Министерство образования Республики

Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Дисциплина: Основы алгоритмизации и программирования (ОАиП)

**ОТЧЕТ**

по лабораторной работе №10

Тема работы: Вычисление интегралов

Выполнил

студент: гр. 751003 Стубеда В.Д.

Проверил: Фадеева Е.П.

Минск 2018

Содержание

[1 Постановка задачи 3](#_Toc513335594)

[2 Описание алгоритмов 4](#_Toc513335595)

[3 Структура данных 5](#_Toc513335596)

[3.1 Структура данных основной программы 5](#_Toc513335597)

[3.1.1 Структура типов основной программы 5](#_Toc513335598)

[3.1.2 Структура констант основной программы 5](#_Toc513335599)

[3.1.3 Структура данных основной программы 5](#_Toc513335600)

[3.2 Структура данных подпрограмм 6](#_Toc513335601)

[4 Схема алгоритма решения задачи по ГОСТ 19.701-90 8](#_Toc513335602)

[5 Результаты расчетов 14](#_Toc513335603)

[Приложение А 16](#_Toc513335604)

# Постановка задачи

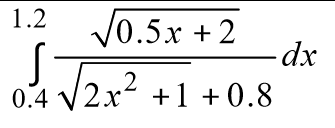
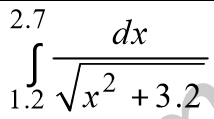
Программа вычисляет интегралы с точностью равной ε= 10-3,10-4,10-5. Проверка правильности расчетов заданных интегралов выполнина в среде mathcad.

Результаты расчетов выводится в таблицу вида:

Таблица – Пример оформления расчетов

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| N | Метод | | | | | | | | |
| Eps | Значение | n | Eps | Значение | n | Eps | Значение | n |
| 1-ый  Интеграл |  |  |  |  |  |  |  |  |  |
| 2-ой  Интеграл |  |  |  |  |  |  |  |  |  |

Интегралы для вычисления:

1. 
2. 

# Описание алгоритмов

Таблица 2 – Описание алгоритмов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| №  п.п. | Наименование  алгоритма | Назначение алгоритма | Формальные параметры | Предлагаемый тип реализации |
| 1 | Основной алгоритм | Выводит результат вычисления интегралов в виде таблиц |  |  |
| 2 | Integral\_1(x, Result) | Функция, определяющая первый интеграл | x, Result.  Result –  Возвращаемый параметр | Функция.  Параметр Result –результат функции |
| 3 | Integral\_2(x, Result) | Функция, определяющая второй интеграл | x, Result.  Result –  Возвращаемый параметр | Функция.  Параметр Result –результат функции |
| 4 | GetIntegralLeft  (Integral, min,max,eps) | Вычисляет функцию Integral методом левых прямоугольников с точностью eps | Integral, min,max,eps | Процедурa |
| 5 | GetIntegralCenter  (Integral, min,max,eps) | Вычисляет функцию Integral методом центральных прямоугольников с точностью eps | Integral, min,max,eps | Процедурa |
| 6 | FirstIntegral  (Proc,eps) | Вызывает метод для вычисления первого интеграла | Proc,eps | Процедурa |
| 7 | SecondIntegral  (Proc,eps) | Вызывает метод для вычисления второго интеграла | Proc,eps | Процедура |

# Структура данных

## Структура данных основной программы

### Структура типов основной программы

Таблица 3 – Описание типов основной программы

|  |  |  |
| --- | --- | --- |
| Наименование типа | Рекомендуемый тип | Назначение |
| TIntegral | Function (const x:Real):Real | Проседурный тип совместимый с Integral\_1 и Integral\_2 |
| TGetIntegral | Procedure (Integral:TIntegral;const min,max,eps:Real); | Проседурный тип совместимый с GetIntegralLeft и GetIntegralCenter |

### Структура констант основной программы

Таблица 4 – Описание констант основной программы

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| Min1и Max1 | Real | Пределы интегрирования 1-го интеграла |
| Min2 и Max2 | Real | Пределы интегрирования 2-го интеграла |
| eps | Array [1..3] of Real | Массив содержащий точности интегрирования |

### Структура данных основной программы

Таблица 5 – Описание данных основной программы

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| i | Byte | Счетчик цикла для изменения точности вычисления eps |

## Структура данных подпрограмм

Таблица 6 – Данные подпрограммы Integral\_1(x,Result)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра | Способ  передачи |
| x | Real | Аргумент функции | Формальный | Значение |
| Result | Real | Значение функции | Формальный | Адрес |

Таблица 7 – Данные подпрограммы Integral\_2(x,Result)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра | Способ  передачи |
| x | Real | Аргумент функции | Формальный | Значение |
| Result | Real | Значение функции | Формальный | Адрес |

Таблица 8 – Данные подпрограммы GetIntegralLeft(Integral,min,max,eps)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра | Способ  Передачи |
| Integral | TIntegral | Вычисляемый интеграл | Процедурный |  |
| min,max | Real | Нижний и верхний предел интегрирования | Формальный | Значение |
| Eps | Real | Точность вычисления | Формальный | Значение |
| h | Real | Длина интервала | Локальный |  |
| N | Integer | Количество интервалов разбиения | Локальный |  |
| Temp | Real | Предыдущее значение интеграла | Локальный |  |
| Intg | Real | Текущее значение интеграла | Локальный |  |
| i | Integer | Счетчик цикла | Локальный |  |
| R | Real | Разность между текущим и предыдущим значением интеграла | Локальный |  |

Таблица 9 – Данные подпрограммы GetIntegralCenter(Integral,min,max,eps)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра | Способ  Передачи |
| Integral | TIntegral | Вычисляемый интеграл | Процедурный |  |
| min,max | Real | Нижний и верхний предел интегрирования | Формальный | Значение |
| Eps | Real | Точность вычисления | Формальный | Значение |
| h | Real | Длина интервала | Локальный |  |
| N | Integer | Количество интервалов разбиения | Локальный |  |
| Temp | Real | Предыдущее значение интеграла | Локальный |  |
| Intg | Real | Текущее значение интеграла | Локальный |  |
| i | Integer | Счетчик цикла | Локальный |  |
| R | Real | Разность между текущим и предыдущим значением интеграла | Локальный |  |

Таблица 10 – Данные подпрограммы FirstIntegral(Proc,eps)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра | Способ  Передачи |
| Proc | TGetIntegral | Meтод вычисления | Процедурный |  |
| eps | Real | Точность вычисления | Формальный | Значение |

Таблица 11 – Данные подпрограммы SecondIntegral(Proc,eps)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра | Способ  Передачи |
| Proc | TGetIntegral | Meтод вычисления | Процедурный |  |
| eps | Real | Точность вычисления | Формальный | Значение |

# 4 Схема алгоритма решения задачи по ГОСТ 19.701-90



Рисунок – Схема алгоритма решения задачи



Рисунок 2 – Схема алгоритма процедуры GetIntegralCenter



Рисунок – Схема алгоритма процедуры GetIntegralLeft



Рисунок – Схема алгоритма процедуры SecondIntegral



Рисунок – Схема алгоритма процедуры FirstIntegral



Рисунок – Схема алгоритма процедуры Integral\_2



Рисунок – Схема алгоритма процедуры Integral\_1

# Результаты расчетов

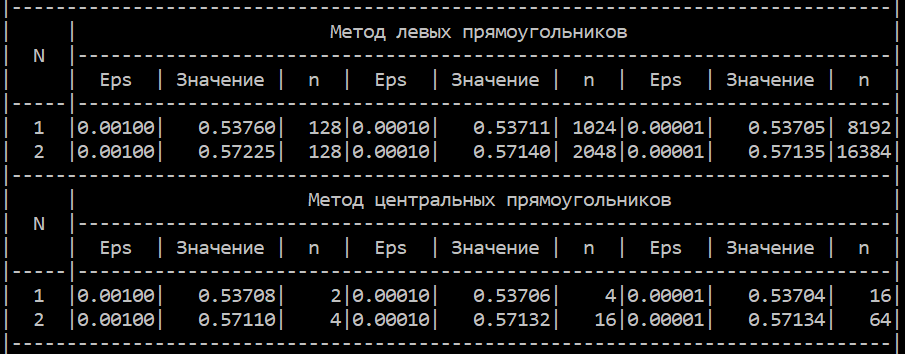


Рисунок – Результат работы программы

Приложение А

(обязательное)

Исходный код программы

**Program** RiemannSum;

{Программа для нахождения заданных интегралов

методом левых и центральных прямоугольников}

{$APPTYPE CONSOLE}

**Uses**

SysUtils,Windows;

**Type**

TIntegral = Function (const x:Real):Real;

TGetIntegral =procedure (Integral:TIntegral;

const min,max,eps:Real);

**Const**

min1=0.7;

max1=2.1;

min2=1.2;

max2=2.6;

eps:array [1..3] of Real = (0.001,0.0001,0.00001);

**Var**

i:Byte;

{Функция, задающая 1-ый интеграл}

**Function** Integral\_1(const x:Real):Real;

begin

Result:=Sqrt(0.6\*x+1.5)/(2\*x+sqrt(x\*x+3))

end;

{Функция, задающая 2-ой интеграл}

**Function** Integral\_2(const x:Real):Real;

begin

Result:=1/sqrt(x\*x+0.6)

end;

{Процедура для нахождения интеграла

методом левых прямоугольников}

**procedure** GetIntegralLeft(Integral:TIntegral;

const min,max,eps:Real);

**var**

h,Intg,Temp,R,j:Real;

n,i:Integer;

begin

n:=1;

h:=(max-min)/n; //Интервал разбиения

Intg:=h\*Integral(min+0\*h);

Temp:=Intg;

R:=eps+1;

while R >eps do //вычисление до заданной точности

begin

n:=n\*2;

h:=(max-min)/n;

Intg:=0;

{Вычисление интеграла}

for i:=0 to n-1 do

Intg:=Intg+Integral(min+i\*h);

Intg:=Intg\*h;

R:=Abs(Temp-Intg);

Temp:=Intg;

end;

{Вывод значения и числа отрезков разбиения}

write('|',Intg:10:5,'|',n:5,'|');

end;

{Процедура для нахождения интеграла

методом центральных прямоугольников}

**procedure** GetIntegralCenter(Integral:TIntegral ;

const min,max,eps:Real);

**var**

h,Intg,Temp,R:Real;

n,i:Integer;

begin

n:=1;

h:=(max-min)/n; //Интервал разбиения

Intg:=h\*Integral(min+0\*h+h/2);

Temp:=Intg;

R:=eps+1;

while R > eps do //вычисление до заданной точности

begin

n:=n\*2;

h:=(max-min)/n;

Intg:=0;

{Вычисление интеграла}

for i:=0 to n-1 do

Intg:=Intg+Integral(min+i\*h+h/2);

Intg:=Intg\*h;

R:=Abs(Temp-Intg);

Temp:=Intg;

end;

{Вывод значения и числа отрезков разбиения}

write('|',Intg:10:5,'|',n:5,'|');

end;

//Вычисление первого интеграла заданным методом

**procedure** FirstIntegral(Proc:TGetIntegral;eps:real);

begin

Proc(Integral\_1,min1,max1,eps);

end;

//Вычисление второго интеграла заданным методом

**procedure** SecondIntegral(Proc:TGetIntegral;eps:real);

begin

Proc(Integral\_2,min2,max2,eps);

end;

Begin

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

{Вывод ''шапки'' таблицы}

writeln('|-----------------------------------------------

---------------------------------|');

writeln('| | Метод левых

прямоугольников |');

writeln('| N |------------------------------------------

--------------------------------|');

writeln('| | Eps | Значение | n | Eps | Значение

| n | Eps | Значение | n |');

writeln('|-----|-----------------------------------------

---------------------------------|');

write('| 1 |' );

{Расчет 1-го интеграла методом левых

прямоугольников для 3х точностей}

for i:=1 to 3 do

begin

Write(eps[i]:7:5);

FirstIntegral(GetIntegralLeft,eps[i]);

end;

writeln;

write('| 2 |' );

{Расчет 2-го интеграла методом левых

прямоугольников для 3х точностей}

for i:=1 to 3 do

begin

Write(eps[i]:7:5);

SecondIntegral(GetIntegralLeft,eps[i]);

end;

writeln;

writeln('|-----------------------------------------------

---------------------------------|');

writeln('| | Метод центральных

прямоугольников |');

writeln('| N |------------------------------------------

--------------------------------|');

writeln('| | Eps | Значение | n | Eps | Значение

| n | Eps | Значение | n |');

writeln('|-----|-----------------------------------------

---------------------------------|');

write('| 1 |' );

{Расчет 1-го интеграла методом центральных

прямоугольников для 3х точностей}

for i:=1 to 3 do

begin

Write(eps[i]:7:5);

FirstIntegral(GetIntegralCenter,eps[i]);

end;

writeln;

write('| 2 |' );

{Расчет 2-го интеграла методом центральных

прямоугольников для 3х точностей}

for i:=1 to 3 do

begin

Write(eps[i]:7:5);

SecondIntegral(GetIntegralCenter,eps[i]);

end;

writeln;

write('|-------------------------------------------------

-------------------------------|');

Readln;

End.