**Тема лабораторной работы :** [ДЦВП с управлением по аргументу. Численное интегрирование](https://moodle.herzen.spb.ru/mod/url/view.php?id=65290)

**Цель л/р :** Научиться реализовывать вычисления определённого интеграла различными численными методами при помощи Pascal.

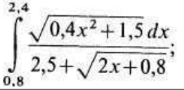
**Используемое оборудование :** draw.io(блок-схемы) , PascalABC(код программы)

**Задание 1**

**Постановка задачи:**

Написать программу для вычисления определенного интеграла из индивидуального задания методом прямоугольника левых частей.

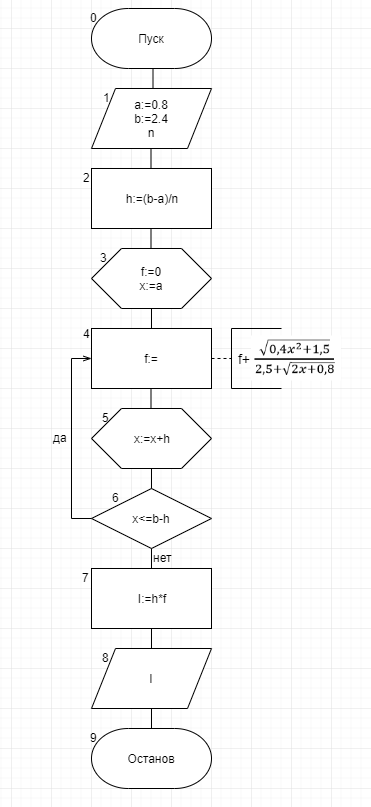
**Математическая модель:**



***h* =**

**I=**

**Блок-схема:**



**Список идентификаторов:**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| a | Нижний предел | real |
| b | Верхний предел | real |
| n | Кол-во разбиений | integer |
| f | Накопительная переменная | real |
| h | Шаг цикла | real |
| x | Параметр цикла | real |
| I | Результат | real |

**Код программы:**

**program** lr31;

**var** f, h, x, I: real;

n:integer;

**const**

a=0.8;

b=2.4;

**begin**

write('n = ');

readln(n);

h:=(b-a)/n;

f:=0;

x:=a;

**while** x<=b-h **do begin**

f:=sqrt(0.4\*x\*x+1.5)/(2.5+sqrt(2\*x+0.8))+f;

x:=x+h;

**end**;

I:=h\*f;

writeln('I = ',I:1:5);

writeln('h = ',h);

**end**.

**Результаты выполненной работы:**

n = 10

I = 0.56480

h = 0.16

n = 100

I = 0.56358

h = 0.016

n = 1000

I = 0.57050

h = 0.0016

n = 10000

I = 0.57049

h = 0.00016

**Анализ результатов вычисления:**

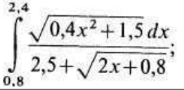
В данном случае интегрирование происходит методом прямоугольников левых частей , а значит цикл начинается в точке **а** и заканчивается на значении **b-h.** Был использован цикл **while** вместо привычного **for** , так как цикл **for** работает только в том случае ,когда параметр цикла и его шаг являются целыми числами.

**Задание 2**

**Постановка задачи:**

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

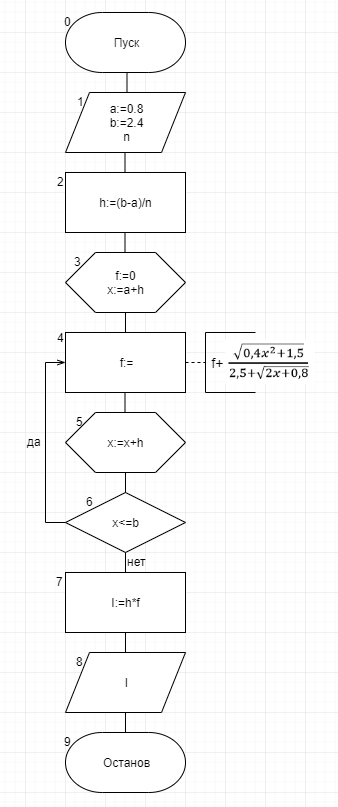
**Математическая модель:**



***h* =**

**I=**

**Блок-схема:**



**Список идентификаторов:**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| a | Нижний предел | real |
| b | Верхний предел | real |
| n | Кол-во разбиений | integer |
| f | Накопительная переменная | real |
| h | Шаг цикла | real |
| x | Параметр цикла | real |
| I | Результат | real |

**Код программы:**

**program** lr32;

**var** f, h, x, I: real;

**const**

a=0.8;

b=2.4;

n=10;

**begin**

h:=(b-a)/n;

f:=0;

x:=a+h;

**while** x<=b **do begin**

f:=sqrt(0.4\*x\*x+1.5)/(2.5+sqrt(2\*x+0.8))+f;

x:=x+h;

**end**;

I:=h\*f;

writeln('I = ',I:1:5);

writeln('h = ',h);

**end**.

**Результат выполненной работы:**

n = 10

I = 0.57657

h = 0.16

n = 100

I = 0.56474

h = 0.016

n = 1000

I = 0.57062

h = 0.0016

n = 10000

I = 0.57050

h = 0.00016

**Анализ результатов вычисления:**

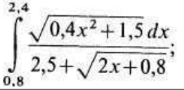
В данном случае интегрирование происходит методом прямоугольников правых частей , а значит цикл начинается на значении **а+h** и заканчивается в точке **b.** Берётся именно такое значение шага, чтобы соответствовать количеству разбиений. К примеру, если количество разбиений равно **n=10 ,** а значит и повторений цикла будет 10.

**Задание 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 3

**Математическая модель:**



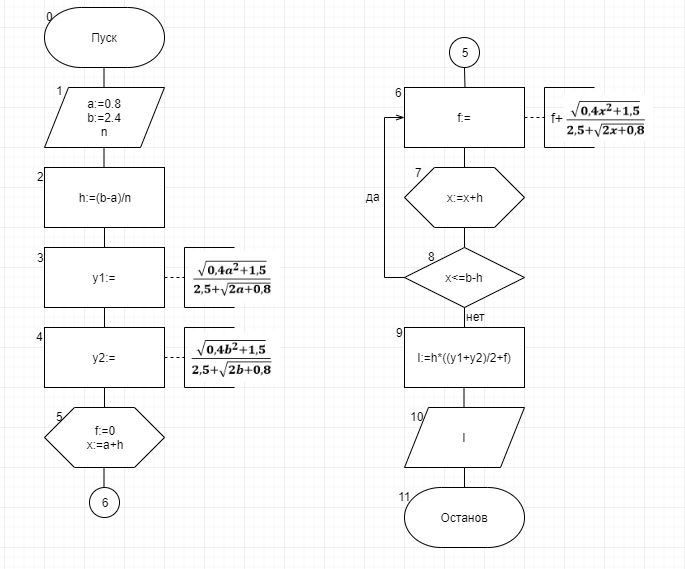
***h* =**

**y1=**

**y2=**

**I=**

**Блок-схема:**



**Список идентификаторов:**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| a | Нижний предел | real |
| b | Верхний предел | real |
| f | Накопительная переменная | real |
| n | Кол-во разбиений | integer |
| h | Шаг цикла | real |
| y1 | Знач. функции в нижнем пределе | real |
| y2 | Знач. функции в верхнем пределе | real |
| x | Параметр цикла | real |
| I | Результат | real |

**Код программы:**

**program** lr33;

**var** f,h,y1,y2,x,I:real;

n:integer;

**const**

a=0.8;

b=2.4;

**begin**

write('n = ');

readln(n);

h:=(b-a)/n;

y1:=sqrt(0.4\*a\*a+1.5)/(2.5+sqrt(2\*a+0.8));

y2:=sqrt(0.4\*b\*b+1.5)/(2.5+sqrt(2\*b+0.8));

f:=0;

x:=a+h;

**while** x<=b-h **do begin**

f:=f+ sqrt(0.4\*x\*x+1.5)/(2.5+sqrt(2\*x+0.8));

x:=x+h;

**end**;

I:=h\*((y1+y2)/2+f);

writeln('I = ',I:1:5);

writeln('h = ',h);

**end**.

**Результаты выполненной работы:**

n = 10

I = 0.57069

h = 0.16

n = 100

I = 0.56417

h = 0.016

n = 1000

I = 0.57056

h = 0.0016

n = 10000

I = 0.57050

h = 0.00016

**Анализ результатов вычислений:**

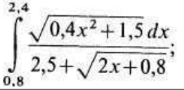
В методе трапеций цикл начинается при значении **а+h** и заканчивается при значении **b-h.** При этом крайние точки заданного отрезка вычисляются отдельно, в программе им присвоены значения y1 и y2.

**Задание 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 3

**Математическая модель:**



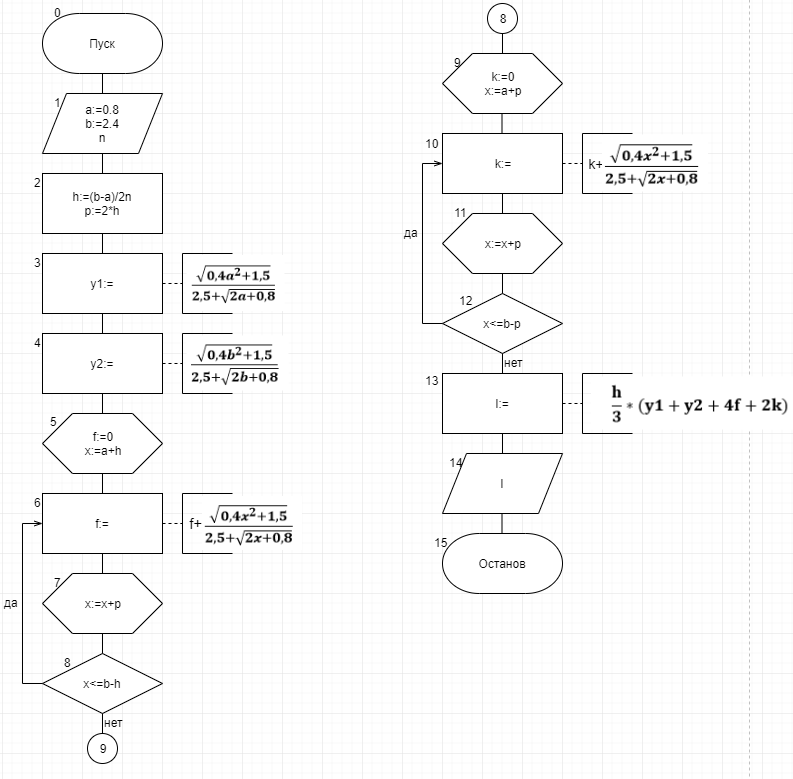
***h* =**

**y1=**

**y2=**

**I=**

**Блок-схема:**



**Список идентификаторов:**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| a | Нижний предел | real |
| b | Верхний предел | real |
| f | Накопительная переменная | real |
| n | Кол-во разбиений | integer |
| h | Шаг цикла | real |
| y1 | Знач. функции в нижнем пределе | real |
| y2 | Знач. функции в верхнем пределе | real |
| x | Параметр цикла | real |
| k | Накопительная переменная | real |
| I | Результат | real |
| p | Промежуточная переменная | real |

**Код программы:**

**program** lr34;

**var** f,h,y1,y2,x,I,k,p:real;

n:integer;

**const**

a=0.8;

b=2.4;

**begin**

write('n = ');

readln(n);

h:=(b-a)/n;

y1:=sqrt(0.4\*a\*a+1.5)/(2.5+sqrt(2\*a+0.8));

y2:=sqrt(0.4\*b\*b+1.5)/(2.5+sqrt(2\*b+0.8));

f:=0;

p:=2\*h;

x:=a+h;

**while** x<=b-h **do begin**

f:=f+ sqrt(0.4\*x\*x+1.5)/(2.5+sqrt(2\*x+0.8));

x:=x+p;

**end**;

k:=0;

x:=a+p;

**while** x<=b-p **do begin**

k:=k + sqrt(0.4\*x\*x+1.5)/(2.5+sqrt(2\*x+0.8));

x:=x+p;

**end**;

I:=h/3\*(y1+y2+4\*f+2\*k);

writeln('I = ',I:1:5);

writeln('h = ',h);

**end**.

**Результаты выполненной работы:**

n = 10

I = 0.57056

h = 0.16

n = 100

I = 0.55778

h = 0.016

n = 1000

I = 0.56928

h = 0.0016

n = 10000

I = 0.57043

h = 0.00016

**Анализ результатов вычислений:**

Метод парабол решается с помощью двух циклов. В первом высчитываются нечётные значения, а во втором чётные. Для того, чтобы всё работало правильно , шаг обоих циклов будет равен **2h.**

**Вывод:**

| **Количество разбиений**  **n** | **Шаг цикла**  **h** | **Метод левых частей прямоугольников**  **I** | **Метод правых частей прямоугольников**  **I** | **Метод трапеций**  **I** | **Метод парабол**  **I** |
| --- | --- | --- | --- | --- | --- |
| 10 | 0,16 | 0.56480 | 0.57657 | 0.57069 | 0.57056 |
| 100 | 0,016 | 0.56358 | 0.56474 | 0.56417 | 0.55778 |
| 1000 | 0,0016 | 0.57050 | 0.57062 | 0.57056 | 0.56928 |
| 10000 | 0,00016 | 0.57049 | 0.57050 | 0.57050 | 0.57043 |

Была изучена реализация вычисления определённого интеграла различными численными методами при помощи компилятора Pascal. В данном случае наиболее точным из рассмотренных методов является метод трапеций. Можно увеличить точность любого метода увеличением количества разбиений.