Министерство образования Республики Беларусь

Учреждение образования

БелорусскиЙ государственный университет

информатики и радиоэлектроники

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

|  |
| --- |
|  |
|  |

# оТЧЕТ

по лабораторной работе

на тему:

итерационные процессы

|  |  |  |
| --- | --- | --- |
| Выполнил  Студент гр. 951002 |  | В. Н. Протасеня |
| Проверил |  | Асс. Е.Е. Фадеева |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Минск, 2019

1. Теоретические сведения по теме лабораторной работы

**Классификация циклов**

В соответствии со взаимным расположением циклов в теле программы или алгоритма различают следующие циклы:

1. простые – циклы, не содержащие внутри себя других циклов;  
2. сложные – циклы, содержащие внутри себя другие циклы;  
3. вложенные (внутренние) – циклы, входящие в состав других циклов (цикл в цикле);  
4. внешние – циклы, не являющиеся составной частью других циклов, но содержащие в своем составе внутренние циклы.

***Циклический вычислительный процесс*** – процесс, в котором отдельные участки вычислений выполняются многократно.

Участок схемы, многократно повторяемый в ходе вычислений, называется ***циклом***. При повторениях обычно используются новые значения исходных данных.

**Классификация циклов**

В зависимости от *местоположения условия выполнения цикла* различают следующие циклы:

1) циклы с предусловием; 2) циклы с постусловием.

В соответствии с *видом условия выполнения* циклы делятся на следующие виды:

1) циклы с параметром; 2) итерационные циклы.

**Способы описания алгоритмa:**

1. запись на естественном языке (словесное описание)
2. изображение в виде схемы (графическое описание)
3. запись на алгоритмическом языке (составление программы)

**Свойства правильного алгоритма:**

1. Дискретность – значения величин в каждый следующий момент времени должны получаться по определенным правилам из значений величин, имевшихся в предшествующий момент времени.
2. Определенность (детерминированность) – каждое правило алгоритма должно быть однозначным. Значения величин, получаемых в какой-то момент времени, однозначно связаны со значениями величин, вычисленных ранее.
3. Результативность (конечность) – алгоритм должен приводить к решению задачи за конечное число шагов.
4. Массовость – алгоритм должен разрабатываться в общем виде так, чтобы его можно было применить для класса задач, различающихся лишь исходными данными.

**Типы алгоритмов:**

1. Линейные (четко прописанная цепочка действий, в которой нет условий)
2. Разветвляющиеся (разделение на несколько альтернативных ветвей алгоритма)
3. Циклические (многократное повторение одних и тех же операций над данными до удовлетворения нескольких условий)

**Циклы**

Циклический вычислительный процесс – это процесс, в котором отдельные участки вычислений выполняются многократно. Участок схемы, многократно повторяемый̆ в ходе вычислений, называется циклом. При повторениях обычно используются новые значения исходных данных.

**Виды циклов:**

В соответствии со взаимным расположением циклов в теле программы или алгоритма различают сле-

дующие циклы:

1. простые – циклы, не содержащие внутри себя других циклов;
2. сложные – циклы, содержащие внутри себя другие циклы;
3. вложенные (внутренние) – циклы, входящие в состав других циклов (цикл в цикле);
4. внешние – циклы, не являющиеся составной частью других циклов, но содержащие в своем составе внутренние циклы.

В зависимости от местоположения условия выполнения цикла различают следующие циклы:

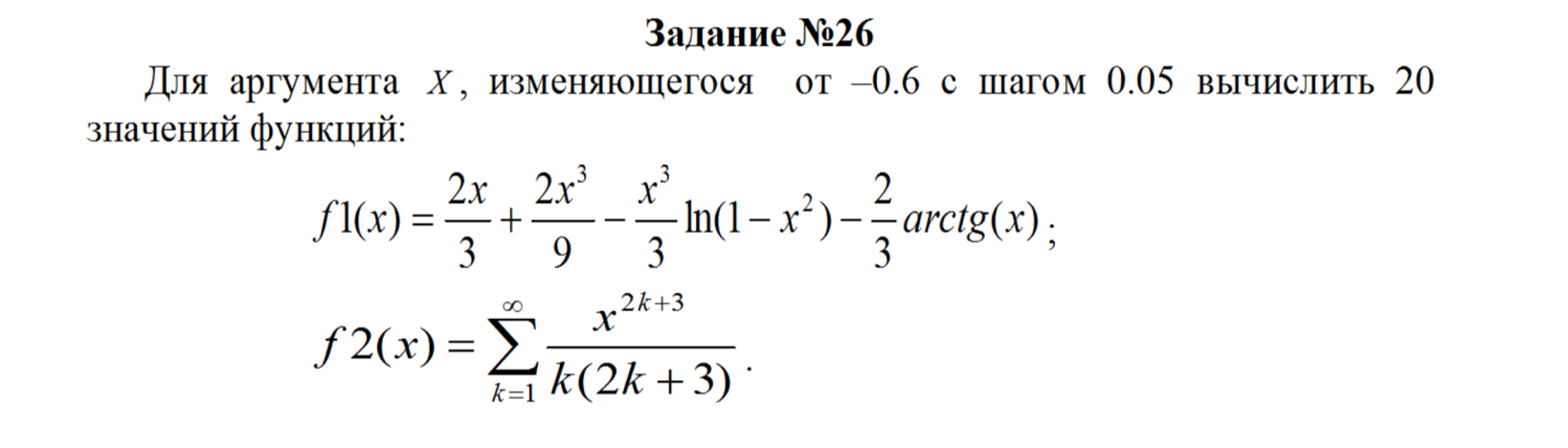
* + 1. Циклы с предусловием
    2. Циклы с постусловием

В соответствии с видом условия выполнения циклы делятся на следующие виды:

* 1. циклы с параметром;
  2. итерационные циклы.

1. Задание на лабораторную работу

Вычислить значения функций *f*1*(x)* и *f*2*(x)* для значений аргументов, указанных в пунктах заданий №1-30. Функцию *f*2*(x)* вычислить для ряда точностей . Для указанных точностей определить количество N элементов ряда, суммируемых для достижения заданной точности.



* 1. Постановка задачи

Задача: написать программу, которая выводит значения функции *y=f(x)* для начального значения аргумента *x,* равного *a,* конечного значения аргумента *x,* равного *b*, и в промежутке от *a* до *b* с шагом *h* приращения аргумента на языке Delphi в среде разработки Embarcadero Delphi.

* 1. Эскиз ожидаемого результата

Enter the initial value of x

a

Enter the final value of x

b

Enter a step value

h

Value of x: Value of y:

a f(a)

a+ h f(a+h)

a + 2\*h f(a+2\*h)

… ….

a + k\*h y is not defined'

… ….

a + n\*h f(a + n\*h)

b f(b)

1. Выполнение
   1. Разработка алгоритма

Таблица 3.1 используемые идентификаторы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя идентификатора | назначение | Тип идентификатора | Начальное значение | Закон изменения | Имя цикла, в котором происходит изменение переменной |
| x | Текущее значение аргумента | real | — | x:=x+h | 49-55, 58-63 |
| a | Начальное значение аргумента | real | Ввод с клавиатуры | Константа | — |
| ,b | Конечное значение аргумента | real | Ввод с клавиатуры | Константа | — |
| h | Шаг функции | real | Ввод с клавиатуры | Константа | — |
| y | Значение функции | real | — |  | Output x |

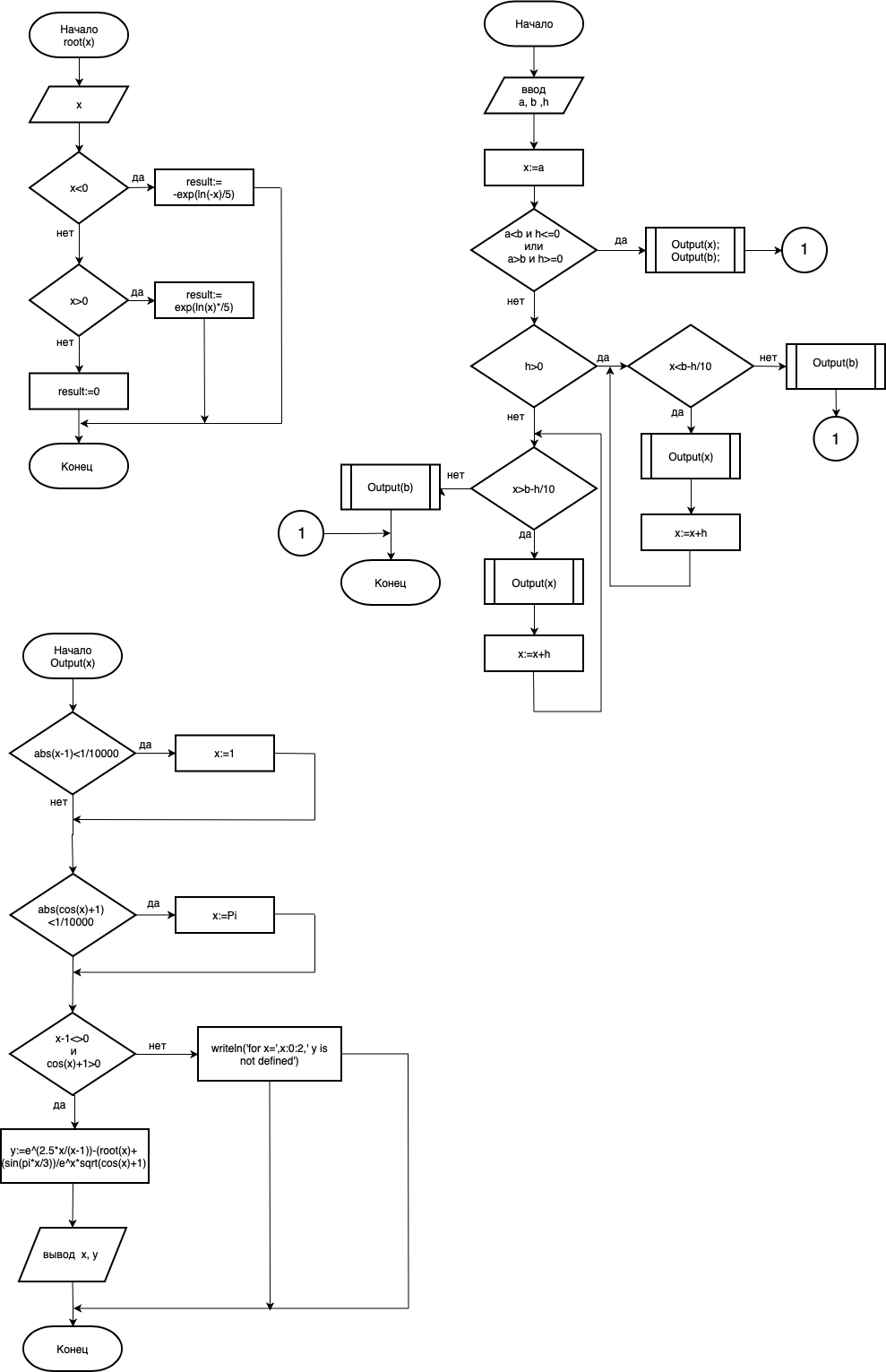


Рисунок 3.1 – Схема работы программы

Описание алгоритма:

После ввода значений пользователем в алгоритме рассмотрены три случая для различных введённых значений:

1. a>b, h<0
2. a>b, h>0
3. Иной (например, a<b<h; a=b;h=0 и др.)

Изначально переменной x присваивается значение a, после чего она изменяется по закону x:=x+h (для 1 и 2 случаев) вплоть до границы с переменной b. Если значение x находится вне области определения заданной функции, выводится сообщение ‘For x (значение аргумента) y is not defined’ и работа программы продолжается. Вне зависимости от значений программа рассчитывает значения функции для x=a и x=b и выводит их на экран, согласно эскизу. Программа состоит из двух циклов с предусловием: первый(49-55) и второй(58-63).

3.2 Текст программы и его описание

program laba1;

{$APPTYPE CONSOLE}

uses

Math,

System.SysUtils;

var x,y,a,b,h:real;

function root(x:real):real;

begin

if x<0 then result:=-exp(ln(-x)/5)

else

begin

if x>0 then result:=exp(ln(x)/5)

else result:=0;

end;

end;

procedure outpit(x:real);

begin

if abs(x-1)<1/10000 then x:=1;

if abs(cos(x)+1)<1/10000 then x:=pi;

if (x-1<>0) and (cos(x)+1>0) then

begin

y:=exp(2.5\*x/(x-1))-(root(x)+sin(pi\*x/3))/(exp(x)\*sqrt(cos(x)+1);

writeln('x= ',x:5:2,' y=',y:5:2)

end

else writeln('fox x= ',x:5:2,' y is not defined');

end;

begin

writeln ('Enter the initial value of x');

readln(a);

writeln ('Enter the final value of x');

readln(b);

writeln ('Enter a step value');

readln(h);

x:=a;

if ((a<b) and (h<=0)) or ((a>b) and (h>=0)) then

begin

Outpit(x);

Outpit(b);

end

else

if h>0 then

begin

while x<b-h/10 do

begin

Outpit(x);

x:=x+h;

end;

Outpit(b);

end

else

begin

while x>b-h/10 do

begin

Outpit(x);

x:=x+h;

end;

Outpit(b);

end;

readln;

end.

Описание программы: программа выводит значения x от a (начальное значение аргумента x) до b (конечное значение аргумента x) с шагом h приращения аргумента для заданной в условии задания функции. Программа состоит из двух циклов с предусловием: первый(49-55) и второй(58-63).

* 1. Тестирование и отладка программы

Таблица 3.2 Прохождение тестов программы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Специфика тестирования | Номер теста | Вводимые данные | Ожидаемый результат | Полученный результат |
| a < b, h > 0 | 1 | a=3.4  b=5.2  h=0.3 | x= 3.40 y= 34.52  x= 3.70 y= 30.75  x= 4.00 y= 28.03  x= 4.30 y= 25.98  x= 4.60 y= 24.39  x= 4.90 y= 23.12  x= 5.20 y= 22.09 | Тест пройден |
| a>b, h<0 | 2 | a=6  b=3  h=-0.5 | x= 6.00 y= 20.08  x= 5.50 y= 21.23  x= 5.00 y= 22.76  x= 4.50 y= 24.88  x= 4.00 y= 28.03  x= 3.50 y= 33.11  x= 3.00 y= 42.51 | Тест пройден |
| a<0, a=b | 3 | a=-3  b=-3  h=1 | x= -3.00 y= 9.02 | Тест пройден |
| х не входит в область определения(x=Pi) | 4 | a=3.14  b=5  h=2 | fox x= 3.14 y is not defined  x= 5.00 y= 22.76 | Тест пройден |
| a>0, a=b | 5 | a=3.33  b=3.33  h=0.3 | x= 3.33 y= 35.62 | Тест пройден |
| h = 0 | 6 | a=1  b=2  h=0 | fox x= 1.00 y is not defined  x= 2.00 y= 148.20 | Тест пройден |
| a=b, h=0 | 7 | a=2.5  b=2.5  h=0 | x= 2.50 y= 64.44 | Тест пройден |
| a >b, h > 0 | 8 | a=5  b=2.5  h=0.4 | x= 5.00 y= 22.76  x= 2.50 y= 64.44  x= 5.00 y= 22.76  x= 2.50 y= 64.441.81 | Тест пройден |
| a < b, h < 0 | 9 | a=1.7  b=6.8  h=-0.4 | x= 1.70 y= 432.94  x= 6.80 y= 18.74 | Тест пройден |
| х не входит в область определения(x=1) | 10 | a=1  b=3  h=0.5 | fox x= 1.00 y is not defined  x= 1.50 y= 1807.56  x= 2.00 y= 148.20  x= 2.50 y= 64.44  x= 3.00 y= 42.51 | Тест пройден |

Анализ прохождения тестов, причины возникших ошибок, внесение изменений в алгоритм и текст программы

Программа прошла тестирование, ошибок не обнаружено.

* 1. Итоговый текст программы

program labaa1;

{$APPTYPE CONSOLE}

uses

Math,

System.SysUtils;

var x,y,a,b,h:real;

function root(x:real):real;

begin

if x<0 then result:=-exp(ln(-x)/5)

else

begin

if x>0 then result:=exp(ln(x)/5)

else result:=0;

end;

end;

procedure outpit(x:real);

begin

if abs(x-1)<1/10000 then x:=1;

if abs(cos(x)+1)<1/10000 then x:=pi;

if (x-1<>0) and (cos(x)+1>0) then

begin

y:=exp(2.5\*x/(x-1))-(root(x)+sin(pi\*x/3))/exp(x)\* sqrt(cos(x)+1);

writeln('x= ',x:5:2,' y=',y:5:2)

end

else writeln('fox x= ',x:5:2,' y is not defined');

end;

begin

writeln ('Enter the initial value of x');

readln(a);

writeln ('Enter the final value of x');

readln(b);

writeln ('Enter a step value');

readln(h);

x:=a;

if ((a<b) and (h<=0)) or ((a>b) and (h>=0)) then

begin

Outpit(x);

Outpit(b);

end

else

if h>0 then

begin

while x<b-h/10 do

begin

Outpit(x);

x:=x+h;

end;

Outpit(b);

end

else

begin

while x>b-h/10 do

begin

Outpit(x);

x:=x+h;

end;

Outpit(b);

end;

readln;

end.