Лабораторная работа №5

тема: " Обработка строковых данных в Visual Prolog**"**

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

Порядок выполнения:

1. Разработать программу на языке Пролог в соответствии с вариантом задания..
2. Написать отчет, содержащий тексты программы, реализующей индивидуальное задание.

Задания для самостоятельного выполнения

1) Подсчитать количество слов в заданной строке.
2) Подсчитать количество букв “а” в последнем слове заданной строки.
3) Найти количество слов в заданной строке, начинающихся с буквы “б”.
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 "а" на букву "е", если "а" стоит на четной позиции в слове, и заменить букву "б" на сочетание "ак", если буква "б" стоит на нечетной позиции в слове.
7) В заданной строке, содержащий от 2 до 30 слов, в каждом из которых от 2 до 10 латинских букв; между соседними словами - не менее одного пробела. 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 "А", 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 "а" в слове.
12) Составить программы для перевода арабских чисел в римские и для обратной операции.
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 – палиндромомы (одинаково читаются слева направо и справа – налево)?

Пример выполнения работы

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

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 кнопочку «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. 11).

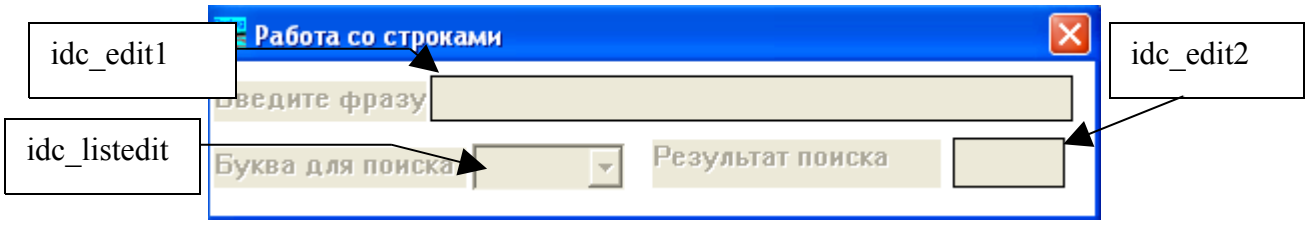

Рис. 11. Макет диалогового окна

9. После того как компонента ListEdit размещена на окне, ее необходимо растянуть так, как показано на рис. 12.

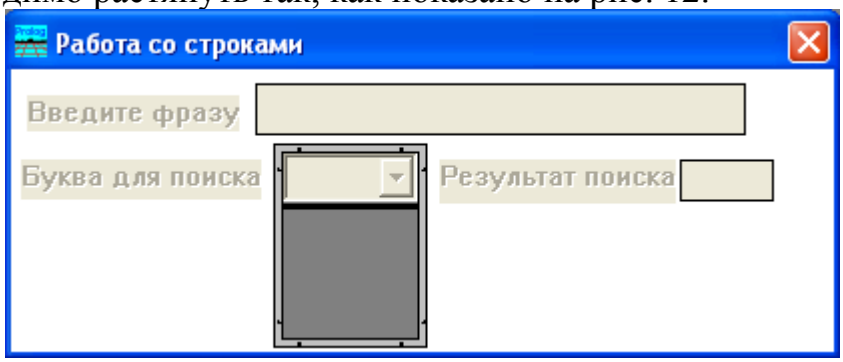

Рис. 12. Формирование выпадающего списка

10. Выберем пункт Control Attributes, используя контекстное меню объекта idc_edit2.
11. В открывшемся диалоговом окне EditText Attrributes нажмем кнопочку Control Flags.. и включим свойство Disabled, тем самым запретим редактирование в данном текстовом окне.
12. Закроем диалоговые окна свойств, нажав кнопочку 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. 2 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 кнопочку 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 */
```

1. Обработаем событие изменения компоненты idc_listedit. Для этого в окне приложения рис. 2 нажмем на кнопку Code Expert.
2. Из списка Event Type выберем пункт Control, а в списке Event or Item выберем строку selchanget . Нажмем сначала на кнопку Update Code, затем на Add Clause и Edit Clause.

17. В открывшемся окне редактирования кода справа от правила win_window_eh(_Win,e_Control(idc_listedit,_CtrlType,_CtrlWin,selchanged),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),!.
```

18. 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),!.
```

19. Пример выполнения программы можно видеть на рис. 13.

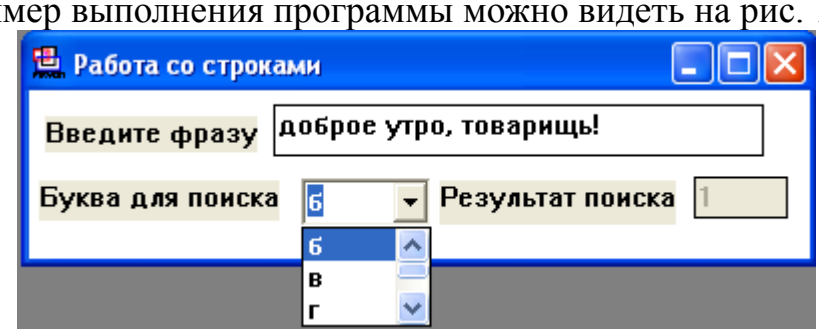

Рис. 13. Результат выполнения программы