**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 А.А. Циттель

(Подпись)

\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_ 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**

**Описание и вызов процедур и функций**

Структура описания процедур и функций до некоторой степени похожа на структуру Паскаль-программы: у них также имеются заголовок, раздел описаний и исполняемая часть. Раздел описаний содержит те же подразделы, что и раздел описаний программы: описания констант, типов, меток, процедур, функций, перменных. Исполняемая часть содержит собственно операторы процедур.

Формат описания процедуры имеет вид:

procedure *имя процедуры* (*формальные параметры*);

*раздел описаний процедуры*

begin

*исполняемая часть процедуры*

end;

Формат описания функции:

function *имя функции* (*формальные параметры*):*тип результата*;

*раздел описаний функции*

begin

*исполняемая часть функции*

end;

Формальные параметры в заголовке процедур и функций записываются в виде:

var *имя праметра*: *имя типа*

и отделяются друг от друга точкой с запятой. Ключевое слово var может отсутствовать (об этом далее). Если параметры однотипны, то их имена можно перечислять через запятую, указывая общее для них имя типа. При описании параметров можно использовать только стандартные имена типов, либо имена типов, определенные с помощью команды type.Список формальных параметров может отсутствовать.

Вызов процедуры производится оператором, имеющим следующий формат:

*имя процедуры*(*список фактических параметров*);

*Список фактических параметров* - это их перечисление через запятую. При вызове фактические параметры как бы подставляются вместо формальных, стоящих на тех же местах в заголовке. Таким образом происходит передача входных параметров, затем выполняются операторы исполняемой части процедуры, после чего происходит возврат в вызывающий блок. Передача выходных параметров происходит непосредственно во время работы исполняемой части.

Вызов функции в Турбо Паскаль может производиться аналогичным способом, кроме того имеется возможность осуществить вызов внутри какого-либо выражения. В частности имя функции может стоять в правой части оператора присваивания, в разделе условий оператора if и т.д.

Для передачи в вызывающий блок выходного значения функции в исполняемой части функции перед возвратом в вызывающий блок необходимо поместить следующую команду:

*имя функции* := *результат*;

При вызове процедур и функций необходимо соблюдать следущие правила:

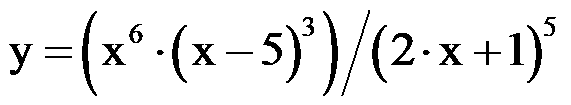
* количество фактических параметров должно совпадать с количеством формальных;
* соответствующие фактические и формальные параметры должны совпадать по порядку следования и по типу.

Заметим, что имена формальных и фактических параметров могут совпадать. Это не приводит к проблемам, так как соответствующие им переменные все равно будут различны из-за того, что хранятся в разных областях памяти. Кроме того, все формальные параметры являются временными переменными - они создаются в момент вызова подпрограммы и уничтожаются в момент выхода из нее.

**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** lab7;

**var**

x, y: real;

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

**begin**

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

**end**;

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

**begin**

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

**end**;

**begin**

x := 10;

f(x, y);

writeln(y);

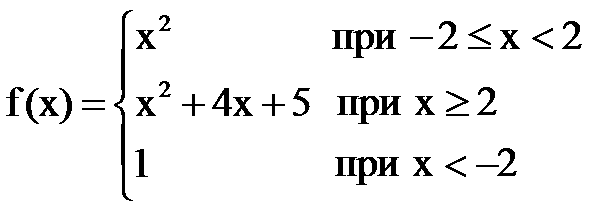
**end**.

**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** lab7;

**var**

y, x: real;

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

**begin**

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

result := x \* x;

**else**

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

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

**else**

result := 1;

**end**;

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

**begin**

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

y := x \* x;

**else**

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

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

**else**

y := 1;

**end**;

**begin**

x := -3;

writeln('Решение с помощью функции');

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

**repeat**

y := f(x);

writeln(x, ' ', y);

x := x + 1;

**until** x > 3;

x := -3;

writeln('Решение с помощью процедуры');

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

**repeat**

pr(x, y);

writeln(x, ' ', y);

x := x + 1;

**until** x > 3;

**end**.

**Ответ**

Решение с помощью функции

x y

-3 1

-2 4

-1 1

0 0

1 1

2 17

3 26

Решение с помощью процедуры

x y

-3 1

-2 4

-1 1

0 0

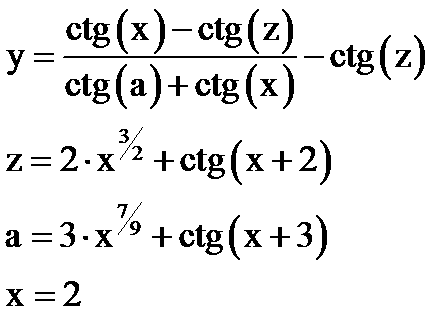
1 1

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** lab7;

**var**

x, y: real;

**function** sqr(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(x: real; **var** y: real);

**var** z,a: real;

**begin**

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

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

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

**end**;

**begin**

x := 2;

f(x,y);

writeln(y);

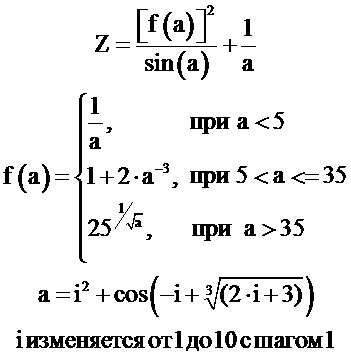
**end**.

**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** lab7;

**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 / sqr(a);

**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) + сos(-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 были изучены и составлены программы с помощью функций и процедур.