**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. Е.С. Мухина

(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**

В языке Паскаль, как и в большинстве языков программирования, предусмотрены средства, позволяющие оформлять вспомогательный алгоритм как подпрограмму. Это бывает необходимо тогда, когда какой-либо подалгоритм неоднократно повторяется в программе или имеется возможность использовать некоторые фрагменты уже разработанных ранее алгоритмов. Кроме того, подпрограммы применяются для разбиения крупных программ на отдельные смысловые части в соответствии с модульным принципом в программировании.

Для использования подалгоритма в качестве подпрограммы ему необходимо присвоить имя и описать алгоритм по правилам языка Паскаль. В дальнейшем, при необходимости вызвать его в программе, делают вызов подпрограммы упоминанием в нужном месте имени соответствующего подалгоритма со списком входных и выходных данных. Такое упоминание приводит к выполнению входящих в подпрограмму операторов, работающих с указанными данными. После выполнения подпрограммы работа продолжается с той команды, которая непосредственно следует за вызовом подпрограммы.

В языке Паскаль имеется два вида подпрограмм - **процедуры** и **функции**.

Процедуры и функции помещаются в раздел описаний программы. Для обмена информацией между процедурами и функциями и другими блоками программы существует механизм **входных** и **выходных параметров**. Входными параметрами называют величины, передающиеся из вызывающего блока в подпрограмму (исходные данные для подпрограммы), а выходными - передающиеся из подрограммы в вызывающий блок (результаты работы подпрограммы).

Одна и та же подпрограмма может вызываться неоднократно, выполняя одни и те же действия с разными наборами входных данных. Параметры, использующиеся при записи текста подпрограммы в разделе описаний, называют **формальными**, а те, что используются при ее вызове - **фактическими**.

**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** number1;

**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** number2;

**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** number3;

**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**

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** number4;

**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 успешно обучилась составлять программы с помощью функций и процедур.