# Лабораторная работа № 3

# Детерминированные вычислительные процессы с управлением по аргументу. Численное интегрирование.

## Цель работы: научиться реализовывать алгоритмы численного интегрирования посредством детерминированных циклических вычислительных процессов с управлением по аргументу с помощью Free Pascal.

## Оборудование: PC, Lazarus

# Задача № 1

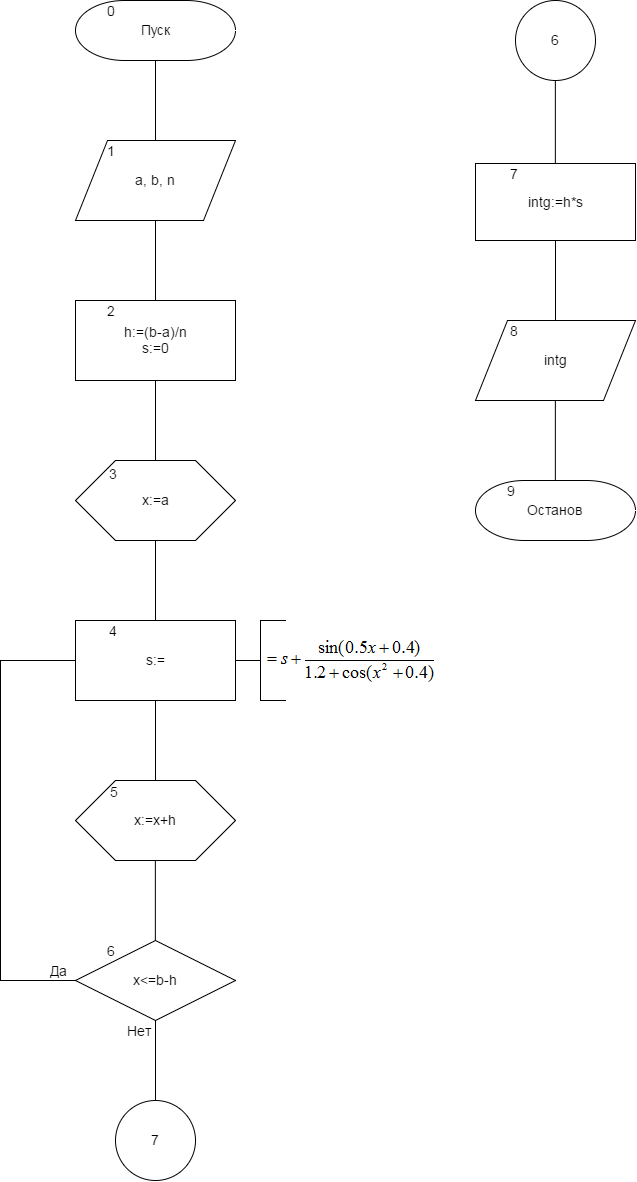
Постановка задачи: написать программу для вычисления определенного интеграла из индивидуального задания методом прямоугольника левых частей. Протестировать программу на определенном интеграле, вычисленным в ходе выполнения самостоятельной работы 3.

## Математическая модель:





## Блок-схема:



## Список идентификаторов (обозначение переменных):

*Таблица 1*

|  |  |  |
| --- | --- | --- |
| Имя | Смысл | Тип |
| a | real | Значение нижнего предела интегрирования |
| b | real | Значение верхнего предела интегрирования |
| n | real | Кол-во шагов |
| h | real | Величина шага |
| s | real | Сумма значений функции |
| x | real | Параметр цикла |
| intg | real | Значение интеграла |

Код программы:

program Zadacha1;

var h,x,a,b,s, intg, n:real;

begin

a:=0.5;

b:=1.3;

writeln('vvedite kolichestvo shagov');

readln (n);

h:= (b-a)/n;

s:=0;

x:=a;

while x<=(b-h) do

begin

s:= s+(sin(0.5\*x+0.4)/(1.2+cos(x\*x+0.4)));

x:= x+h;

end;

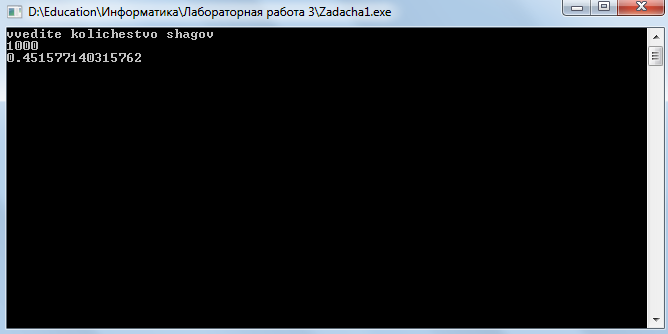
intg:= h\*s;

writeln(intg:3:10);

readln;

end.

## Результаты выполненной работы:



## Анализ результатов вычисления: Программа вычисляет определенный интеграл методом левых частей и выводит его на экран. Чем больше количество отрезков разбиения, тем точнее результат.

# Задача №2

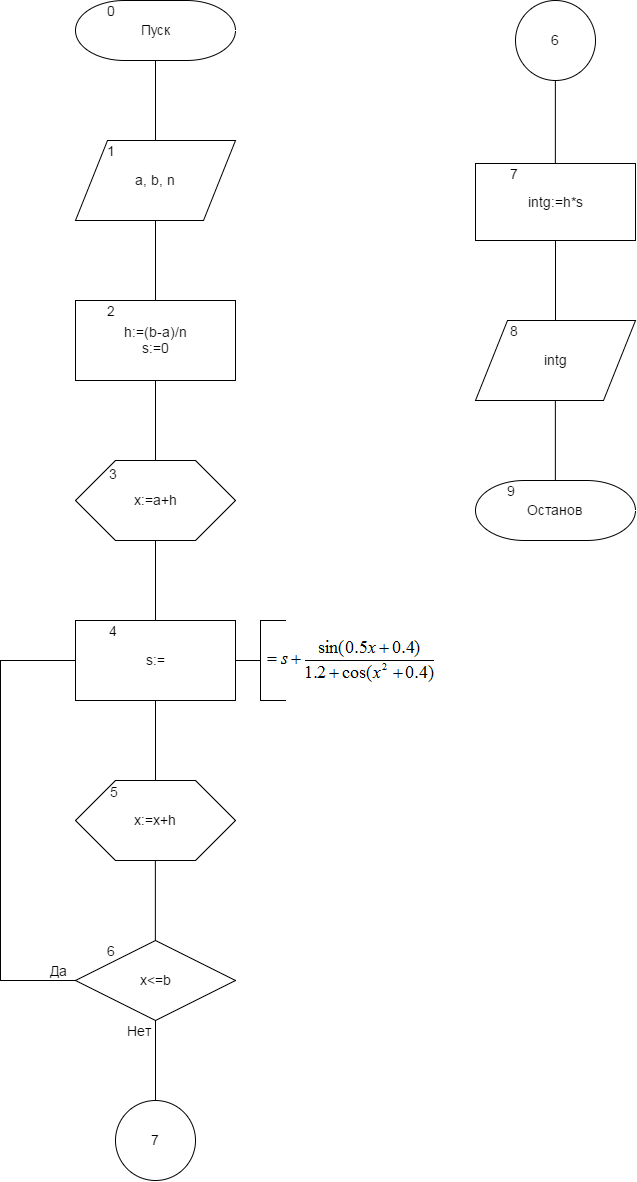
## Постановка задачи: Написать программу для вычисления определенного интеграла из индивидуального задания методом прямоугольника левых частей. Протестировать программу на определенном интеграле, вычисленным в ходе выполнения самостоятельной работы 3..

## Математическая модель:





## Блок-схема:



## Список идентификаторов (обозначение переменных):

*Таблица 2*

|  |  |  |
| --- | --- | --- |
| Имя | Смысл | Тип |
| a | real | Значение нижнего предела интегрирования |
| b | real | Значение верхнего предела интегрирования |
| n | real | Кол-во шагов |
| h | real | Величина шага |
| s | real | Сумма значений функции |
| x | real | Параметр цикла |
| intg | real | Значение интеграла |

## Код программы:

## program Zadacha2;

## var h,x,a,b,s, intg, n:real;

## begin

## a:=0.5;

## b:=1.3;

## writeln('vvedite kolichestvo shagov');

## readln (n);

## h:= (b-a)/n;

## s:=0;

## x:=a+h;

## while x<=b do

## begin

## s:= s+(sin(0.5\*x+0.4)/(1.2+cos(x\*x+0.4)));

## x:= x+h;

## end;

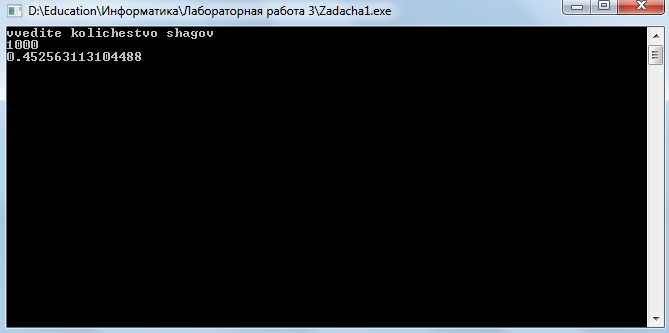
## intg:= h\*s;

## writeln(intg:3:15);

## readln;

## end.

## Результаты выполненной работы:



## Анализ результатов вычисления: Программа вычисляет определенный интеграл методом правых частей и выводит его на экран. Чем больше количество отрезков разбиения, тем точнее результат.

# Задача №3

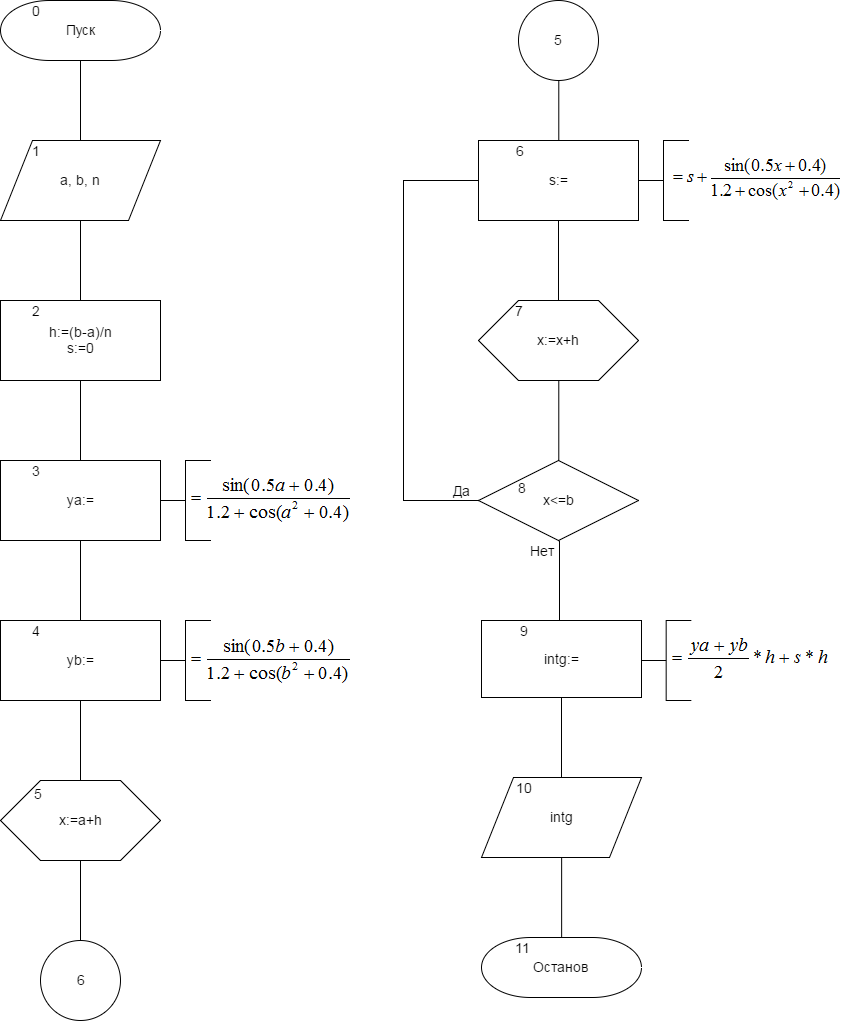
## Постановка задачи: Написать программу для вычисления определенного интеграла из индивидуального задания методом трапеций. Протестировать программу на определенном интеграле, вычисленным в ходе выполнения самостоятельной работы.

## Математическая модель:





## Блок-схема:



## Список идентификаторов (обозначение переменных):

*Таблица 3*

|  |  |  |
| --- | --- | --- |
| Имя | Смысл | Тип |
| a | real | Значение нижнего предела интегрирования |
| b | real | Значение верхнего предела интегрирования |
| n | real | Кол-во шагов |
| h | real | Величина шага |
| s | real | Сумма значений функции |
| x | real | Параметр цикла |
| intg | real | Значение интеграла |
| ya | real | Значение функции при аргументе равном a |
| yb | real | Значение функции при аргументе равном b |

## Код программы:

## program Zadacha3;

## var h,x,a,b,s,intg,n,ya,yb :real;

## begin

## a:=0.5;

## b:=1.3;

## writeln('vvedite kolichestvo shagov');

## readln (n);

## h:= (b-a)/n;

## s:=0;

## x:=a+h;

## ya:=(sin(0.5\*a+0.4)/(1.2+cos(a\*a+0.4)));

## yb:=(sin(0.5\*b+0.4)/(1.2+cos(b\*b+0.4)));

## while x<=(b-h) do

## begin

## s:= s+(sin(0.5\*x+0.4)/(1.2+cos(x\*x+0.4)));

## x:= x+h;

## end;

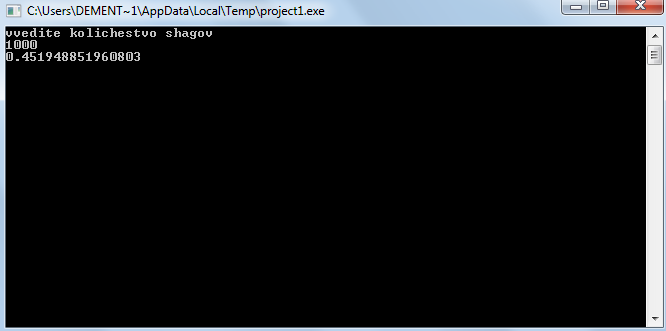
## intg:= h\*(((ya+yb)/2)+s);

## writeln(intg:3:15);

## readln;

## end.

## Результаты выполненной работы:



## Анализ результатов вычисления: Программа вычисляет определенный интеграл методом трапеций и выводит его на экран. Чем больше количество отрезков разбиения, тем точнее результат.

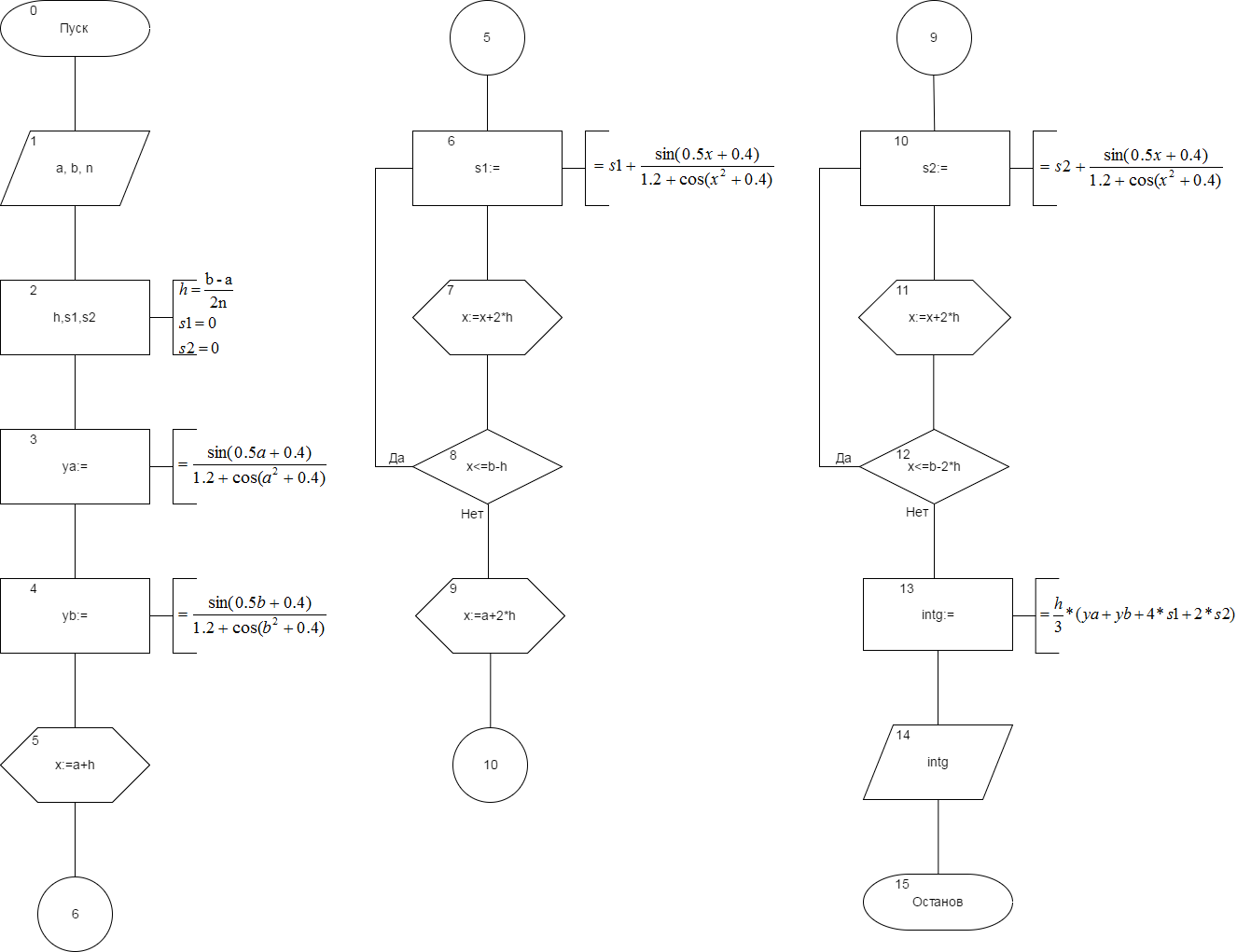
# Задача №4

## Постановка задачи: Написать программу для вычисления определенного интеграла из индивидуального задания методом парабол. Протестировать программу на определенном интеграле, вычисленным в ходе выполнения самостоятельной работы.

## Математическая модель:



## Блок-схема:



## Список идентификаторов (обозначение переменных):

*Таблица 4*

|  |  |  |
| --- | --- | --- |
| Имя | Смысл | Тип |
| a | real | Значение нижнего предела интегрирования |
| b | real | Значение верхнего предела интегрирования |
| n | real | Кол-во шагов |
| h | real | Величина шага |
| x | real | Параметр цикла |
| intg | real | Значение интеграла |
| ya | real | Значение функции при аргументе равном a |
| yb | real | Значение функции при аргументе равном b |
| s1 | real | Сумма значений функции в нечетных шагах |
| s2 | real | Сумма значений функции в четных шагах |

## Код программы:

program Zadacha4;

var h,x,a,b,s1,s2,intg,n,ya,yb :real;

begin

a:=0.5;

b:=1.3;

writeln('vvedite kolichestvo shagov');

readln (n);

h:= (b-a)/(2\*n);

s1:=0;

s2:=0;

ya:=(sin(0.5\*a+0.4)/(1.2+cos(a\*a+0.4)));

yb:=(sin(0.5\*b+0.4)/(1.2+cos(b\*b+0.4)));

x:=a+h;

while x<=(b-h) do

begin

s1:=s1+(sin(0.5\*x+0.4)/(1.2+cos(x\*x+0.4)));

x:=x+2\*h;

end;

x:=a+2\*h;

while x<=(b-2\*h) do

begin

s2:=s2+(sin(0.5\*x+0.4)/(1.2+cos(x\*x+0.4)));

x:=x+2\*h;

end;

intg:= (h/3)\*(ya+4\*s1+2\*s2+yb);

writeln(intg:3:15);

readln;

end.

## Результат выполненной работы:

## D:\YandexDisk\Скриншоты\2017-11-28_23-10-09.png

## Анализ результатов вычисления: Программа вычисляет определенный интеграл методом парабол и выводит его на экран. Чем больше количество отрезков разбиения, тем точнее результат.

## Таблица сравнений результатов

*Таблица 5*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| N  Количество разбиений | H  шаг | I  Метод левых частей прямоугольника | I  Метод правых частей прямоугольника | I  Метод трапеций | I  Метод парабол |
| 10 | 0,08 | 0.5155068634267 | 0.4912519135624 | 0.4540807490 | 0.451950951168 |
| 100 | 0,008 | 0.4482529342067 | 0.4458274392203 | 0.4423797770 | 0.442269563193 |
| 1000 | 0,0008 | 0.4525631131044 | 0.4523205636058 | 0.4519488519 | 0.451948637832 |
| 10000 | 0,00008 | 0.4519114688000 | 0.4518872138502 | 0.4518500702 | 0.451850058915 |

# Вывод.

**Вывод:** Наиболее точным способом вычисления определенного интеграла является метод Симпсона (метод парабол). Для того, чтобы повысить точность любого метода, необходимо задать большее количество отрезков разбиения.