**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 Е.А. Резинкина

(Подпись)

28 марта 2020 г.

Отчет принят:

Преподаватель

доцент ОХИ ИШПР, к.т.н. В.А. Чузлов

(Подпись)

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

Томск 2020 г.

**Цель работы:** изучить использование процедур и функций для вычисления на языке Pascal, а также научиться применять полученные знания на практике.

**Теоретическая часть**

**Процедуры и функции в языке Pascal** являются подпрограммами. Подпрограмма – это именованная часть программы, представляющая собой некоторое собрание операторов, структурированных аналогично основной программе.

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

Подпрограммы в Pascal представлены в лице функций и процедур, которые могут быть встроенными и пользовательскими. Первые, как правило, изначально присутствуют в системе, а вторые создаются программистом.

Для передачи данных, после объявления имени подпрограммы, в круглых скобках необходимо указать параметры, которые в свою очередь можно разделить на передаваемые и возвращаемые. Передаваемые параметры – это переменные, которые подпрограмма получает из главной части программы, а возвращаемые – отдает (возвращает) ей.

Также различают **формальные** и **фактические** параметры. Те, что используются при вызове процедуры или функции называются фактическими, а формальные описываются в заголовке подпрограммы, и принимают значения фактических параметров. Формальные параметры должны совпадать типом и количеством с фактическими.

Переменные по отношению к программе делятся на **локальные** и **глобальные**. Те, которые объявляются в основной части, называются глобальными, а в процедуре или функции – локальными. Локальные переменные в отличие от глобальных могут использоваться лишь внутри подпрограммы, к которой они принадлежат.

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.

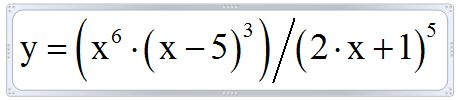
В отличие от процедур, функции могут быть задействованы в основной программе, лишь в каком либо выражении. Также для обращения к функции необходимо знать тип данных результата, возвращаемого ей.

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

**Практическая часть**

**Задание 1**

**Исходные данные:**

****

1. Составить программу для вычисления данного выражения.
2. Возведение в степень оформить в виде подпрограммы-функции
3. вычисление *y* в виде подпрограммы-процедуры:

**Программная реализация**

**program** lb07\_1;

**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);

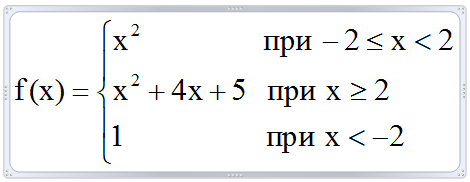
**end**.

**Ответ**

30.6064908776742

**Задание 2**

**Исходные данные**:



Описать вычисление *f(x)* по формуле

1. используя подпрограмму-функцию;
2. используя подпрограмму-процедуру;

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

**Программная реализация**

**program** lb07\_2;

**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('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

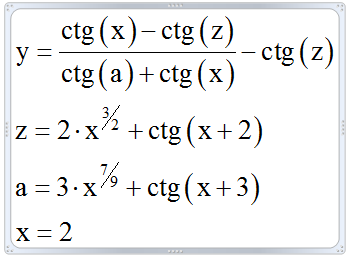
2 17

3 26

**Задание 3**

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

**Исходные данные:**

****

**Программная реализация**

**program** lb07\_3;

**var**

x, y: 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(x: real; **var** y: real);

**var** z,a: real;

**begin**

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

a := 3 \* pow(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**.

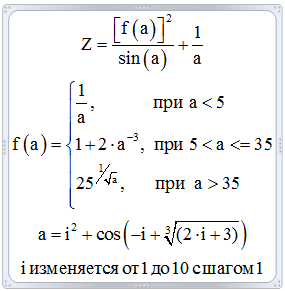
**Ответ**

3.59902982359085

**Задание 4**

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

**Исходные данные:**

****

**Программная реализация**

**program** lb07\_4;

**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) + 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 была изучена и применена на практике для вычисления теория по теме процедуры и функции