Лабораторная работа №3

Детерминированные вычислительные процессы с управлением по аргументу. Численное интегрирование.

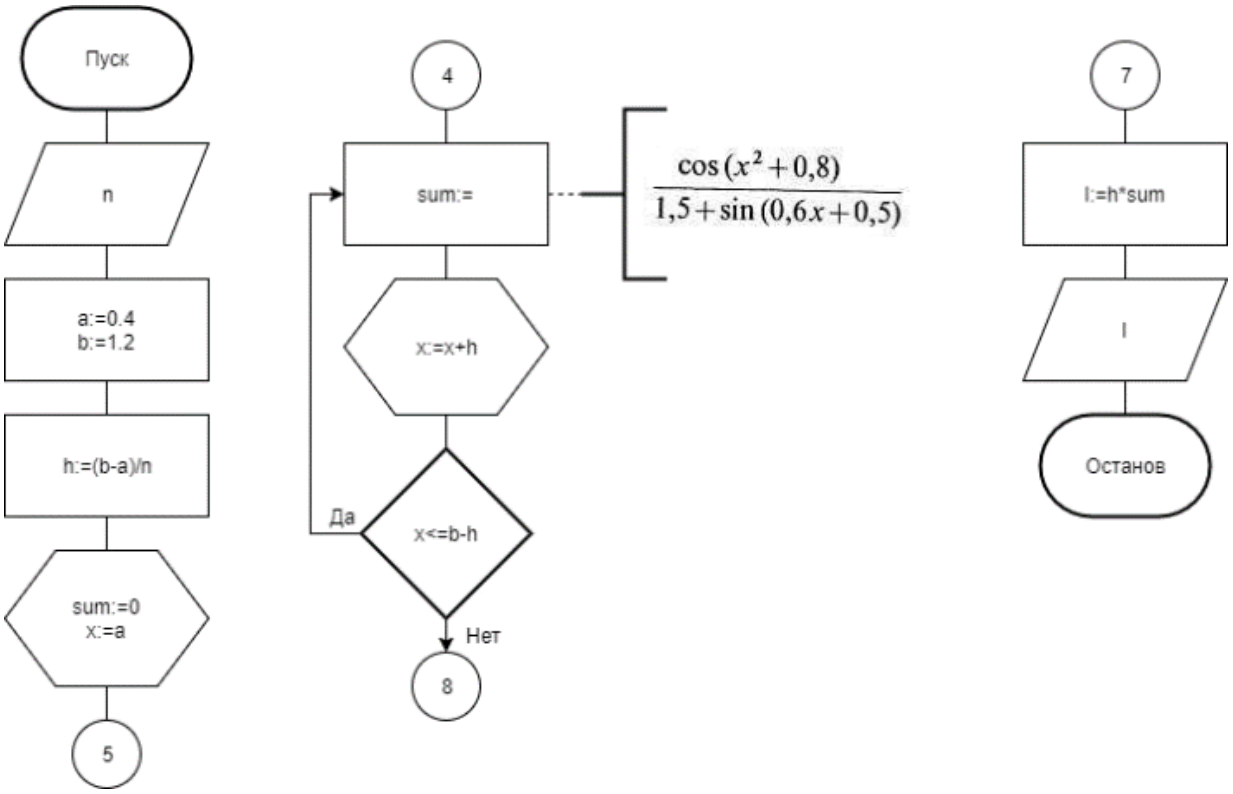
Цель: научиться реализовывать алгоритмы на детерминированные вычислительные процессы с управлением по аргументу средствами компилятора Free Pascal на примерах численного интегрирования

Оборудование: ПК, Pascal ABC

Вычислить определенный интеграл разными методами

$$\int_{0,4}^{1,2} \frac{\cos(x^2+0,8)\,dx}{1,5+\sin(0,6x+0,5)}$$

Метод левых частей прямоугольников

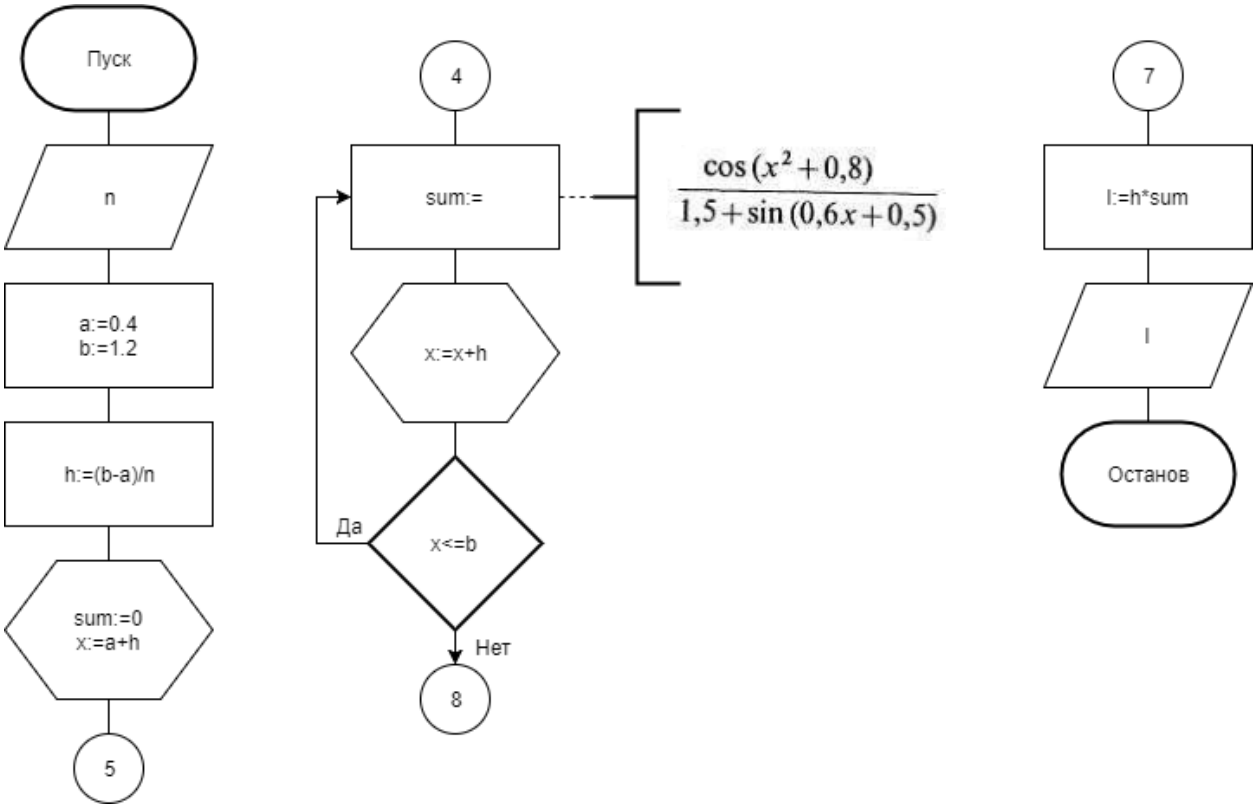


Имя	Смысл	Тип
n	переменная	integer
a,b,h	постоянные	real
x	параметр цикла	real
sum	промежуточная	real
I	результирующая	real

```
program lr31;
var a,b,h,sum,x,I:real;
   n:integer;

begin
read(n);
a:=0.4;
b:=1.2;
h:=(b-a)/n;
sum:=0;
x:=a;
while x<=b-h do
   begin
   sum:=sum+(cos(x*x+0.8)/(1.5+sin(0.6*x+0.5)));
   x:=x+h;
   end;
I:=h*sum;
write(I);
end.
```

Метод правых частей прямоугольников

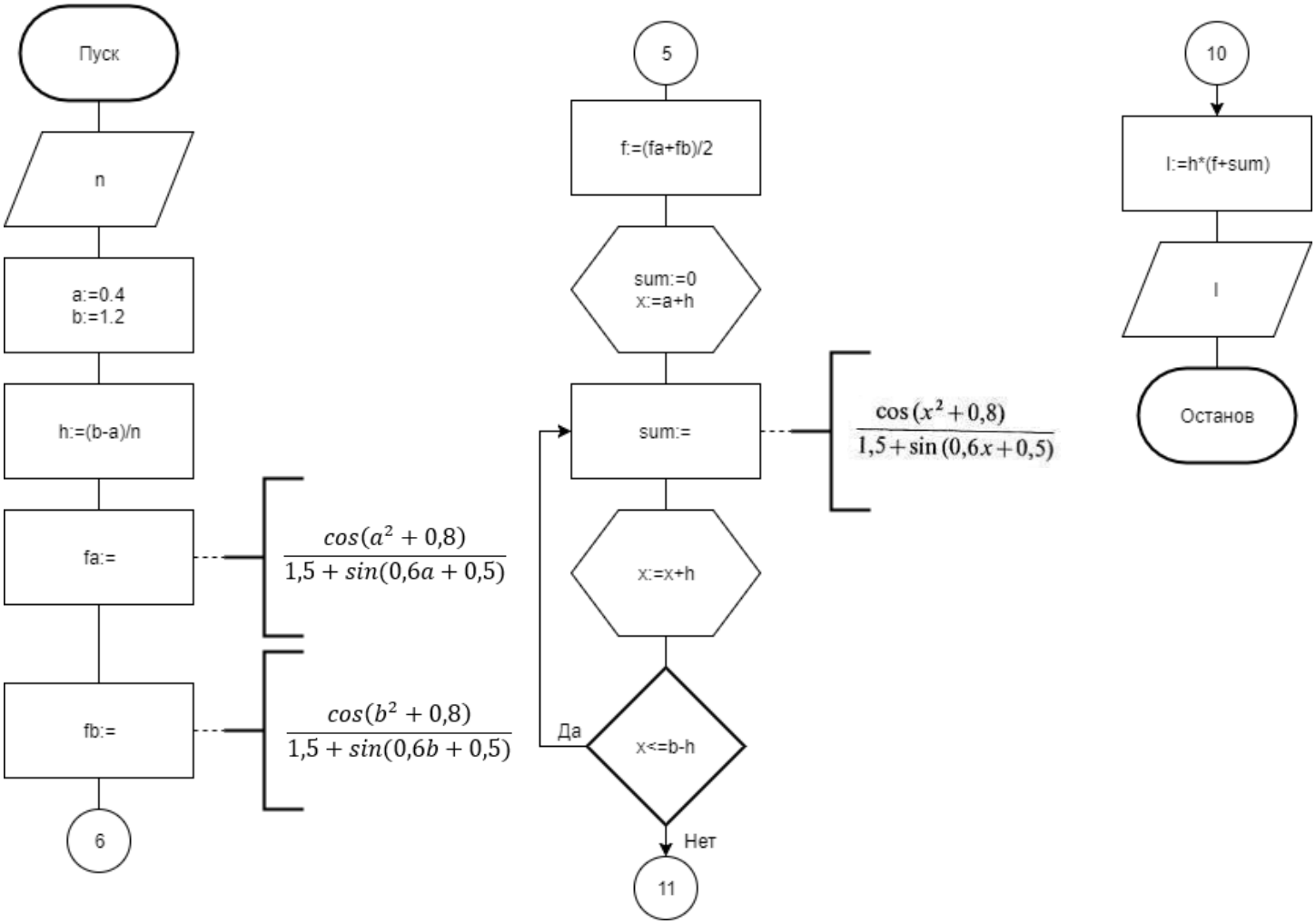


Имя	Смысл	Тип
n	переменная	integer
a,b,h	постоянные	real
x	параметр цикла	real
sum	промежуточная	real
I	результирующая	real

```
program lr32;
var a,b,h,sum,x,I:real;
   n:integer;

begin
read(n);
a:=0.4;
b:=1.2;
h:=(b-a)/n;
sum:=0;
x:=a+h;
while x<=b do
   begin
   sum:=sum+(cos(x*x+0.8)/(1.5+sin(0.6*x+0.5)));
   x:=x+h;
   end;
I:=h*sum;
write(I);
end.
```

Метод трапеций



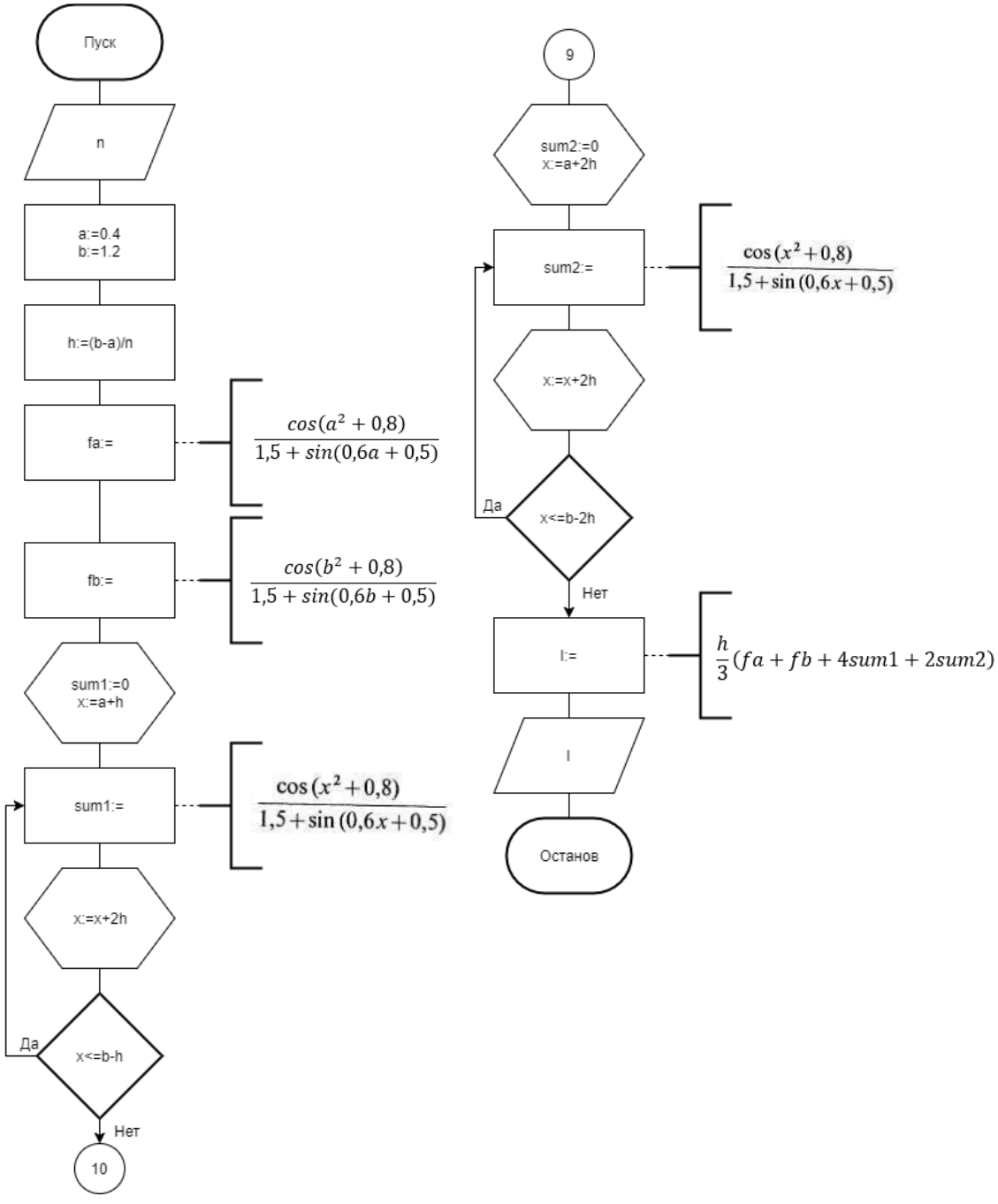
Имя	Смысл	Тип
n	переменная	integer
a,b,h	постоянные	real
x	параметр цикла	real
sum,fa,fb,f	промежуточные	real
I	результирующая	real

```
program lr33;
var a,b,h,sum,x,I,fa,fb,f:real;
   n:integer;

begin
read(n);
a:=0.4;
b:=1.2;
h:=(b-a)/n;
fa:=cos(a*a+0.8)/(1.5+sin(0.6*a+0.5));
fb:=cos(b*b+0.8)/(1.5+sin(0.6*b+0.5));
f:=(fa+fb)/2;
```

```
sum:=0;
x:=a+h;
while x<=b-h do
   begin
   sum:=sum+(cos(x*x+0.8)/(1.5+sin(0.6*x+0.5)));
   x:=x+h;
   end;
I:=h*(f+sum);
write(I);
end.
```

Метод парабол



Имя	Смысл	Тип
n	переменная	integer
a,b,h	постоянные	real
x	параметр цикла	real
sum1,sum2,fa,fb	промежуточные	real
I	результирующая	real

```
program lr34;
var a,b,h,sum1,sum2,x,I,fa,fb:real;
   n:integer;

begin
read(n);
a:=0.4;
b:=1.2;
h:=(b-a)/n;
fa:=cos(a*a+0.8)/(1.5+sin(0.6*a+0.5));
fb:=cos(b*b+0.8)/(1.5+sin(0.6*b+0.5));
sum1:=0;
x:=a+h;
while x<=b-h do
   begin
   sum1:=sum1+(cos(x*x+0.8)/(1.5+sin(0.6*x+0.5)));
   x:=x+2*h;
   end;
sum2:=0;
x:=a+2*h;
while x<=b-2*h do
   begin
   sum2:=sum2+(cos(x*x+0.8)/(1.5+sin(0.6*x+0.5)));
   x:=x+2*h;
   end;
I:=h/3*(fa+fb+4*sum1+2*sum2);
write(I);
end.
```

Результаты вычислений представлены в таблице:

N *Количество разбиений*	**h** *Шаг*	**I** *Метод левых частей прямоугольников*	**I** *Метод правых частей прямоугольников*	**I** *Метод трапеций*	**I** *Метод парабол*
10	0,08	0.0503553611337855	0.00890622896410758	0.0296307950489465	0.0298438687274363
100	0,008	0.0338988041658622	0.0298021351903264	0.0318263475573783	0.0337824800069678
1000	0,0008	0.0300491999799178	0.029634708658221	0.0298419543190694	0.0302476387534899
10000	0,00008	0.0298830421791016	0.0298415978487188	0.0298623176130168	0.0298826570558386

Вывод.

Относительно вычислений, вывод такой, что чем большее количество разбиений мы возьмем, тем точнее будет результат вычисления определенного интеграла любым из способов.

Относительно того, какой способ предпочтительнее для более точного вычисления интеграла, я не могу ничего сказать, так как каждый способ дает приблизительный результат, и погрешность в любом случает будет существовать.

И относительно решения задач, могу сказать, что в программировании без циклов не обойтись. Для более эффективной работы программы нужно использовать промежуточные переменные и грамотно выстраивать алгоритмы вычислений.