**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 Е.В.Ветрова

(Подпись)

02.06.2020 г.

Отчет принят:

Преподаватель

доцент ОХИ ИШПР, к.т.н. В.А. Чузлов

(Подпись)

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

Томск 2020 г.

**Цель работы:** научиться писать программы с использованием подпрограмм, необходимых для улучшения структуры программ и предотвращения многих ошибок, появление которых возможно во время написания кода.

**Теоретическая часть**

**Подпрограмма** — это фрагмент кода, который имеет свое имя и создается в случае необходимости выполнять этот код несколько раз.

Использование подпрограмм обусловлено несколькими причинами, среди которых:

* Улучшение структуры программы, что облегчает её понимание.
* Отсутствие необходимости повторения фрагментов кода в программе
* Избавление от ошибок, которые можно допустить при написании кода программы.

Виды подпрограмм:

* Функции
* Процедуры

Описание функций и процедур производится в блоках function и procedure, соответственно.

Формат задания функции в PascalABC:

**Program** LB\_02**;**

**var**

// описание глобальных параметров

**function** **<**имя функции**>(**список формальных параметров**):** тип результата**;**

**var**

// описание локальных переменных

**begin**

// исполняемая часть функции

**end;**

**begin**

// операторы основной программы

**end.**

Аргументами функции могут быть переменные, выражения или другие функции.

Формат задания процедуры в PascalABC:

**Program** LB\_07**;**

**var**

// описание глобальных параметров

**procedure** **<**имя процедуры**>(**список формальных параметров**);**

**var**

// описание локальных переменных

**begin**

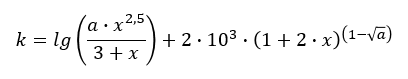
// исполняемая часть процедуры

**end;**

**begin**

// операторы основной программы

**end.**

Пример использования подпрограммы функции для решения выражения:

**program** lb\_7\_ex\_1**;**

**var**

x**,** a**,** k**:** real**;**

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

**begin**

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

**end;**

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

**begin**

result **:=** ln**(**x**)** **/** ln**(**10**)**

**end;**

**begin**

read**(**a**,** x**);**

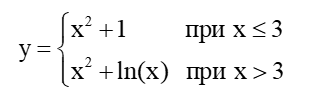
k **:=** lg**(**a **\*** pow**(**x**,** 2.5**)** **/** **(**3 **+** x**))** **+** 2e3 **\*** pow**(**1 **+** 2 **\*** x**,** 1 **-** sqrt**(**a**));**

writeln**(**k**:**8**:**4**)**

**end.**

* В данном примере подпрограммы были использованы для задания нестандартных функций xn и lg(x).
* result – встроенная переменная для хранения значения результата, возвращаемого функцией.

Пример использования подпрограммы процедуры для решения выражения

:

**program** lb\_7\_ex\_2**;**

**var**

x**,** y**:** real**;**

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

**begin**

**if** x **<=** 3 **then**

y **:=** sqr**(**x**)** **+** 1

**else**

y **:=** sqr**(**x**)** **+** ln**(**x**)**

**end;**

**begin**

x **:=** 1.5**;**

**repeat**

f**(**x**,** y**);**

writeln**(**x**:**4**:**1**,** y**:**8**:**2**);**

x **:=** x **+** 0.5

**until** x **>** 3.5

**end.**

* Процедуры не имеют встроенной переменной result.
* Перед выходными параметрами процедуры необходимо указывать служебное слово **var**, иначе значения этих параметров не будут переданы в основную программу.
* Обращение к процедуре происходит по ее имени с указанием фактически передаваемых в нее параметров (порядок перечисления параметров должен соответствовать описанию).

**Практическая часть**

**Задание 1 лабораторной работы №7**

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



При *x* =10

**Программная реализация**

**program** lb7\_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=', y);

**end**.

**Ответ**

y=30.6064908776742

**Задание 2 лабораторной работы №7**

Описать вычисление *f(x)* по формуле:



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

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

**Программная реализация**

**1)**

**program** lb7\_2\_1;

**var**

x, y: real;

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

**begin**

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

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

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

result := 1

**else** result := sqr(x);

**end**;

**begin**

x := -3;

**repeat**

y := f(x);

writeln('При х = ', x, ' результат программы = ', y);

x := x + 1;

**until** x > 3

**end**.

**Ответ**

При х = -3 результат программы = 1

При х = -2 результат программы = 4

При х = -1 результат программы = 1

При х = 0 результат программы = 0

При х = 1 результат программы = 1

При х = 2 результат программы = 17

При х = 3 результат программы = 26

**2)**

**program** lb7\_2\_2;

**var**

x, y: 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('При х = ', x, ' результат программы = ', y);

x := x + 1;

**until** x > 3

**end**.

**Ответ:**

При х = -3 результат программы = 1

При х = -2 результат программы = 4

При х = -1 результат программы = 1

При х = 0 результат программы = 0

При х = 1 результат программы = 1

При х = 2 результат программы = 17

При х = 3 результат программы = 26

**Задание 3 лабораторной работы №7**

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** lb7\_3;

**var**

y, z, a: real;

**const**

x = 2;

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

**begin**

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

**end**;

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

**begin**

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

**end**;

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

**var**

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

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

**begin**

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

**end**;

**begin**

f(x, y);

writeln('Результат выполнения программы: ', y)

**end**.

**Ответ**

Результат выполнения программы: 3.59902982359085

**Задание 4 лабораторной работы №7**

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** lb7\_4;

**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('При i = ', i, ' результат программы = ', z:3:3);

i := i + 1;

**until** i > 10;

**end**.

**Ответ**

При i = 1 результат программы = 0.898

При i = 2 результат программы = 0.158

При i = 3 результат программы = -5.473

При i = 4 результат программы = -11.294

При i = 5 результат программы = -1.137

При i = 6 результат программы = -5.956

При i = 7 результат программы = -2.495

При i = 8 результат программы = 2.279

При i = 9 результат программы = 6.529

При i = 10 результат программы = 17.067

**Выводы**

В ходе работы было изучено написание программ с использованием подпрограмм – функций и процедур, которые улучшают структуру программы и предотвращают появление ошибок в коде с большим количеством вычислений.