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:

ТЕМА ЛАБОРАТОРНОЙ РАБОТЫ

|  |  |  |  |
| --- | --- | --- | --- |
| Выполнил  Студенты гр. 851002 |  | В.В. Веко  А.К. Денисевич  К.Д. Цыбулько | |
| Проверил |  | Асс. Е.Е. Фадеева | |
|  |  |  |
|  |  |  |
|  |  |  |

Минск, 2018

1. Теоретические сведения по теме лабораторной работы

Основное назначение процедурноого типа - предоставить возможность передачи имени функции или процедуры в качестве фактического параметра в другие процедуры или функции.

При объявлении процедурного типа используется заголовок процедуры (или функции) без указания ее имени, например:

Const n = 20;

Type TMas = Array [1..n] of integer;

TProc = Procedure (var A: TMAs);

TFunc = Function (x: Real): Real;

Существуют два процедурных типа: тип-процедура и тип-функция.

Если подпрограмма должна передаваться в качестве фактического параментра в другую подпрограмму, она должна удовлетворять следующим требованиям:

1. Она должна компилирваться в состоянии Far;
2. Она не должна быть стандартной процедурой или функцией;
3. Она не должна быть вложенной;
4. Она не должна быть подпрограммой типа Inline;
5. Она не должна быть подпрограммой прерывания;
6. Ее заголовок должен соответствовать типу процедурной переменной, используемой в качестве формального параметра вызывающей подпрограммы.

На физическом уровне при присваивании процедурной переменной имени подпрограммы в данную переменную заносится адрес подпрограммы.

МЕТОД ПРАВЫХ ПРЯМОУГОЛЬНИКОВ

Разобьем отрезок интегрирования на n равных частей h=(b-a)/n. Для вычисления интеграла по формуле правых прямоугольников получаем следующую формулу:



МЕТОД ЦЕНТРАЛЬНЫХ ТРЕУГОЛЬНИКОВ

Разобьем отрезок интегрирования на n равных частей h=(b-a)/n. Для вычисления интеграла по формуле правых прямоугольников получаем следующую формулу:



ВЫЧИСЛЕНИЕ ИНТЕГРАЛЛА С ЗАДАННОЙ ТОЧНОСТЬЮ

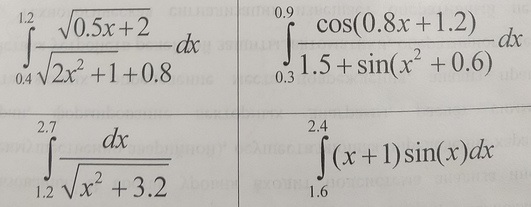
Вычисление интегралла с заданной точностью E предполагает, что для различного кол-ва n интервалов разбиения вычисляется площадь по одному из выше описанных численных методов, т.е. имеется ряд значений интеграла . Тогда в качестве резульатата расчета можно полагать , если поле (I+1) расчетной итерации выполняется следующее условие:.

В качестве начального приближения интеграла можно взять значение .

1. Задание на лабораторную работу

В соответсвии с двумя методами численных вычислений и вариантом задания разработать программу для вычисления интеграла.

Для вычисления подынтегральных функций и вычисления непосредственно интеграла использовать функции с процедурными параметрами. Интегралы вычислять с точностью: .



1. Выполнение
   1. Разработка алгоритма, используя метод Дамке

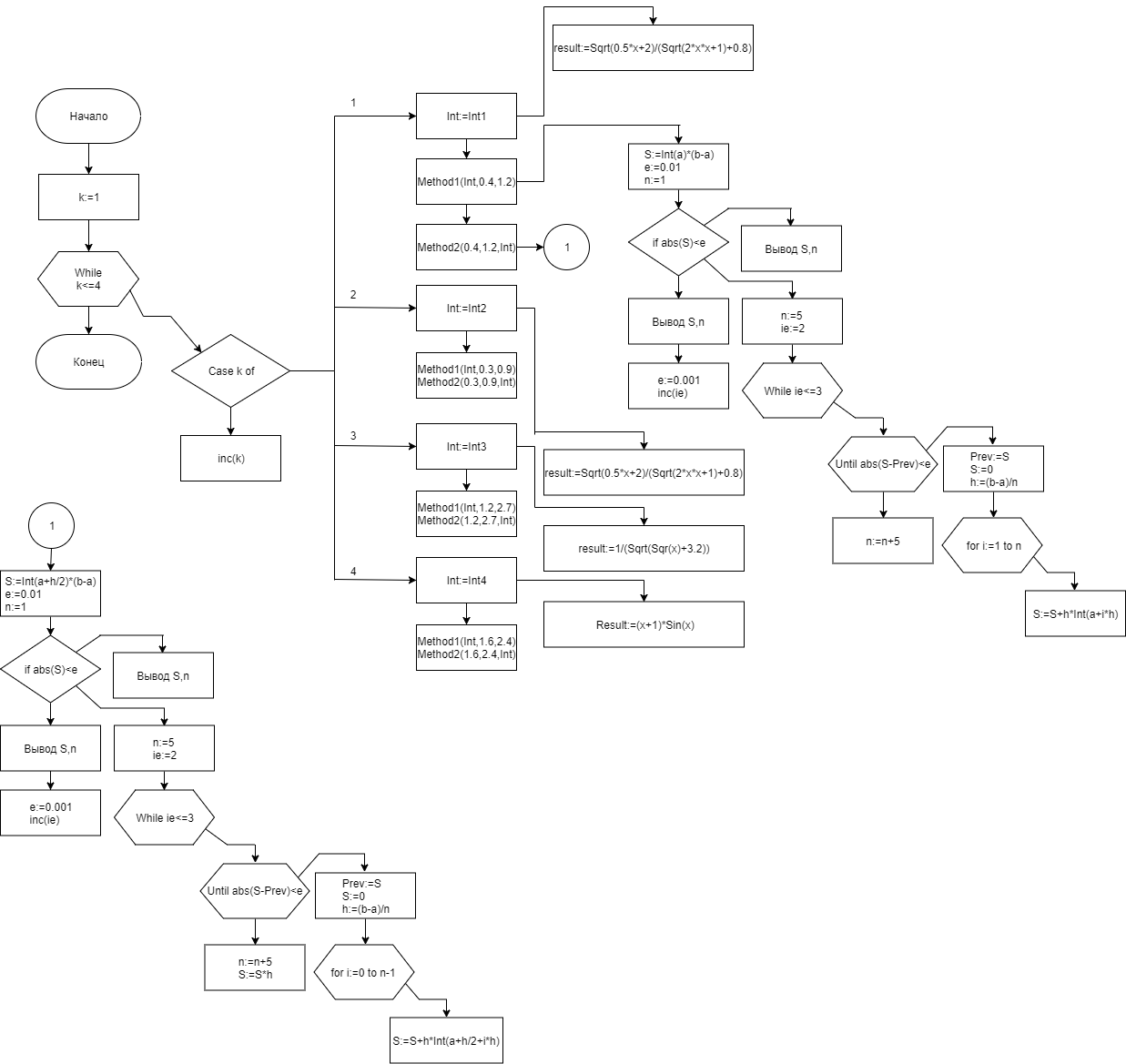


Рисунок 3.1 – Схема работы программы

3.2 Определение подпрограмм и их описание

Таблица 3.1 используемые идентификаторы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя подпрограммы | Назначение подпрограммы | Заголовок подпрограммы | Имя параметра | Назначение параметра |
| Method1 | Находит интеграл с точностью до e=10^-2 и e=10^-3 методом правых прямоугольников, выводит эти значения, выводит количество интервалов разбиения | procedure Method1(Int: TInt; a, b: real); | Int | Хранение адреса функции вычисления интеграла |
| a | Нижняя граница интегрирования |
| b | Верхняя граница интегрирования |
| S | Сумма подынтегральных функций, зависящих от Xi по i=1…n |
| h | Шаг разбиения |
| Prev | Предыдущее значение интеграла |
| e | Точность |
| n | Количество интервалов разбиение |
| i | Параметр цикла |
| ie | Модуль степени 10 е |
| Int3 | Находит значение подынтегральной функции 3 интеграла от заданного X | function Int3 (x: real): Real; | х | Значение Хi, передаваемого в подпрограмму |
| Int4 | Находит значение подынтегральной функции 4 интеграла от заданного X | function Int4 (x: Real): Real; | х | Значение Хi, передаваемого в подпрограмму |
| Method2 | Находит интеграл с точностью до e=10^-2 и e=10^-3 методом центральных прямоугольников, выводит эти значения, выводит количество интервалов разбиения | procedure Method2 (a, b: Real; Int: TInt); | a | Нижняя граница интегрирования |
| b | Верхняя граница интегрирования |
| Int | Хранение адреса функции вычисления интеграла |
| h | Шаг разбиения |
| IPrev | Предыдущее значение интеграла |
| Icur | Текущее значение интеграла |
| e | Точность |
| n | Количество интервалов разбиение |
| i | Параметр цикла |
| ie | Модуль степени 10 е |
| Head | Выводит верхнюю часть таблицы | Procedure head; | - | - |
| Row | Выводит левый столбец таблицы | procedure row(n:integer); | n | Номер интеграла |
| Int1 | Находит значение подынтегральной функции 1 интеграла от заданного X | function Int1(x:real):Real; | X | Значение Хi, передаваемого в подпрограмму |
| Int2 | Находит значение подынтегральной функции 2 интеграла от заданного X | function Int2(x:real):Real; | x | Значение Хi, передаваемого в подпрограмму |

* 1. Текст программы и его описание

program LR3\_2;

{$APPTYPE CONSOLE}

uses

SysUtils;

type

TInt = function(x: real): Real;

procedure Method1(Int: TInt; a, b: real);

var S, h, Prev, e: real;

n,i, ie: integer;

begin

S:=Int(a)\*(b-a);

e:=0.01;

n:=1;

if abs(S)<e then

begin

write(S:9:4,'|',n:3,'|');

end

else

begin

n:=5;

ie:=2;

while ie<=3 do

begin

repeat

Prev:=S;

S:=0;

h:=(b-a)/n;

for i:=1 to n do

S:=S+h\*Int(a+i\*h);

n:=n+5;

until abs(S-Prev)<e;

Write(S:9:6,'|',n-5:3,'|');

e:=0.001;

inc(ie);

end;

end;

end;

function Int3(x:real):Real;

begin

result:=1/(Sqrt(Sqr(x)+3.2));

end;

function Int4(x:Real):Real;

begin

Result:=(x+1)\*Sin(x);

end;

procedure Method2(a,b:Real;Int:TInt);

var h,Iprev,Icur,E:Real;

n,i,ie:integer;

begin

Icur:=Int(a+h/2)\*(b-a);

E:=0.01;

n:=1;

if abs(Icur)<E then

begin

write(Icur:9:4,'|',n:3,'|');

end

else

begin

n:=5;

ie:=2;

while ie<=3 do

begin

repeat

Iprev:=Icur;

Icur:=0;

h:=(b-a)/n;

for i:=0 to n-1 do

begin

Icur:=Icur+Int(a+h/2+i\*h);

end;

Icur:=h\*Icur;

n:=n+5;

until Abs(Icur-Iprev)< E;

Write(Icur:9:6,'|',n-5:3,'|');

inc(ie);

E:=0.001;

end;

end;

end;

procedure head;

begin

Writeln('|------------------------------------------------------------------|');

writeln('| | 1 method | 2 method |');

Writeln('| |---------------------------|---------------------------|');

writeln('| | e=10^-2 | e=10^-3 | e=10^-2 | e=10^-3 |');

Writeln('| |-------------|-------------|-------------|-------------|');

writeln('| | value | N | value | N | value | N | value | N |');

Writeln('|----------|-------------|-------------|-------------|-------------|');

end;

procedure row(n:integer);

begin

write('|',n,' integral|');

end;

function Int1(x:real):Real;

begin

result:=Sqrt(0.5\*x+2)/(Sqrt(2\*x\*x+1)+0.8);

end;

function Int2(x:real):Real;

begin

result:=Cos(0.8\*x+1.2)/(1.5+sin(x\*x+0.6));

end;

var k:Integer;

Int:TInt;

begin

head;

k:=1;

while k<=4 do

begin

case k of

1: begin

row(k);

Int:=Int1;

Method1(Int,0.4,1.2);

Method2(0.4,1.2,Int);

Writeln;

Writeln('|----------|-------------|-------------|-------------|-------------|');

end;

2: begin

row(k);

Int:=Int2;

Method1(Int,0.3,0.9);

Method2(0.3,0.9,Int);

writeln;

Writeln('|----------|-------------|-------------|-------------|-------------|');

end;

3: begin

row(k);

Int:=Int3;

Method1(Int,1.2,2.7);

Method2(1.2,2.7,Int);

writeln;

Writeln('|----------|-------------|-------------|-------------|-------------|');

end;

4: begin

row(k);

Int:=Int4;

Method1(Int,1.6,2.4);

Method2(1.6,2.4,Int);

Writeln;

Writeln('|------------------------------------------------------------------|');

end;

end;

k:=k+1;

end;

Readln;

end.

* 1. Тестирование и отладка программы

Таблица 3.2 Прохождение тестов программы

|  |  |  |  |
| --- | --- | --- | --- |
| Номер теста | Вводимые данные | Ожидаемый результат | Полученный результат |
| 1 |  |  | Тест пройден |

В ходе выполнения программы ошибок не обнаружено

* 1. Итоговый текст программы

См. пункт 3.3