**Министерство науки и высшего образования Российской Федерации**

федеральное государственное автономное образовательное учреждение

высшего образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Инженерная школа природных ресурсов

Направление подготовки Химическая технология

Отделение химической инженерии

**Составление программ с использованием   
процедур и функций**

**Лабораторная работа по дисциплине «Углубленный курс информатики»**

Выполнил студент гр. 2Д91 Г.Н. Омельченко

(Подпись)

20 апреля 2020 г.

Отчет принят:

Преподаватель

доцент ОХИ ИШПР, к.т.н. В.А. Чузлов

(Подпись)

\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

Томск 2020 г.

**Цель работы:** рассмотреть и составить программы с использованием процедур и функций.

**Теоретическая часть**

**Процедуры и функции в языке Pascal** являются подпрограммами.

Описываются подпрограммы между разделом описания переменных и началом тела основной программы.

Процедура – это независимая совокупность операторов, выполняющих заданную последовательность действий, и определенных отдельно от основной части программы. Она начинается со служебного слова **Procedure**, после которого идет имя и заголовок процедуры. Остальная часть аналогична основной программе.

Функции могут быть задействованы в основной программе, лишь в каком либо выражении. Также для обращения к функции необходимо знать тип данных результата, возвращаемого ей.

Описание функций начинается со служебного слова **function**, вслед за которым идет ее заголовок, а дальше те же блоки, что и у процедуры.

Между функциями и процедурами есть существенное отличие. Значение, полученное в результате выполнения кода функции, жестко соотносится с ее именем путем присвоения этому имени конкретного значения. Тип, который может принять вычисляемое значение, указывается в заголовке функции (тип результата). И в теле основной программы функция вызывается только в том случае, если ее имя фигурирует в каком-либо выражении. В то время как процедура вызывается отдельно.

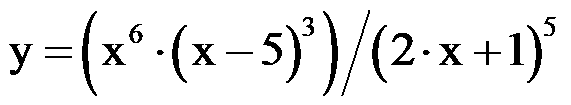
Параметры не являются обязательным компонентом, однако их использование повышает гибкость процедуры или функции, т.к. они перестают быть привязанными к определенным данным.

Тело подпрограммы, как и любого составного оператора на языке Паскаль, ограничивается скобками **begin** и **end**.

**Практическая часть**

**Задание 1**

Составить программу для вычисления данного выражения. Возведение в степень оформить в виде подпрограммы-функции, вычисление ***y*** в виде подпрограммы-процедуры: x=10;



**Программная реализация**

**program** one;

**var**

x, y: real;

///////////////////

**function** pow(x, n:real):real;

**begin**

result := exp(ln(x)\*n)

**end**;

//////////////////

**procedure** f(x:real; **var** y:real);

**begin**

y := ((pow(x,6)\*pow((x-5),3))/pow((2\*x+1),5));

**end**;

//////////////////

**begin**

x := 10;

f(x,y);

writeln(y);

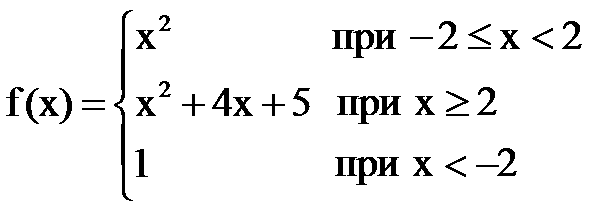
**end**.

**Ответ**

30.6064908776742

**Задание 2**

Описать вычисление ***f(x)*** по формуле:



* используя подпрограмму-функцию;
* используя подпрограмму-процедуру;

***x*** изменяется в интервале от -3 до 3 с шагом 1.

**Программная реализация**

**Program** two;

**var** y,x: real;

///////////////////

**function** f(x:real):real;

**begin**

**if** (x<2.0) **and** (x>=-2.0) **then**

**begin**

result:=x\*x;

**end**

**else**

**if** x>=2.0 **then**

**begin**

result:=x\*x+4\*x+5;

**end**

**else**

**begin**

result:=1;

**end**;

**end**;

/////////////////////

**procedure** pr(x:real; **var** y:real);

**begin**

**if** (x<2.0) **and** (x>=-2.0) **then**

**begin**

y:=x\*x;

**end**

**else**

**if** x>=2.0 **then**

**begin**

y:=x\*x+4\*x+5;

**end**

**else**

**begin**

y:=1;

**end**;

**end**;

////////////////////

**begin**

x:=-3;

writeln('Функция');

writeln('x',' ','y');

**repeat**

y:=f(x);

writeln(x,' ',y);

x:=x+1;

**until** x>3;

x:=-3;

writeln('Процедура');

writeln('x',' ','y');

**repeat**

pr(x,y);

writeln(x,' ',y);

x:=x+1;

**until** x>3;

**end**.

**Ответ**

Функция

x y

-3 1

-2 4

-1 1

0 0

1 1

2 17

3 26

Процедура

x y

-3 1

-2 4

-1 1

0 0

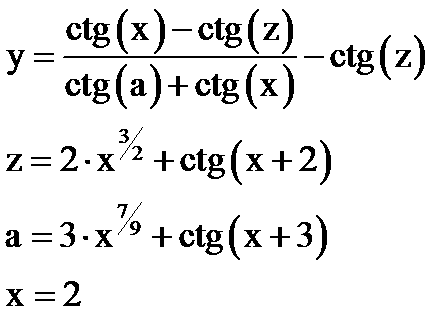
1 1

2 17

3 26

**Задание 3**

Составить программу для вычисления данного выражения. Вычисление нестандартных функций оформить в виде подпрограммы-функции, вычисление ***y*** оформить в виде процедуры (*z и a нужно объявить как локальные переменные процедуры*).



**Программная реализация**

**program** three;

**var**

x, y, z, a: real;

////////////////////////

**function** pow(x, n:real):real;

**begin**

result := exp(ln(x)\*n)

**end**;

///////////////////////

**function** ct(x: real): real;

**begin**

result := cos(x)/sin(x);

**end**;

///////////////////////

**procedure** f(z: real; a:real; **var** y:real);

**begin**

y := (ct(x)-ct(z))/(ct(a)+ct(x))-ct(z)

**end**;

///////////////////////

**begin**

x := 2;

z := 2\*pow(x,3/2)+ct(x+2);

a := 3\*pow(x,7/9)+ct(x+3);

f(z, a, y);

writeln(y);

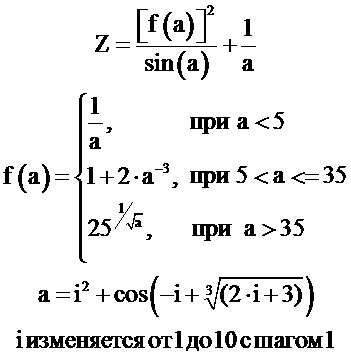
**end**.

**Ответ**

3.59902982359085

**Задание 4**

Составить программу для вычисления ***Z***. Вычисление ***f(a)*** и возведение в степень оформить в виде подпрограмм-функций, вычисление ***Z*** оформить в виде процедуры.



**Программная реализация**

**Program** four;

**var** a, i, Z: real;

//////////////////

**function** pow(x, n:real):real;

**begin**

result := exp(ln(x)\*n)

**end**;

/////////////////

**function** f(a: real): real;

**begin**

**if** (a<5.0) **then**

result := 1/a;

**if** (a>5.0) **and** (a<=35) **then**

result := 1+2\*pow(a, -3);

**if** (a>=35.0) **then**

result := pow(25,1/pow(a,1/2));

**end**;

/////////////////

**procedure** p(a: real; **var** Z: real);

**begin**

Z := (pow(f(a),2)/Sin(a))+1/a;

**end**;

/////////////////

**begin**

i := 1;

**repeat**

a := pow(i,2) + Cos(-i+pow(2\*i+3,1/3));

p(a,Z);

writeln(Z);

i := i+1;

**until** i>10;

**end**.

**Ответ**

0.897907473242297

0.158421414556934

-5.47282090141455

-11.2942909140051

-1.13702257009735

-5.95610096483213

-2.49539850253986

2.27912596686888

6.52941399630068

17.0671926834122

**Выводы**

В ходе работы успешно были составлены программы с помощью функций и процедур.