гаврилова, В. Ф. Хорошевский. - СПб. [и др.] : Питер, 2001. - 382 с.
3. Тельнов Ю.Ф. Интеллектуальные информационные системы в экономике : Учебное пособие для вузов / Ю. Ф. Тельнов. - 3-е изд., расшир. и доп. - М. : СИНТЕГ, 2002. - 306 с.
4. Смирнова Г.Н. и др. Проектирование экономических информационных систем: Учебник / Г.Н.Смирнова, А.А.Сорокин, Ю.Ф.Тельнов; Под ред. Ю.Ф. Тельнова. - М.: Финансы и статистика, 2001. - 512 с.
5. Бухаров М.Н. Разработка и стандартизация программных средств информационных технологий: Учебное пособие. - М.: МУПК, 2002. -68 с.
6. Бухаров М.Н. Применение стандартов в разработке прикладных программ в экономике: Лекции. - М.: МУПК, 2002. - 32 с.
7. Информационные технологии. Методы классификации: Методические указания к лабораторным работам и самостоятельной работе студентов / Сост. В.И. Возяков, В.П. Филиппов – Чебоксары: Салика, ЧКИ МУПК, 2004. – 36 с.
8. Джексон П. Введение в экспертные системы.— М.:Вильямс, 2001. — 624 с.

Учебное издание

**ИНТЕЛЛЕКТУАЛЬНЫЕ
ИНФОРМАЦИОННЫЕ СИСТЕМЫ**

Методические указания к выполнению лабораторных работ и самостоятельной работы студентов специальности 080801 «Прикладная информатика (в экономике)» очной формы обучения.

Редактор А.Г.Семенова

Подписано в печать2007. Формат 60x84/16.
Бумага офсетная. Печать оперативная. Усл. печ. л.
Тираж экз. Заказ №

РИО ЧКИ РУК,
428025, Чебоксары, ул. Горького, 24