

ЛАБОРАТОРНАЯ РАБОТА №4. ПРОГРАММИРОВАНИЕ АЛГОРИТМОВ ЦИКЛИЧЕСКОЙ СТРУКТУРЫ

1. ЦЕЛЬ РАБОТЫ

Целью лабораторной работы является приобретение умений и навыков практического освоения современных технологий визуального объектно-ориентированного программирования в свободно распространяемой среде разработки приложений Lazarus на примере создания приложений, имеющих циклическую структуру с использованием операторов цикла языка программирования FREE Pascal.

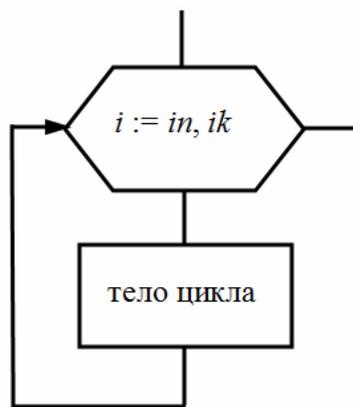
2. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Циклические конструкции обеспечивают многократное выполнение одной и той же последовательности операторов, которая называется *телом цикла*. Существуют два вида элементарных циклических структур:

- циклы с параметром;
- циклы с условием.

2.1. ЦИКЛЫ С ПАРАМЕТРОМ

Циклы с параметром (со счетчиком) используются тогда, когда количество повторов тела цикла заранее известно. Схематично такой цикл может быть изображен следующим образом:



Здесь i – параметр цикла порядкового типа (целого, символьного, логического);

in и ik – выражения, определяющие начальное и конечное значение i ;

<тело цикла> – простой или составной оператор (группа операторов, ограниченная операторными скобками Begin ... End).

Параметр цикла i , начальное in и конечное значения ik должны быть одинакового типа.

Цикл с параметром повторяется пока значение параметра (счетчика) цикла i лежит в интервале $[in, ik]$. Значения in и ik вычисляются только один раз в начале цикла.

В языке Pascal циклы с параметром реализуются с помощью оператора For. Он может быть представлен в двух форматах:

- 1) **For** < i >:=< in > **To** < ik > **Do** <Тело цикла>;
- 2) **For** < i >:=< in > **DownTo** < ik > **Do** <Тело цикла>;

В первом варианте при каждом повторении цикла значение параметра увеличивается на 1, во втором – уменьшается на 1.

Если начальное значение параметра цикла превышает конечное значение, при шаге цикла равным 1, то тело цикла ни разу не будет выполнено.

Пример 1. Вычислить сумму конечного ряда $y = \sum_{k=1}^n \frac{k+0.3}{3k^2+5}$. т.е. сумму первых n членов последовательности, k -ый член которой вычисляется по формуле $y_k = \frac{k+0.3}{3k^2+5}$.

```
Var
  k,n: integer;
  y,s: real;
Begin
  n:=StrToInt(Edit1.Text);
  s:=0;
  {Организация цикла по числу членов последовательности}
  For k:=1 To n Do
    Begin
      y:=(k+0.3)/(3*k*k+5); {Очередной член последовательности}
      s:=s+y; {Вычисление суммы}
    End;
```

```
Edit2.Text:=FloatToStr(s); {Вывод результирующей суммы}  
End;
```

Пример 2. Протабулировать функцию $y=\sin(x)$ на отрезке $[a, b]$ с шагом h . Результат табуляции вывести в компоненту класса TLabel.

Фрагмент исходного кода:

```
Procedure TForm1.Button2Click(Sender: TObject);  
Var  
  a,b,h,x,y: real;  
  n, i: integer;  
  st:string;  
Begin  
  a:=StrToFloat(Edit1.Text); {Начало отрезка}  
  b:=StrToFloat(Edit2.Text); {Конец отрезка}  
  h:=StrToFloat(Edit3.Text); {Шаг табуляции}  
  n:=Ceil((b-a)/h)+1;      {Расчет количества точек}  
  st:='';  
  x:=a;  
  For i:=1 To n Do  
    Begin  
      y:=sin(x);  
      {Строка st заполняется значениями x и y}  
      st:=st+FloatToStr(x)+' '+FloatToStr(y)+chr(13)+chr(10);  
      x:=x+h;  
    End;  
  Label1.Caption:=st; {Вывод строки st на метку}  
End;
```

Замечание. Для того чтобы воспользоваться функцией Ceil, необходимо подключить библиотечный модуль Math в разделе Uses.

2.2. ЦИКЛЫ С УСЛОВИЕМ

Циклы с условием используются тогда, когда число повторений заранее неизвестно, но задано условие окончания цикла. Причем, если условие окончания цикла проверяется перед выполнением тела цикла, то такие циклические структуры называют *циклами с предусловием* («Выполнять пока»), а если проверка условия происходит после выполнения тела цикла – циклами *с постусловием* («Выполнять до тех пор, пока не»).

Схематично такие циклы могут быть изображены следующим образом:

Цикл с предусловием	Цикл с постусловием
 <p data-bbox="207 884 797 1031">Повторение действий происходит пока логическое выражение, которое проверяется <i>в начале</i> цикла, остается <i>истинным</i></p>	 <p data-bbox="823 884 1412 1031">Повторение действий происходит пока логическое выражение, которое проверяется <i>в конце</i> цикла, остается <i>ложным</i></p>

В языке Pascal *циклы с предусловием* (цикл – пока) реализуются с помощью оператора While, а *циклы с постусловием* – с помощью оператора Repeat ... Until.

Формат *оператора цикла с предусловием* While:

```
While <логическое выражение> Do <тело цикла>;
```

Значение логического выражения вычисляется перед каждым выполнением тела цикла. Если оно равно True, то тело цикла выполняется, после чего снова вычисляется значение логического выражения. Если результат имеет значение False, происходит выход из цикла.

Формат *оператора цикла с постусловием* Repeat ... Until:

```
Repeat
    < тело цикла >
Until < логическое выражение >;
```

Вначале выполняется тело цикла, затем проверяется условие выхода из цикла. Если значение логического выражения имеет значение `False`, тело цикла выполняется еще раз, если `True` – происходит выход из цикла. Тело цикла с постусловием выполняется всегда хотя бы один раз.

Замечание 1. Тело циклов может быть простым или составным оператором.

Замечание 2. Чтобы циклы с предусловием или с постусловием успешно завершались необходимо, чтобы в теле цикла был хотя бы один оператор, изменяющий значения, входящих в логическое выражение переменных.

Замечание 1. Использование операторных скобок `begin ... end` для ограничения тела цикла с постусловием не требуется.

Пример 3. Протабулировать функцию:

$$y = e^{x/4} \cos^2 x \quad \text{для } x \in [0; 6] \text{ с шагом } 0,5.$$

Фрагмент исходного кода и форма приведены на рис 17, 18. Для реализации алгоритма используется оператор цикла с постусловием. Результат табуляции выводится в компоненту класса `TLabel`.

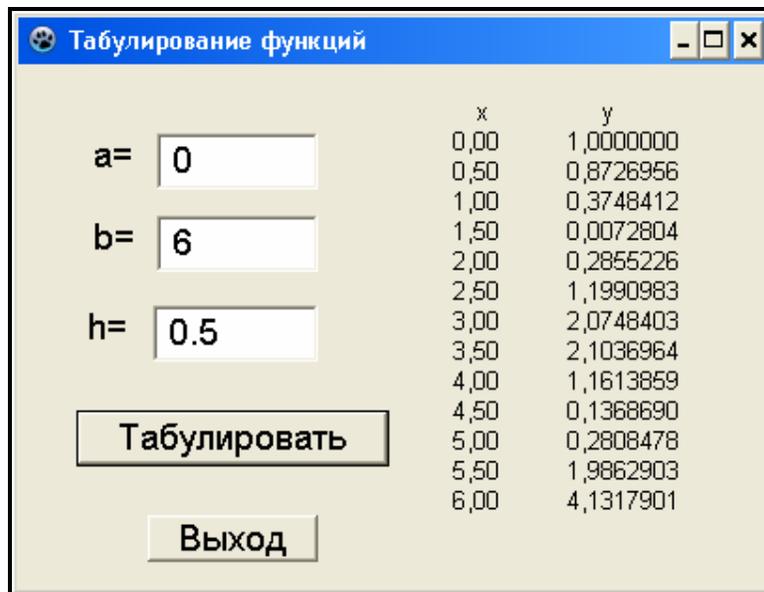


Рис. 17. Пример табулирования функции (Вывод в компоненту класса `TLabel`)

```

procedure TForm1.Button1Click(Sender: TObject);
var x,y,a,b,h:real;
    begin
        a:=StrToFloat(Edit1.Text); {начало отрезка}
        b:=StrToFloat(Edit2.Text); {конец отрезка}
        h:=StrToFloat(Edit3.Text); {шаг табуляции}
        Label1.Caption:='      x              y      '
                        + Chr(13)+Chr(10);

        x:=a;
        Repeat
            y:=exp(x/4)*sqr(cos(x));
            Label1.Caption:= Label1.Caption +
                            FloatToStrf(x,ffixed,5,2)+
                            '      ' +
                            FloatToStrf(y,ffixed,5,7)+
                            Chr(13)+Chr(10);

            x:=x+h;
        Until x > b;
    end;

```

Рис. 18. Программный код табулирования функции

2.3. ВВОД ДАННЫХ ИЗ ДИАЛОГОВОГО ОКНА С ПОМОЩЬЮ ФУНКЦИИ INPUTBOX

Окно ввода – это стандартное диалоговое окно, которое появляется на экране в результате вызова функции InputBox. В общем виде оператор ввода данных с использованием этой функции записывают так:

```
Имя := InputBox(Заголовок, Приглашение, НачЗначение);
```

где

Заголовок – это надпись в строке заголовка окна ввода,

Приглашение – текст поясняющего сообщения – подсказка пользователю, какую информацию он должен ввести в специальное поле ввода, находящееся в этом окне,

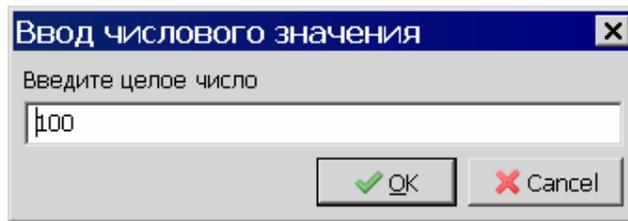
НачЗначение – пустая строка или значение, которое будет введено автоматически, если пользователь с этим согласится.

Имя – переменная *строкового* типа, которой будет присвоено значение из поля ввода.

Например, событийная процедура:

```
procedure TForm1.Button1Click(Sender: TObject);  
Var  
  x:integer;  
Begin  
  x:=StrToInt(InputBox('Ввод числового значения',  
                        'Введите целое число', '100'));  
  Edit1.Text:=IntToStr(x);  
End;
```

вызывает появление диалогового окна



После того, как пользователь введет значение в поле ввода этого окна и нажмет кнопку ОК, это число будет выведено в компоненту Edit1.

3. ПРАКТИЧЕСКАЯ ЧАСТЬ

3.1. ПРИМЕРЫ ПРИЛОЖЕНИЙ ЦИКЛИЧЕСКОЙ СТРУКТУРЫ

Самостоятельно создайте формы для примеров, приведенных ниже. Изучите и наберите в разделе операторов обработки событий приведенные ниже готовые программы. Запустите созданные приложения на выполнение. Проанализируйте результат.

Задание 1. Вычислить сумму конечного ряда:

$$S = \sum_{k=1}^n (-1)^{k+1} \frac{k}{3(k+1)^2},$$

т.е. сумму первых n членов последовательности, k -ый член которой вычисляется по формуле $y_k = \frac{k+0.3}{3k^2+5}$ ($k=1,2,3,\dots,n$).

Для вычисления суммы конечного ряда S целесообразно организовать цикл с параметром. В теле цикла вычисляется значение очередного слагаемого и осуществляется накопление суммы прибавлением полученного слагаемого к сумме всех предыдущих. Перед началом цикла значение переменной S обнуляется.

Фрагмент исходного кода:

```
Procedure TForm1.Button2Click(Sender: TObject);  
var  
    S: real;  
    n,z,k: integer; {переменная z используется для смены знака}  
Begin  
    n:=StrToInt(InputBox('Ввод количества слагаемых,  
        'Введите количество слагаемых', ''));  
    S:=0;  
    z:=1; {Знак положительный}  
    For k:=1 To n Do  
        Begin  
            a:=z/(k*k);  
            S:=S+a;  
            z:=-z; {меняем знак на противоположный}  
        End;  
    Edit1.Text:=FloatToStr(S); {Вывод суммы}  
End;
```

Задание 2. Подсчитать количество цифр в заданном целом числе.

Для подсчета количества цифр в числе организован цикл с условием, в теле которого каждая цифра из числа выделяется с помощью операции целочисленного деления `div` на 10. Тело цикла состоит из одного оператора.

Фрагмент исходного кода:

```
Procedure TForm1.Button1Click(Sender: TObject);  
var  
    a,k: integer; {переменная k – количество цифр в числе}  
Begin  
    a:=StrToInt(Edit1.Text);  
    k:=0;  
    while a<>0 do a:=a div 10;  
    Edit1.Text:=IntToStr(k);  
End;
```

Использование рекуррентного соотношения для вычисления суммы ряда

Рекуррентная формула – это формула, выражающая каждый член последовательности через предыдущие. Рекуррентная формула для нахождения суммы ряда $s = \sum_{i=1}^n a_i$ используется в целях избавления от больших чисел при вычислении слагаемых.

Как правило, в этом случае рекуррентная формула имеет вид: $a_{k+1} = c \cdot a_k$, откуда, $c = \frac{a_{k+1}}{a_k}$. Таким образом, вычислив первое слагаемое, можно найти все остальные слагаемые.

Задание 3. Вычислить сумму конечного ряда:

$$S = \sum_{k=1}^n (-1)^k \frac{1}{k!(k+1)!}$$

Так как знаменатель слагаемых этой суммы может быть довольно большим числом, воспользуемся рекуррентным соотношением для вычисления очередного слагаемого. Найдем коэффициент c , разделив $(k+1)$ -ое слагаемое на k -ое:

$$c = \frac{(-1)^{k+1} \cdot k!(k+1)!}{(k+1)! \cdot (k+2)! (-1)^k} = -\frac{k!}{(k+2)!} = -\frac{1}{(k+1)(k+2)}$$

Вычислим первое слагаемое ($k=1$):

$$a_1 = -\frac{1}{2}.$$

Остальные слагаемые вычисляются по формуле:

$$a_{k+1} = c \cdot a_k, \quad \text{где } c = -\frac{1}{(k+1)(k+2)}.$$

Фрагмент исходного кода:

```
Procedure TForm1.Button2Click(Sender: TObject);  
var  
    S, a, c: real;  
    k: integer; {переменная z используется для смены знака}  
Begin  
    n:=StrToInt(Edit1.Text);  
    S:=0;  
    a:=-1/2; {вычисляется первое слагаемое}  
    For k:=1 To n Do  
        Begin  
            S:=S+a;  
            c:=-1/((k+1)*(k+2)); {коэффициент c}  
            a:=a*c; {очередное слагаемое}  
        End;  
    Edit2.Text:=Floattistr(S); {Вывод суммы}  
End;
```

Задание 4. Вычислить сумму бесконечного ряда:

$$S = \sum_{k=1}^{\infty} (-1)^k \frac{1}{k!(k+1)!} \text{ с точностью } \varepsilon = 0.001.$$

Задание аналогично предыдущему, но поскольку в этом задании не задано количество слагаемых, то использование цикла с параметром здесь не возможно. Но так как задано условие, то для реализации вычисления суммы бесконечного ряда целесообразно использовать циклы с условием. Условие окончания цикла – достижение заданной точности: вычисление продолжается до тех пор, пока модуль очередного слагаемого не станет меньше заданного $\varepsilon = 0.001$.

Фрагмент исходного кода:

```
Procedure TForm1.Button2Click(Sender: TObject);  
var  
    S, a, c, eps: real;  
    k: integer;  
Begin  
    Eps:=StrToFloat(Edit1.Text);  
    S:=0;  
    a:=-1/2; {вычисляется первое слагаемое}  
    k:=1;    {номер первого слагаемого}  
    Repeat  
        S:=S+a;  
        c:=-1/((k+1)*(k+2)); {коэффициент c}  
        a:=a*c; {очередное слагаемое}  
        k:=k+1; {очередной номер следующего слагаемого}  
    Until abs(a) < Eps; {пока условие остается ложным цикл повторяется}  
  
    Edit2.Text:=FloatToStr(S); {Вывод суммы}  
End;
```

Задание 5.

1. Вычислить сумму бесконечного ряда для заданного аргумента x :

$$S = \sum_{k=1}^{\infty} (-1)^{k+1} \frac{2^{2k-1} x^{2k}}{(2k)!} = x^2 - \frac{2^3 x^4}{4!} + \frac{2^5 x^6}{6!} - \dots$$

с точностью ε . Этот ряд является разложением функции $y = \sin^2(x)$. Значение аргумента x задается.

2. Найти абсолютную и относительную погрешность вычисления.

Абсолютная погрешность вычисляется по формуле:

$$\Delta = |S - y|.$$

Относительная погрешность вычисляется по формуле:

$$\partial = \frac{\Delta}{S}.$$

Для вычисления очередного слагаемого нужно использовать рекуррентное соотношение (проверьте данное соотношение!):

$$a_{k+1} = -\frac{4x^2}{(2k+1)(2k+2)} a_k.$$

Первое слагаемое: $a_1 = x^2$.

Фрагмент исходного кода и формы (рис. :

```
procedure TForm1.Button1Click(Sender: TObject);  
var  
    S, a, c, eps, x, y, a_d, o_d: real;  
    k: integer;  
Begin  
    x:= StrToFloat(Edit1.Text);  
    Eps:=StrToFloat(Edit2.Text);  
    S:=0;  
    a:=x*x; {вычисляется первое слагаемое}  
    k:=1;  
    Repeat  
        S:=S+a;  
        c:=-4*x*x/((2*k+1)*(2*k+2)); {коэффициент c}  
        a:=a*c; {очередное слагаемое}  
        k:=k+1; {очередной номер следующего слагаемого}  
    Until abs(a)<Eps; {пока условие остается ложны цикл повторяется}  
    y:=sqr(sin(x)); {значение функции}  
    a_d:=abs(s-y); o_d:=a_d/s; {вычисление погрешности}  
    If CheckBox1.Checked Then  
        begin Edit7.Text:=FloatToStrF(s, ffFixed, 4, 8);  
            Edit6.Text:=FloatToStrF(y, ffFixed, 7, 8); end  
        Else  
        begin Edit7.Text:=FloatToStr(S);  
            Edit6.Text:=FloatToStr(y); end;  
    Edit9.Text:=FloatToStr(a_d); {абсолютная погрешность}  
    Edit8.Text:=FloatToStr(o_d); {относительная погрешность}  
End;
```

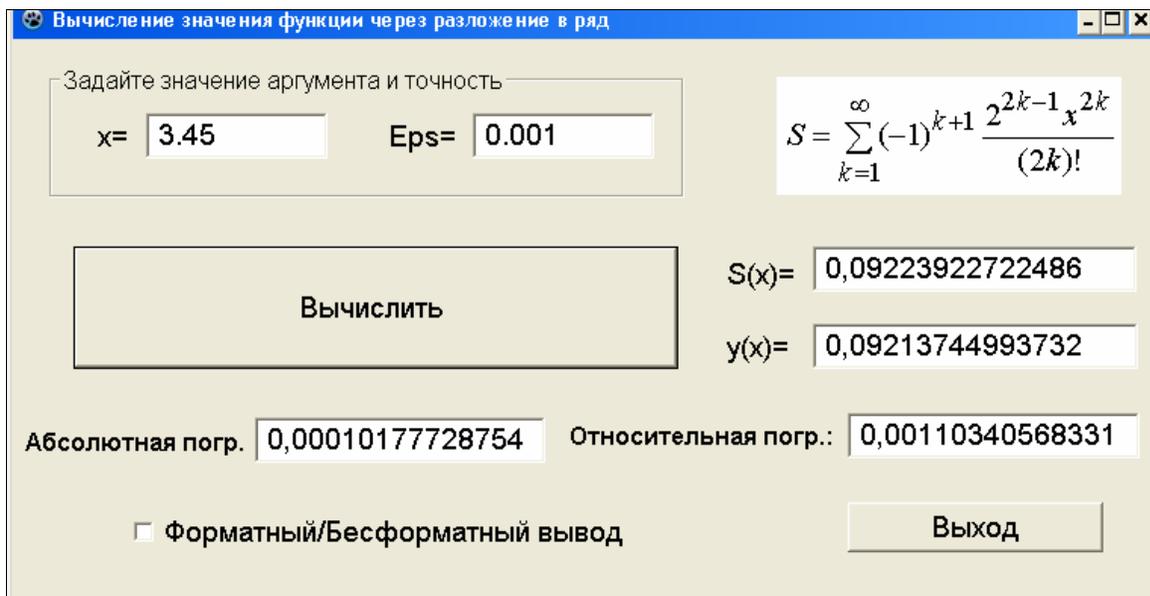


Рис. 19. Пример вычисления суммы бесконечного ряда с помощью рекуррентной формулы

3.2. КОМПОНЕНТЫ LAZARUS, ИСПОЛЬЗУЕМЫЕ В ПРИЛОЖЕНИЯХ С ЦИКЛИЧЕСКИМИ КОНСТРУКЦИЯМИ

Компонент Мемо (многострочное окно редактирования) . Используется для ввода, отображения и редактирования многострочных текстов. Относится к группе Standard.

Основные свойства компонента Memo :

Свойство	Описание
Lines	Определяет текст, который будет выведен построчно в окне Мемо при запуске программы. Текст может быть задан на этапе проектирования формы в Диалоге ввода строк, который откроется, если в Инспекторе объектов выбрать свойство Lines и щелкнуть на значок 
MaxLength	Позволяет ограничивать число вводимых пользователем символов
ScrollBars	Задаёт наличие полос прокрутки

Метод Clear, примененный к компоненту Мемо очищает его содержимое (удаляет все строки).

На этапе выполнения самую первую строку можно задать в Мемо оператором:

```
memo1.Lines[0]:='<текст первой строки>';
```

Остальные строки на этапе выполнения добавляются в компонент Мемо с помощью метода Add, примененного к свойству Lines (без указания номера):

```
memo1.Lines.Add '<текст добавляемой строки> ';
```

Ниже приведен программный код и форма (рис. 20) использования компонента Мемо на примере табулирования функции.

```
Procedure TForm1.Button1Click(Sender: TObject);  
Var x,y,a,b,h:real;  
  
Begin  
  a:=StrToFloat(Edit1.Text); {начало отрезка}  
  b:=StrToFloat(Edit2.Text); {конец отрезка}  
  h:=StrToFloat(Edit3.Text); {шаг табуляции}  
  x:=a;  
  Memo1.Clear;  
  {Формирование заголовка таблицы}  
  Memo1.Lines[0]:='      x                y';  
  Memo1.Lines.Add ('_____');  
  
  Repeat  
    y:=exp(x/4)*sqr(cos(x));  
    Memo1.Lines.Add (FloatToStrf(x,ffixed,5,2)+  
                      '      ' +  
                      FloatToStrf(y,ffixed,5,7));  
    x:=x+h;  
  Until x > b;  
  
End;
```

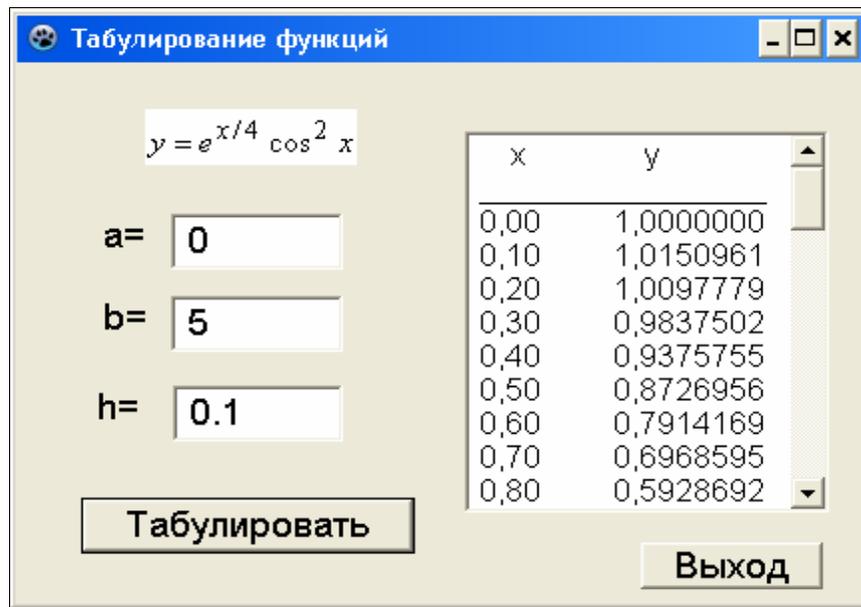


Рис. 20. Пример формы табулирования функции с компонентой Memo

Компонент TListBox  (Список). Относится к группе Standard. Элементами списка являются строки (String). Строки нумеруются, начиная с нуля.

Основные свойства компонента ListBox:

Свойство	Описание
Items	Определяет текст, который будет выведен построчно в списке. На этапе проектирования формы текст может быть задан в Диалоге ввода строк, если выбрать свойство Items и щелкнуть на значок  . Items[i] – доступ к элементу с индексом i
Count	Общее количество элементов в компоненте
MultiSelect	Признак множественного выбора. Если MultiSelect = true, то разрешается выбор одновременно нескольких элементов
ExtendedSelect	Если свойство равно true, то выбор нескольких элементов можно производить стандартным способом (с помощью клавиш Shift и Ctrl)
Selected[i]	Если выбран элемент с индексом i, то значение этого свойства равно true

ItemIndex	Индекс выбранного элемента. Если не выбран ни один, то значение свойства равно -1. Если выбрано несколько элементов (MultiSelect=True), то – индекс элемента, на котором установлен фокус
Sorted	Если значение свойства равно true, то элементы компонента автоматически сортируются

Строки на этапе выполнения добавляются с помощью метода Add, примененного к свойству Items (без указания номера):

```
Listbox1.Items.Add '<текст добавляемой строки> ';
```

Ниже приведен пример использования компонента Listbox на примере табулирования функции (рис. 21):

```

procedure TForm1.Button1Click(Sender: TObject);
var
  a, b, h, x, y: real;
begin
  a:=StrToFloat(Edit1.Text);  {начало отрезка}
  b:=StrToFloat(Edit2.Text);  {конец отрезка}
  h:=StrToFloat(Edit3.Text);  {шаг табуляции}
  x:=a;
  ListBox1.Clear;
  Repeat
    y:=exp(x/4)*sqr(cos(x));
    ListBox1.Items.Add (FloatToStrf(x, ffixed, 5, 2) +
      ' ' +
      FloatToStrf(y, ffixed, 5, 7));

    x:=x+h;
  Until x > b;
  {количество элементов в списке}
  Edit4.Text:=IntToStr(ListBox1.Count);
end;

procedure TForm1.Button2Click(Sender: TObject);
begin {номер выбранного пользователем элемента}
  Edit5.Text:=IntToStr(ListBox1.ItemIndex);
end;

```

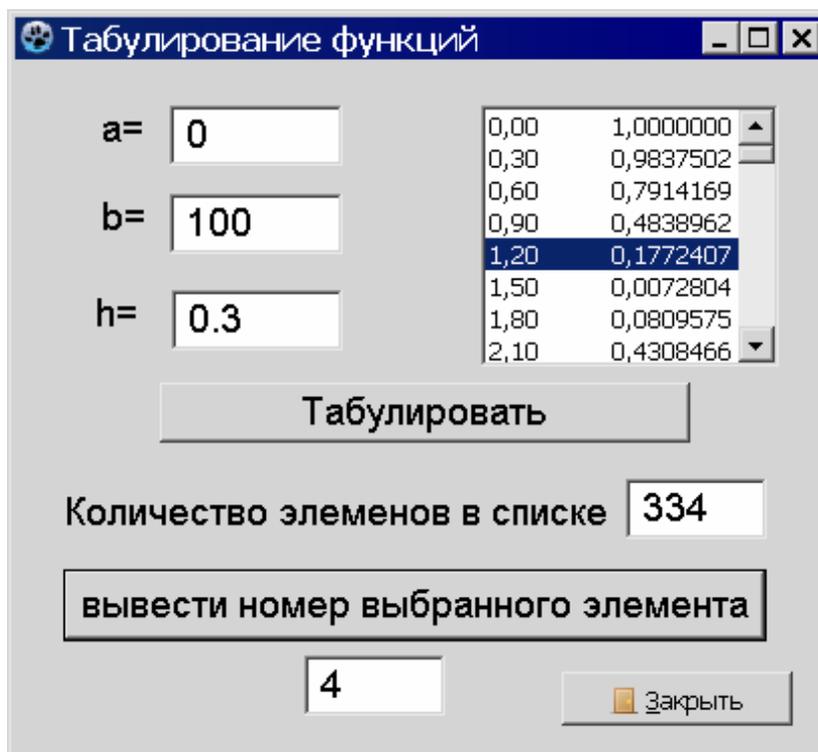


Рис. 21. Пример формы табулирования функции с компонентой `ListBox`

Компонент `TComboBox`  (Раскрывающийся список). Методы работы с таким списком практически не отличаются от `Listbox`. По сравнению с `Listbox` у него есть преимущество, заключающееся в том, что элементы списка можно редактировать, удалять, а также добавлять в список новые элементы. Кроме того, компонент позволяет сэкономить пространство, когда на форме расположены много компонентов. К недостаткам можно отнести то, что в нем нельзя выбрать одновременно несколько элементов.

Основные свойства компонента `ComboBox` :

Свойство	Описание
Items	Определяет текст, который будет выведен построчно в списке. На этапе проектирования формы текст может быть задан в Диалоге ввода строк, если выбрать свойство Items и щелкнуть на значок  . Items[i] – доступ к элементу с индексом i
Selected[i]	Если выбран элемент с индексом i, то значение этого свойства равно true
ItemIndex	Индекс выбранного элемента. По умолчанию в момент проектирования ItemIndex = -1, но чаще всего устанавливают ItemIndex = 0, в этом случае в окно ComboBox будет выведен первый элемент списка
Text	Содержит выбранный элемент списка или вновь введенную строку
Sorted	Если значение свойства равно true, то элементы компонента автоматически сортируются

Ниже приведен пример использования компонента ComboBox. Для свойства Sorted выбрано значение True, основной список сформирован в свойстве Items с помощью Редактора ввода строк.

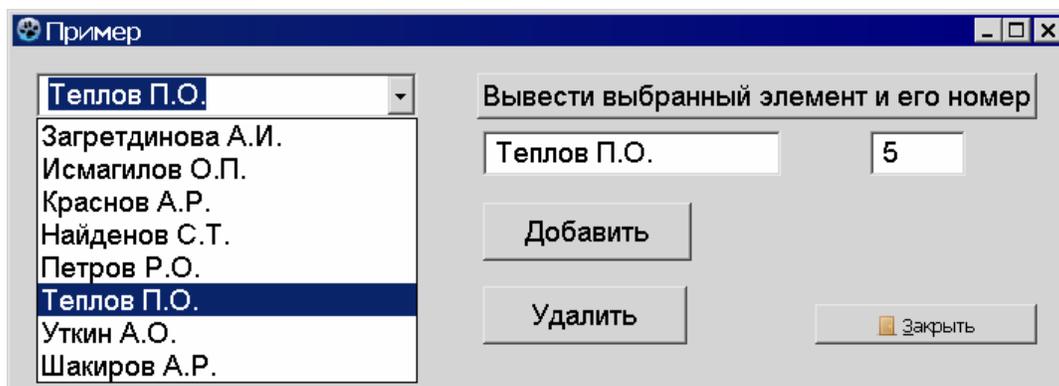


Рис. 22. Пример формы с компонентой ComboBox

```

{ TForm1 }
procedure TForm1.Button2Click(Sender: TObject);
begin
    { вывод номера выбранного пользователем элемента }
    Edit2.Text := IntToStr(ComboBox1.ItemIndex);
    { вывод выбранного пользователем элемента }
    Edit1.Text := ComboBox1.Text;
    { вывод первого элемента }
    Edit3.Text := ComboBox1.Items[1];
end;

procedure TForm1.Button5Click(Sender: TObject);
begin { удалить выбранный элемент }
    ComboBox1.Items.Delete(ComboBox1.ItemIndex);
end;

procedure TForm1.Button4Click(Sender: TObject);
begin { добавить элемент }
    ComboBox1.Items.Add(ComboBox1.Text);
end;

```

4. ЗАДАНИЯ К ЛАБОРАТОРНОЙ РАБОТЕ №4

4.1. ПЕРВЫЙ УРОВЕНЬ СЛОЖНОСТИ

Задача 1. Табулирование функции

№ вар.	Функция	Диапазон изменения аргумента	Шаг
1.	$y = 3 \sin \sqrt{x} + 0,35x - 3,8$	[2,3]	0,1
2.	$y = 0,25x^3 + x - 1,2502$	[0,2]	0,2
3.	$y = x + \sqrt{x} + \sqrt[3]{x} - 2,5$	[0,4;1]	0,05
4.	$y = \sin(\ln x) - \cos(\ln x) + 2 \ln x$	[1;3]	0,2
5.	$y = \cos \frac{2}{x} - 2 \sin \frac{1}{x} + \frac{1}{x}$	[1;2]	0,1
6.	$y = 3x - 4 \ln x - 5$	[2;4]	0,2

№ вар.	Функция	Диапазон изменения аргумента	Шаг
7.	$y = \sqrt{1-x} - \cos \sqrt{1-x}$	[0;1]	0,1
8.	$y = \operatorname{tg} x - \frac{1}{3} \operatorname{tg}^3 x + \frac{1}{5} \operatorname{tg}^5 x - \frac{1}{3}$	[0;0,8]	0,05
9.	$y = 0,1x^2 - x \ln x$	[1;2]	0,1
10.	$y = x - 1/(3 + \sin 3,6x)$	[0;0,85]	0,05
11.	$y = x + \cos(x^{0,52} + 2)$	[0,5;1]	0,05
12.	$y = 3 \ln^2 x + 6 \ln x - 5$	[1;3]	0,2
13.	$y = 3x - 14 + e^x - e^{-x}$	[1;3]	0,2
14.	$y = \sqrt{1-x} - \operatorname{tg} x$	[0;1]	0,1
15.	$y = \cos x - e^{-\frac{x^2}{2}} + x - 1$	[1;2]	0,1
16.	$y = \cos x - \sin \frac{1}{x^2} + \frac{4}{x}$	[1;2]	0,1

Задача 2. Вычисление суммы ряда

По заданной формуле члена последовательности с номером k составить два проекта:

1. вычисление суммы первых n членов последовательности ($k=1,2,3,..,n$);
2. вычисление суммы всех членов последовательности, не меньших заданного числа e .

При создании второго проекта воспользуйтесь первым: создайте новую папку для второго проекта, скопируйте в нее все содержимое папки первого проекта, откройте его, внесите необходимые изменения и сохраните их, нажав на значок дискеты.

№ вар.	Член последовательности
1.	$\frac{1}{(2k-1)(2k+1)}$
2.	$\frac{k}{(k+1)^2+3}$
3.	$\frac{2k}{(k^2+1)(k+2)}$
4.	$\frac{k+1}{k-\sqrt{k+2}}$
5.	$\frac{k}{(3k^2+7)(k^2+1)}$
6.	$\frac{k+4}{(k^2+2)(k+8)}$
7.	$\frac{3(k+1)}{7k^2+9}$
8.	$\frac{1}{\sqrt{k+15}}$

№ вар.	Член последовательности
9.	$\frac{1}{k^2+3k+4}$
10.	$\frac{k+1}{k(k+2)(k+3)}$
11.	$\frac{k+2}{k^2+4}$
12.	$\frac{2k}{3k+k^2+4}$
13.	$\frac{\sqrt{k+1}}{2\sqrt{k^2+1}}$
14.	$\frac{4k}{5k^2+8k-1}$
15.	$\frac{2k+1}{(2k^2+3)k}$
16.	$\frac{(k+1)^2}{k(k+2)}$

Задача 3. Арифметические задачи

1. Определить сумму цифр введенного числа a ($a < 1000$).
2. Определить сумму чисел от 3 до 99 кратных числу 3.
3. Вывести на экран таблицу квадратов целых чисел от 1 до 10.
4. Напечатать таблицы температур по Цельсию от 0 до 100 градусов с дискретностью в один градус и их эквивалентов по шкале Фаренгейта, используя для перевода формулу $t_F = 9 \cdot t_C / 5 + 32$.
5. Вывести в столбец произведения чисел $a = 143$, $b = 777$ и числа c , последовательно принимающего значения 1, 2, 3, ... 9.
6. Получить произведения числа $a = 12345689$ на числа 9, 18, 27, ... 81.

7. Произвести суммирование натуральных чисел 1, 2, 3, ..., пока их сумма s не станет равной или превысит величину h . Вывести на экран последнее слагаемое и значение суммы.
8. Сколько чисел последовательности 2, 4, 6, 8, ... нужно взять, чтобы их сумма превысила 1000? Вывести величину последнего слагаемого и суммы.
9. Определить количество цифр в натуральном числе N .
10. Вычислить факториал натурального числа N .
11. Определить произведение цифр натурального числа N .
12. Найти сумму всех четных натуральных чисел от 1 до 100.
13. Найти сумму первой и последней цифры натурального числа N .
14. Дано натуральное число. Верно ли, что оно начинается и заканчивается одной и той же цифрой.
15. Дано натуральное число. Верно ли, что в данном числе нет данной цифры A . A задается.
16. Дано натуральное число. Верно ли, что оно заканчивается нечетной цифрой.
17. Дано натуральное число. Верно ли, что в данном числе цифра A встречается более двух раз. A задается.
18. Дано натуральное число. Верно ли, что в данном числе сумма цифр больше A , а само число делится на A . A задается.
19. Дано натуральное число. Верно ли, что число принадлежит промежутку от A до B и кратно 3, 4 и 5. A и B задаются.
20. Сколько раз первая цифра встречается в данном числе.

4.2. ВТОРОЙ УРОВЕНЬ СЛОЖНОСТИ

Задача 1. Табулирование функции

Протабулировать функцию y на заданном диапазоне, с заданным шагом изменения аргумента и исходными данными.

№ вар.	Функция	Исходные данные	Диапазон и шаг изменения аргумента
1.	$y = \begin{cases} at^{2\ln t}, & 1 \leq t \leq 2 \\ 1, & t < 1 \\ e^{at} \cos bt, & t > 2 \end{cases}$	$a = -0.5$ $b = 2$	$t \in [0;3]$ $\Delta t = 0.15$

2.	$y = \begin{cases} \pi x^2 - 7/x^2, & x < 1.3 \\ ax^3 + 7\sqrt{x}, & x = 1.3 \\ \lg(x + 7\sqrt{x}), & x > 1.3 \end{cases}$	$a = 1.5$	$x \in [0.8; 2]$ $\Delta x = 0.1$
3.	$y = \begin{cases} ax^2 + bx + c, & x < 1.2 \\ a/x + \sqrt{x^2 + 1}, & x = 1.2 \\ (a + bx)/\sqrt{x^2 + 1}, & x > 1.2 \end{cases}$	$a = 2.8$ $b = -0.3$ $c = 4$	$x \in [1; 2]$ $\Delta x = 0.05$
4.	$y = \begin{cases} \pi x^2 - 7/x^2, & x < 1.34 \\ ax^3 + 7\sqrt{x}, & x = 1.4 \\ \ln(x + 7\sqrt{ x+a }), & x > 1.4 \end{cases}$	$a = 1.65$	$x \in [0.7; 2]$ $\Delta x = 0.1$
5.	$y = \begin{cases} 1.5 \cos^2 x, & x = 1 \\ 1.8ax, & 1 < x < 2 \\ (x-2)^2 + 6, & x > 2 \end{cases}$	$a = 2.3$	$x \in [0.2; 8]$ $\Delta x = 0.2$
6.	$y = \begin{cases} x\sqrt[3]{x-a}, & x > a \\ x \sin ax, & x = a \\ e^{-ax} \cos ax, & x < a \end{cases}$	$a = 2.5$	$x \in [1; 5]$ $\Delta x = 0.5$
7.	$y = \begin{cases} bx - \lg bx, & bx < 1 \\ 1, & bx = 1 \\ bx + \lg bx, & bx > 1 \end{cases}$	$b = 1.5$	$x \in [0.1; 1]$ $\Delta x = 0.1$
8.	$y = \begin{cases} \sin x \lg x, & x > 3.5 \\ \cos^2 x, & x \leq 3.5 \end{cases}$	-	$x \in [2; 5]$ $\Delta x = 0.25$
9.	$y = \begin{cases} \lg(x+1), & x > 1 \\ \sin^2 \sqrt{ax}, & x \leq 1 \end{cases}$	$a = 20.3$	$x \in [0.5; 2]$ $\Delta x = 0.1$
10.	$y = \begin{cases} (\ln^3 x + x^2)/\sqrt{x+t}, & x < 0.5 \\ \sqrt{x+t} + 1/x, & x = 0.5 \\ \cos x + t \sin^2 x, & x > 0.5 \end{cases}$	$t = 2.2$	$x \in [0.2; 2]$ $\Delta x = 0.2$

11.	$y = \begin{cases} \frac{a+b}{e^x + \cos x}, & x < 2.8 \\ (a+b)/(x+1), & 2.8 \leq x < 6 \\ e^x + \sin x, & x \geq 6 \end{cases}$	$a = 2.6$ $b = -0.39$	$x \in [0;7]$ $\Delta x = 0.5$
12.	$y = \begin{cases} a \lg x + \sqrt[3]{ x }, & x > 1 \\ 2a \cos x + 3x^2, & x \leq 1 \end{cases}$	$a = 0.9$	$x \in [0.8;2]$ $\Delta x = 0.1$
13.	$y = \begin{cases} \frac{a}{i} + bi^2 + c, & i < 4 \\ i, & 4 \leq i \leq 6 \\ ai + bi^3, & i > 6 \end{cases}$	$a = 2.1$ $b = 1.8$ $c = -20.5$	$i \in [0;12]$ $\Delta i = 1$
14.	$y = \begin{cases} a \sin\left(\frac{i^2+1}{n}\right), & \sin\left(\frac{i^2+1}{n}\right) > 0 \\ \cos\left(i + \frac{1}{n}\right), & \sin\left(\frac{i^2+1}{n}\right) \leq 0 \end{cases}$	$a = 0.3$ $n = 10$	$i \in [1;10]$ $\Delta i = 1$
15.	$y = \begin{cases} \sqrt{at^2 + b \sin t + 1}, & t < 0.1 \\ at + b, & t = 0.1 \\ \sqrt{at^2 + b \cos t + 1}, & t > 0.1 \end{cases}$	$a = 2.5$ $b = 0.4$	$t \in [-1;1]$ $\Delta t = 0.2$

Задача 2. Вычисление суммы ряда

Составить две программы:

1. программу вычисления суммы первых n членов последовательности ($k=1,2,3,..,n$);
2. программу вычисления суммы всех членов последовательности, не меньших заданного числа e .

№ вар.	Ряд
1.	$\sum_{k=1}^n \frac{k^4}{k!}$

№ вар.	Ряд
9.	$\sum_{k=1}^n \frac{(-1)^{k-1}}{(2k-1)!}$

2.	$\sum_{k=0}^n \frac{(-1)^k}{k!(k+1)!}$
3.	$\sum_{k=0}^n \frac{(-1)^k}{k!}$
4.	$\sum_{k=1}^n (-1)^k \frac{1}{(k!)^2}$
5.	$\sum_{k=0}^n \frac{1}{(k+2)!(k+3)!}$
6.	$\sum_{k=1}^n \frac{k!}{(5+k-1)!}$
7.	$\sum_{k=1}^n \frac{k}{(k+1)!}$
8.	$\sum_{k=0}^n \frac{1}{(2k)!}$

10.	$\sum_{k=0}^n \frac{(-1)^k}{k!} \cdot k$
11.	$\sum_{k=1}^n \frac{(k+1)^3}{k!}$
12.	$\sum_{k=0}^n \frac{(-1)^k}{(2k)!}$
13.	$\sum_{k=0}^n (-1)^k \frac{1}{k!(k+1)!}$
14.	$\sum_{k=0}^n \frac{1}{k!}$
15.	$\sum_{k=1}^n (-1)^k \frac{k}{(k!)^2 + (k+1)!}$
16.	$\sum_{k=1}^n (-1)^{k+1} \frac{1}{k!(k+1)!}$

Задача 3. Арифметические задачи

1. В переменную последовательно вводятся числа. Окончание ввода либо по желанию пользователя, либо когда сумма отрицательных чисел превысит -1000 . Определить среднее арифметическое отрицательных чисел.
2. В переменную последовательно вводятся десять чисел. Определить среднее арифметическое отрицательных чисел.
3. В простую переменную последовательно вводятся N вещественных чисел. Вычислить максимальное значение.
4. В простую переменную последовательно вводятся N чисел. Сколько чисел больше своих соседей слева?
5. В простую переменную последовательно вводятся N чисел. Все ли числа меньше заданного числа K ?
6. Найти наибольшую и наименьшую цифры в записи данного натурального числа.