

## Уважаемые студенты групп!

Вашему вниманию представлена лабораторная работа на тему «Организация программы, содержащие подпрограммы, процедуры - функции».

### Задание

1. Реализовать в системе Delphi приведенный пример выполнения задания, протестировать программу и исправить ошибки.
2. Преобразовать введенную программу в соответствии со своим вариантом.
3. Скриншоты экранных копий выполнения программы предоставить преподавателю не позднее 31.01.

С уважением Ганзенко Ирина Владимировна

!!! Если возникнут вопросы обращаться по телефону 0721134803 (вацап), +79591134803 (телеграмм)

[disobuch.ganzenko2020@mail.ru](mailto:disobuch.ganzenko2020@mail.ru)

## ЛАБОРАТОРНАЯ РАБОТА

### ОРГАНИЗАЦИЯ ПРОГРАММЫ, СОДЕРЖАЩИЕ ПОДПРОГРАММЫ, ПРОЦЕДУРЫ - ФУНКЦИИ

**Цель работы:** получение навыков по использованию функции в алгоритмах и программах.

#### Пример программирования с подпрограммами

Вычислить функцию:

$$S = \frac{\cos x}{1!} + \dots + \frac{\cos Nx}{N!}, \text{ где } N - \text{ количество членов ряда.}$$

Структура программы будет включать в себя два модуля: программу с именем Project1 и модуль с именем Unit1, связанный с формой Form1.

Чтобы продемонстрировать на учебном примере особенности и отличия в использовании процедуры и функции, реализуем подпрограмму вычисления факториала тремя способами: в виде процедуры, в виде функции и рекурсивным методом.

1. Разработка алгоритма (рис. 7.1):

- Входные данные:

а) X – вещественная переменная, являющаяся аргументом функции  $\cos(NX)$ ;

б)  $N$  – целочисленная переменная, обозначающая количество членов ряда.

- Выходные данные:  $S$  – вещественная переменная, значение которой есть сумма членов ряда.

- Промежуточные данные:

- а)  $K$  – целочисленная переменная, используемая как счетчик цикла;

- б)  $R$  – вещественная переменная.

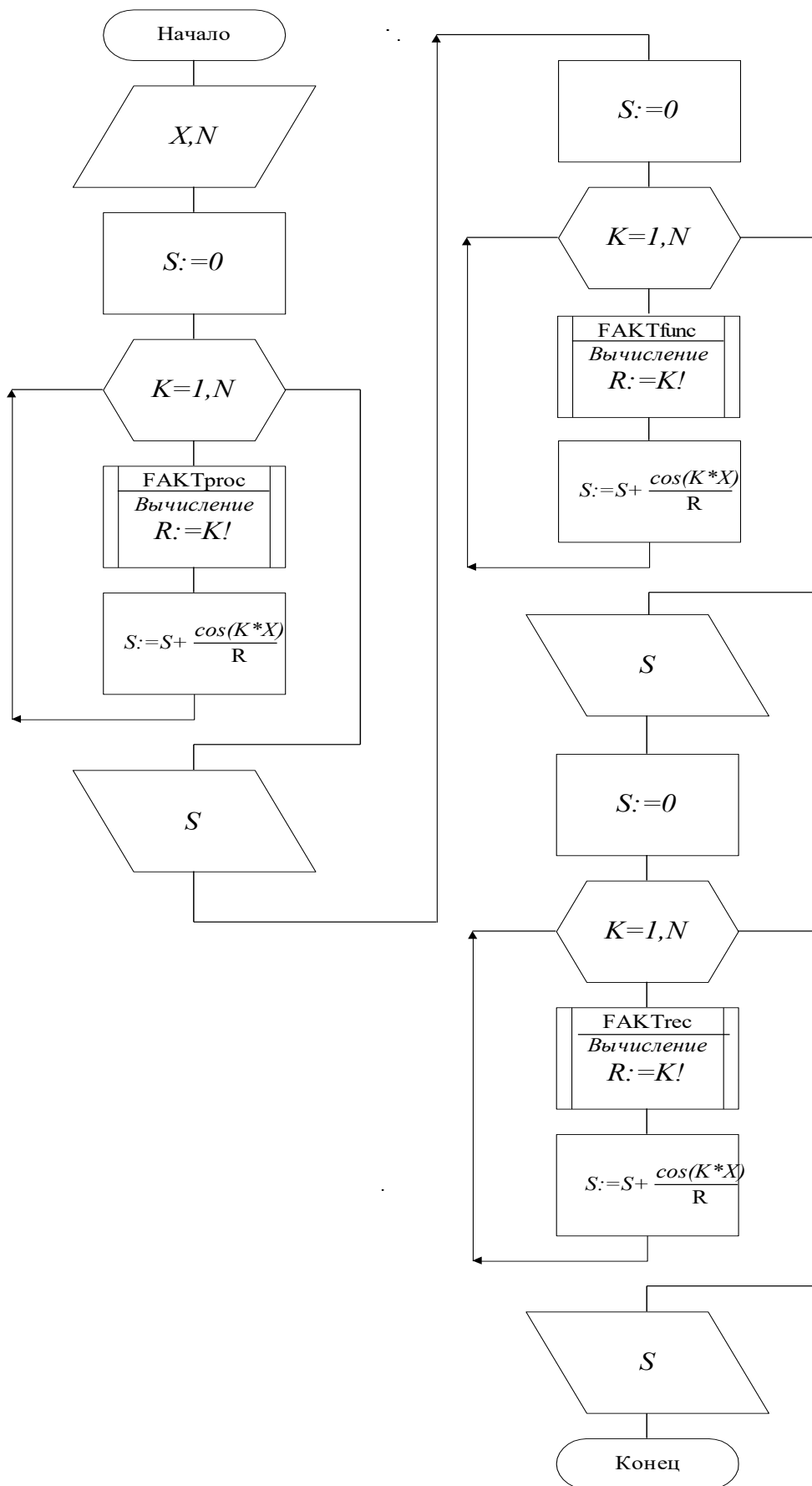


Рис. 7.1. Схема алгоритма основной программы

Процедура ФАКТпрос (рис. 7.2). Данная процедура должна вычислить  $R=K!$  .

- Входные данные:  $K$  – целочисленная переменная.
- Выходные данные:  $R$  – вещественная переменная, являющаяся значением  $K!$ .
- Промежуточные данные:  $I$  – целочисленная переменная, используемая как счетчик цикла.

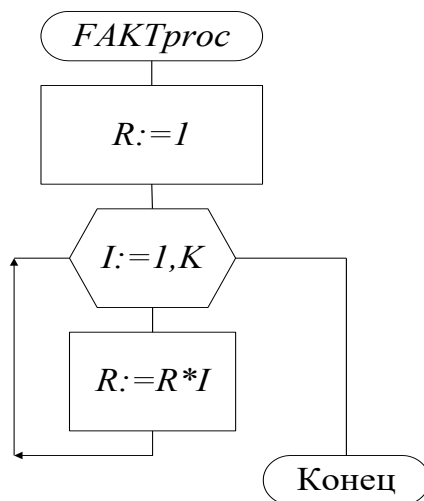


Рис. 7.2. Схема алгоритма процедуры ФАКТproc

Функция ФАКТfunk (рис. 7.3). Функция ФАКТfunk вычисляет  $K!$ .

- Входные данные:  $K$  – целочисленная переменная.
- Выходные данные: ФАКТfunk – имя функции вещественного типа.
- Промежуточные данные:  $I$  – целочисленная переменная, используемая как счетчик цикла.

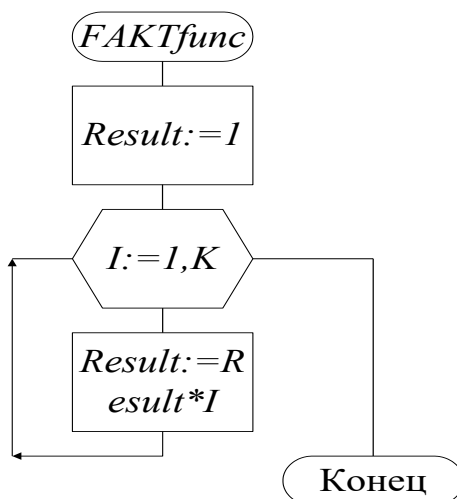


Рис. 7.3. Схема алгоритма функции ФАКТfunk

Функция ФАКТrec (рис. 7.4). Функция ФАКТrec вычисляет  $K!$  рекурсивным методом.

- Входные данные:  $K$  – целочисленная переменная.
- Выходные данные: ФАКТrec – имя функции вещественного типа.

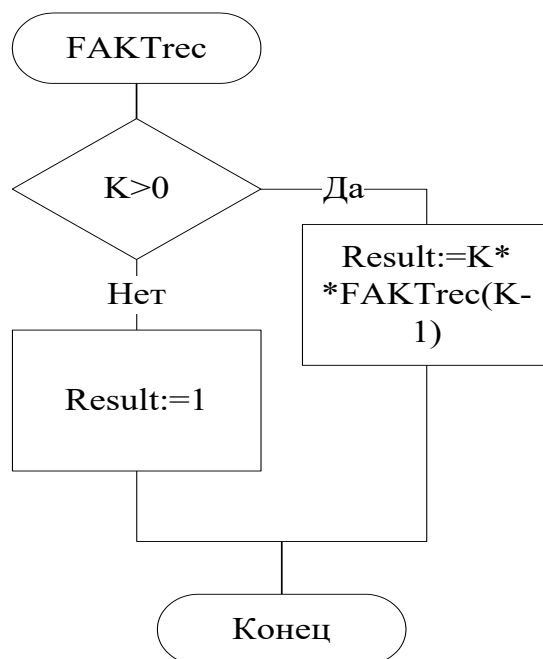


Рис. 7.4. Схема алгоритма функции ФАКТrec

2. Разработка формы (табл. 7.1, рис. 7.5).

Таблица 7.1

#### Используемые компоненты

Имя компонента	Страница палитры компонент	Настраиваемое свойство	Значение
1. Form1	–	Caption	
2. StaticText1	Additional	Caption	Введите N
3. StaticText2	Additional	Caption	Результат S
4. Edit1	Standard	Text	
5. Edit2	Standard	Text	
6. Edit3	Standard	Text	
7. Edit4	Standard	Text	
8. Edit5	Standard	Text	
9. Button1	Standard	Caption	Вычисление функции

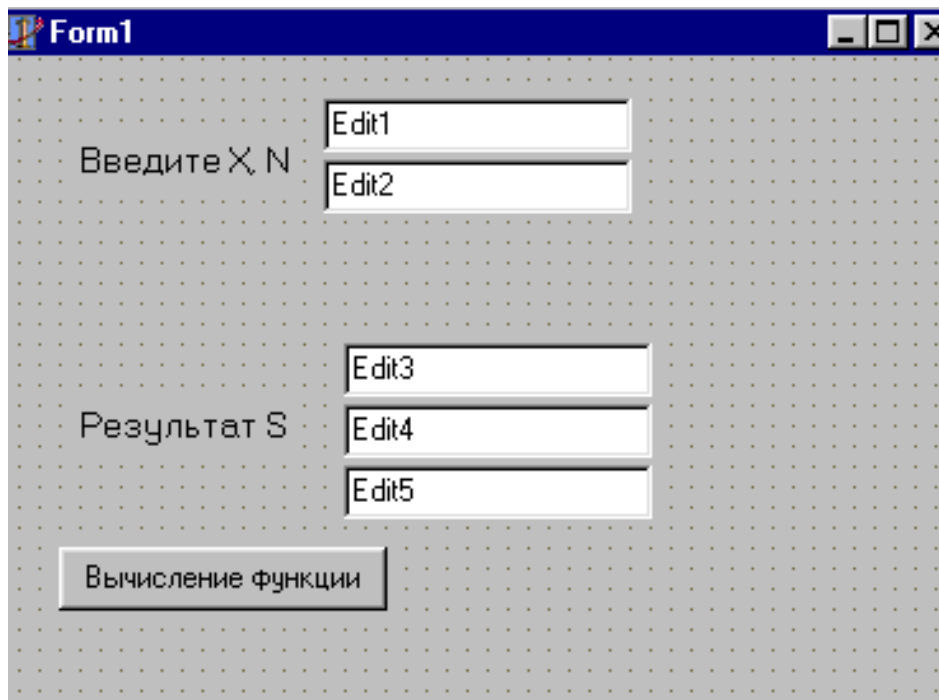


Рис. 7.5. Изображение формы

### 3. Пример программы:

```

program Project1;
uses
  Forms,
  Unit1 in 'Unit1.pas' {Form1};
{$R *.res}
begin
  Application.Initialize;
  Application.CreateForm(TForm1, Form1);
  Application.Run;
end.
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls;
type
  TForm1 = class(TForm)
    Edit1: TEdit;           // поле для ввода X
    Edit2: TEdit;           // поле для ввода N
    Edit3: TEdit;           // поле для вывода S (процедура)
    Edit4: TEdit;           // поле для вывода S (функция)
    Edit5: TEdit;           // поле для вывода S (рекурсивная
                          // функция)
    StaticText1: TStaticText; //
  end;

```

```

    StaticText2: TStaticText; // комментарии к полям
    Button1: TButton; // кнопка выполнить
    procedure Button1Click(Sender: TObject); // процедура,
    //вызываемая при щелчке левой кнопкой мыши на Button1;
    end;
var
    Form1: TForm1;
implementation
{$R *.dfm}
procedure TForm1.Button1Click(Sender: TObject);
var X,S,R:extended;
    N,K:integer;
    procedure ФАКТproc(K:integer;var R:extended); // вычисление с
    // помощью процедуры
    var i:integer;
    begin
        R:=1;
        for i:=1 to K do
            R:=R*i;
        end;
    function ФАКТfunc(K:integer):extended; // вычисление с помощью
    // функции
    var i:integer;
    begin
        Result:=1;
        for i:=1 to K do
            Result:=Result*i;
        end;
    function ФАКТrec(K:integer):extended; // вычисление с помощью
    // рекурсии
    begin
        if k>0 then
            Result:=k*ФАКТrec(k-1)
        else Result:=1;
    end;
begin
    X:=StrToFloat(Edit1.Text);
    N:=StrToInt(Edit2.Text);
    S:=0;
    for K:=1 to N do
        begin
            ФАКТproc(K,R);
            S:=S+cos(K*X)/R;
        end;
    Edit3.Text:=FloatToStr(S);

```

```

S:=0;
  for K:=1 to N do
  begin
    R:=ФАКТfunc(K);
    S:=S+cos(K*X)/R;
  end;
Edit4.Text:=FloatToStr(S);
S:=0;
for K:=1 to N do
  begin
    R:=ФАКТrec(K);
    S:=S+cos(K*X)/R;
  end;
Edit5.Text:=FloatToStr(S);
end;
end.

```

### Задание к лабораторной работе

1. Внимательно рассмотреть пример приведенный выше!!!
  2. Составить программу вычисления функции суммы S. При составлении программы необходимо использовать процедуры или функции, основанные на нерекурсивном и рекурсивном методах вычислений.
- Варианты заданий приведены ниже (табл. 7.2).

Таблица 7.2

Варианты заданий

№ варианта	Сумма S	Значение конечного n
1	$S = 1 + \frac{\ln 3}{1!} x + \frac{\ln^2 3}{2!} x^2 + \dots + \frac{\ln^n 3}{n!} x^n$	10
2	$S = \cos x + \frac{\cos 2x}{2} + \dots + \frac{\cos nx}{n}$	40
3	$S = x - \frac{x^3}{3!} + \dots + (-1)^n \frac{x^{2n+1}}{(2n+1)!}$	10
4	$S = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots + \frac{x^n}{n!}$	15
5	$S = 1 + \frac{\cos \pi/4}{1!} x + \dots + \frac{\cos n\pi/4}{n!} x^n$	25



№ варианта	Сумма S	Значение конечного n
6	$S = 1 - \frac{x^2}{2!} + \dots + (-1)^n \frac{x^{2n}}{(2n)!}$	10
7	$S = x + \frac{x^5}{5} + \dots + \frac{x^{4n+1}}{4n+1}$	40
8	$S = 1 + \frac{\cos x}{1!} + \dots + \frac{\cos nx}{n!}$	20
9	$S = 1 + 3x^2 + \dots + \frac{2n+1}{n!} x^{2n}$	10
10	$S = \frac{x-1}{x+1} + \frac{1}{3} \left( \frac{x-1}{x+1} \right)^3 + \dots$ $\dots + \frac{1}{2n+1} \left( \frac{x-1}{x+1} \right)^{2n+1}$	10
11	$S = -\cos x + \frac{\cos 2x}{2^2} + \dots + (-1)^n \frac{\cos nx}{n^2}$	20
12	$S = 1 + \frac{x^2}{2!} + \dots + \frac{x^{2n}}{(2n)!}$	10

### Контрольные вопросы к главе

1. Для чего предназначаются подпрограммы?
2. Что включает в себя заголовок процедуры?
3. Что включает в себя заголовок функции?
4. Какую структуру имеет процедура?
5. Какую структуру имеет функция?
6. Чем отличается процедура от функции?
7. Какая существует взаимосвязь между формальными и фактическими параметрами?
8. Какие существуют разновидности параметров подпрограмм?
9. Чем отличаются параметры-переменные от параметров-констант?

10. Чем отличаются параметры-значения от параметров-констант?
11. С какой целью используются открытые массивы?
12. В чем суть рекурсивного метода организации вычислений?
13. Какие достоинства и недостатки рекурсивного метода?
14. Каким образом реализуется косвенная рекурсия?