**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. 2Д93 Батбаяр Цолмон

(Подпись)

26.04.2020 г.

Отчет принят:

Преподаватель

доцент ОХИ ИШПР, к.т.н. В.А. Чузлов

(Подпись)

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

Томск 2020 г.

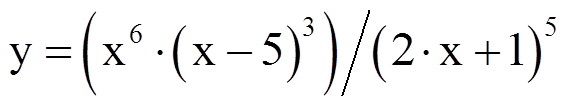
**Цель работы:** Изучение подпрограмм и работа с ними.

**Теоретическая часть**

* Подпрограмма — это фрагмент кода, который имеет свое имя и создается в случае необходимости выполнять этот код несколько раз.
* Использование подпрограмм облегчает понимание программы, а также помогает избежать ошибок при написании кода программы.
* Вся программа условно может быть разделена на две части: основную и вспомогательную. В основной части производится простейшая обработка информации, организуется обращение к разным вспомогательным модулям (подпрограммам).
* Вспомогательный алгоритм тоже может вызывать другие вспомогательные, длина такой цепочки вызовов теоретически не ограничена. Здесь и далее следующие пары слов используются как синонимы: алгоритм и программа, вспомогательный алгоритм и подпрограмма, команда и оператор, программа и модуль. Вспомогательными и основными алгоритмы являются не сами по себе, а по отношению друг к другу.
* При использовании вспомогательных алгоритмов необходимо учитывать способ передачи значений исходных данных для них и получения результата от них. Аргументы вспомогательного алгоритма — это переменные, в которых должны быть помещены исходные данные для решения соответствующей подзадачи. Результаты вспомогательного алгоритма — это также переменные, где содержаться результаты решения этих подзадач, а также результатом может быть конкретное действие, которое совершает компьютер под действием подпрограммы.
* Подпрограммы могут быть двух видов: подпрограмма без параметров и подпрограмма с параметрами. Обращение к подпрограмме может быть организовано из любого места основной программы или другой подпрограммы сколько угодно раз.
* При работе с подпрограммами важными являются понятия формальных и фактических параметров. Формальные параметры — это идентификаторы входных данных для подпрограммы. Если формальные параметры получают конкретные значения, то они называются фактическими. Формальные параметры могут получить конкретные значения только в той программе, где производится обращение к данному модулю-подпрограмме. Тип и порядок записи фактических параметров должны быть такими же, как и формальных параметров. В противном случае результат работы программы будет непредсказуемым. Из этого следует, что фактические параметры используются при обращении к подпрограмме из основной, а формальные параметры — только в самом модуле.
* Виды подпрограмм: функции и процедуры.

**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** laba7;

**const**

x = 10;

**var**

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**

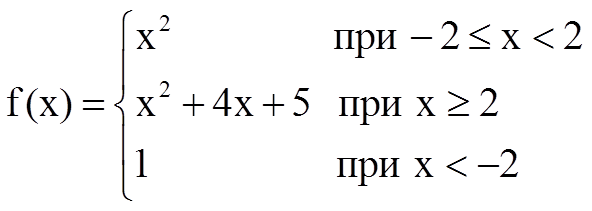
f(x, y);

write('y = ', y:6:2);

**end**.

**Ответ:** y = 30.61

**Задание 2:** 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(с использованием подпрограммы-функции):**

**program** laba7;

**var**

y, x: real;

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

**begin**

**if** x < -2 **then**

result := 1

**else**

**if** x >= 2 **then**

result := sqr(x) + 4 \* x + 5

**else**

result := sqr(x)

**end**;

**begin**

x := -3;

**repeat**

y := pow(x);

writeln(x:4:1, y:6:1);

x := x + 1;

**until** x > 3;

**end**.

**Ответ:**  -3.0 1.0

-2.0 4.0

-1.0 1.0

0.0 0.0

1.0 1.0

2.0 17.0

3.0 26.0

**Программная реализация(с использованием подпрограммы-процедуры):**

**program** laba7;

**var**

y, x: real;

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

**begin**

**if** x >= 2 **then**

y := sqr(x) + 4 \* x + 5

**else**

**if** x < -2 **then**

y := 1

**else**

y := sqr(x)

**end**;

**begin**

x := -3;

**repeat**

f(x, y);

writeln(x:4:1, y:6:1);

x := x + 1;

**until** x > 3;

**end**.

**Ответ:** -3.0 1.0

-2.0 4.0

-1.0 1.0

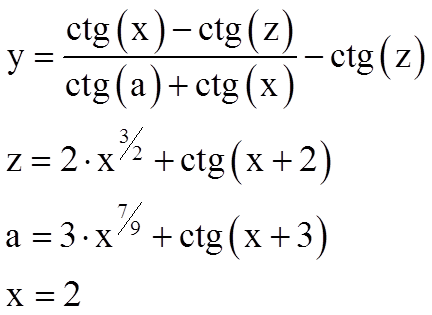
0.0 0.0

1.0 1.0

2.0 17.0

3.0 26.0

**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** laba;

**const**

x = 2;

**var**

y: real;

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

**begin**

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

**end**;

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

**begin**

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

**end**;

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

**var**

a, z: real;

**begin**

x := 2;

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

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

y := (ctg(x) - ctg(z)) / (ctg(a) + ctg(x)) - ctg(z);

**end**;

**begin**

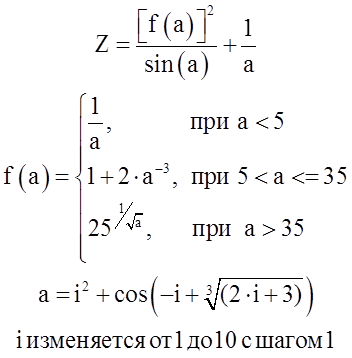
f(x, y);

write('y = ', y:6:3);

**end**.

**Ответ:** y = 3.599

**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** laba;

**var**

z, a: real;

i: integer;

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

**begin**

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

**end**;

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

**begin**

**if** a < 5 **then**

result := 1 / a

**else if** a > 35 **then**

result := pow(25, 1 / sqrt(a))

**else if** (a > 5) **and** (a <= 35) **then**

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

**end**;

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

**begin**

z := (sqr(m(a)) / sin(a)) + 1 / a;

**end**;

**begin**

i := 1;

**repeat**

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

f(a, z);

writeln(' i = ', i, ' ответ = ', z:3:3);

i := i + 1;

**until** i > 10;

**end**.

**Ответ:** i = 1 ответ = 0.898

i = 2 ответ = 0.158

i = 3 ответ = -5.473

i = 4 ответ = -11.294

i = 5 ответ = -1.137

i = 6 ответ = -5.956

i = 7 ответ = -2.495

i = 8 ответ = 2.279

i = 9 ответ = 6.529

i = 10 ответ = 17.067

**Выводы**

В ходе работы были изучены подпрограммы, способы работы с ними, а также составлены программы с их использованием.