**Тема лабораторной работы:** ДЦВП с управлением по аргументу. Численное интегрирование (с использованием пользовательской функции)

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

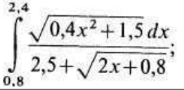
**Используемое оборудование :** draw.io(блок-схемы) , PascalABC(код программы)

**Задание 1**

**Постановка задачи:**

Реализовать вычисление определенного интеграла из индивидуального задания (взять интеграл из предыдущей лабораторной) методом трапеций с использованием пользовательской функции.

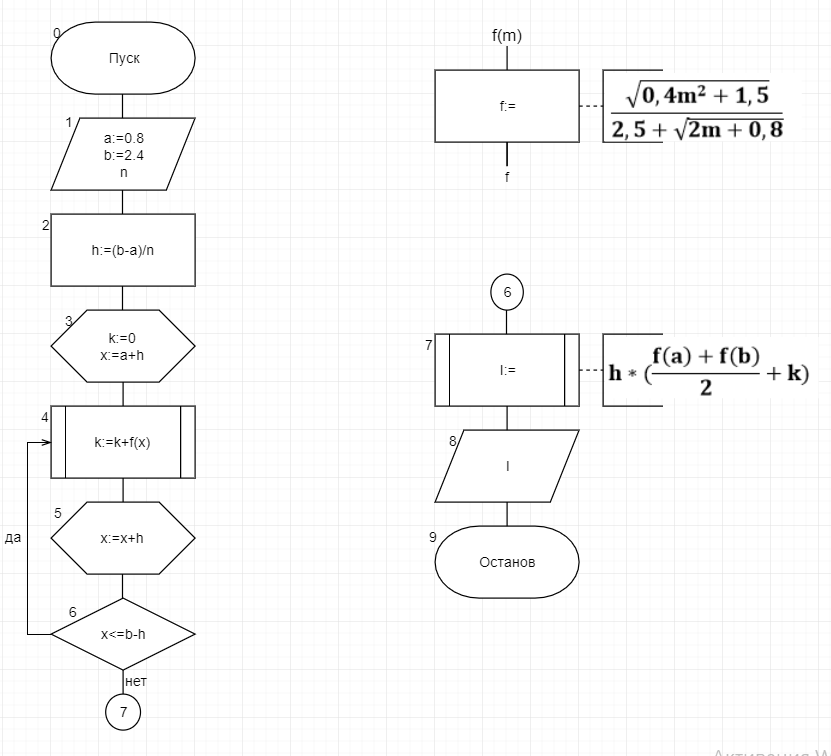
**Математическая модель:**



***h* =**

**I=**

**Блок-схема:**



**Список идентификаторов:**

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

**Код программы:**

**program** lr41;

**var**

h, x, I, k: real;

n: integer;

**const**

a = 0.8;

b = 2.4;

**function** f(m: real): real;

**begin**

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

**end**;

**begin**

write('n = ');

readln(n);

h := (b - a) / n;

k := 0;

x := a + h;

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

**begin**

k := k + f(x);

x := x + h;

**end**;

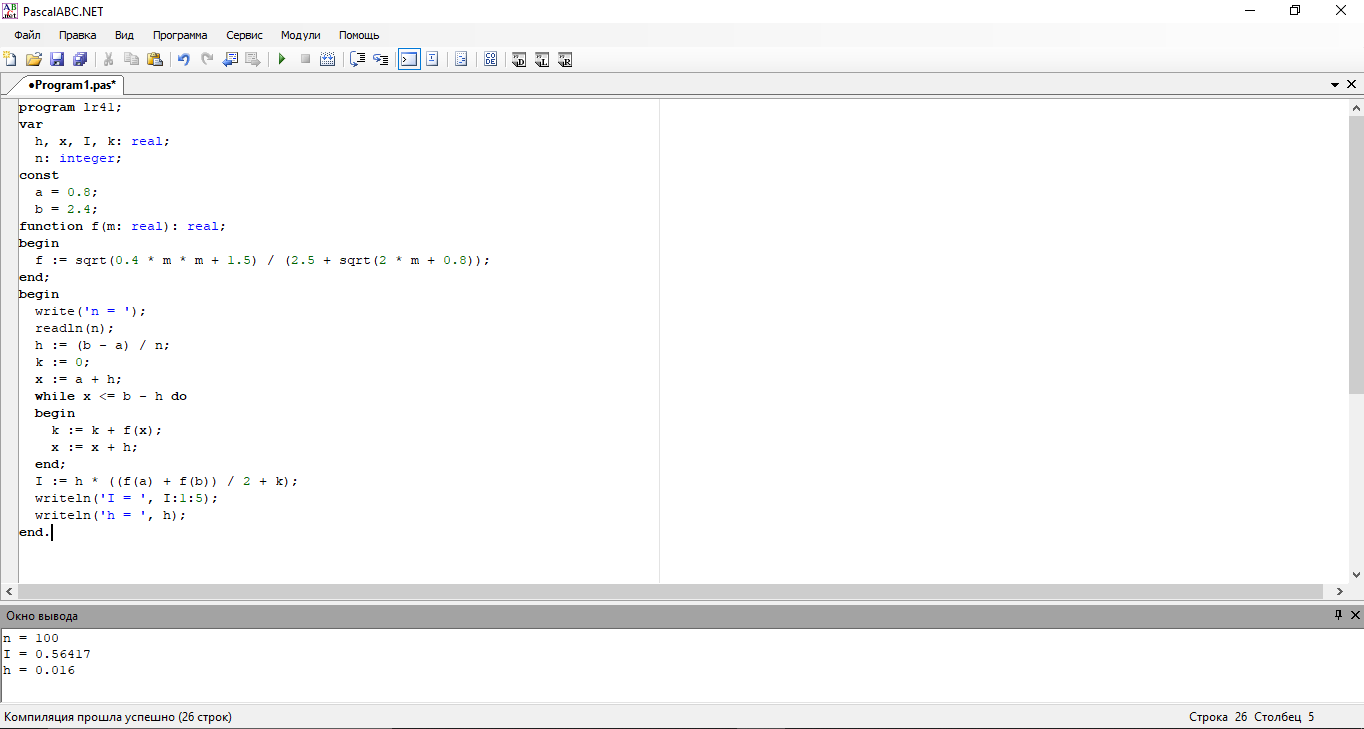
I := h \* ((f(a) + f(b)) / 2 + k);

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

writeln('h = ', h);

**end**.

**Результат выполненной работы:**



n = 10

I = 0.57069

h = 0.16

n = 1000

I = 0.57056

h = 0.0016

n = 10000

I = 0.57050

h = 0.00016

**Анализ результатов вычисления:**

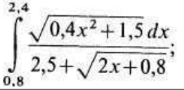
При использовании пользовательской функции отпадает необходимость присваивать значениям функции в нижнем и верхнем пределе отдельные переменные.

**Задача 2**

**Постановка задачи:**

Реализовать вычисление определенного интеграла из индивидуального задания (взять интеграл из предыдущей лабораторной) методом парабол с использованием пользовательской функции.

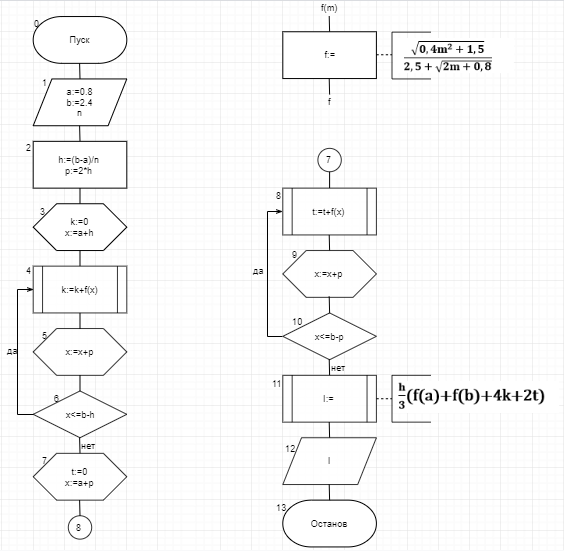
**Математическая модель:**



***h* =**

**I=**

**Блок-схема:**



**Список идентификаторов:**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| a | Нижний предел | real |
| b | Верхний предел | real |
| n | Кол-во разбиений | integer |
| h | Шаг цикла | real |
| p | Промежуточная переменная | real |
| k | Накопительная переменная | real |
| x | Параметр цикла | real |
| t | Накопительная переменная | real |
| f | Значение функции | real |
| m | Переменная функции | real |
| I | Результат | real |

**Код программы:**

**program** lr42;

**var** h, x, I, k, p, t: real; n: integer;

**const**

a = 0.8;

b = 2.4;

**function** f(m: real): real;

**begin**

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

**end**;

**begin**

write('n = ');

readln(n);

h := (b - a) / n;

k := 0;

p := 2 \* h;

x := a + h;

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

**begin**

k := k + f(x);

x := x + p;

**end**;

t := 0;

x := a + p;

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

**begin**

t := t + f(x);

x := x + p;

**end**;

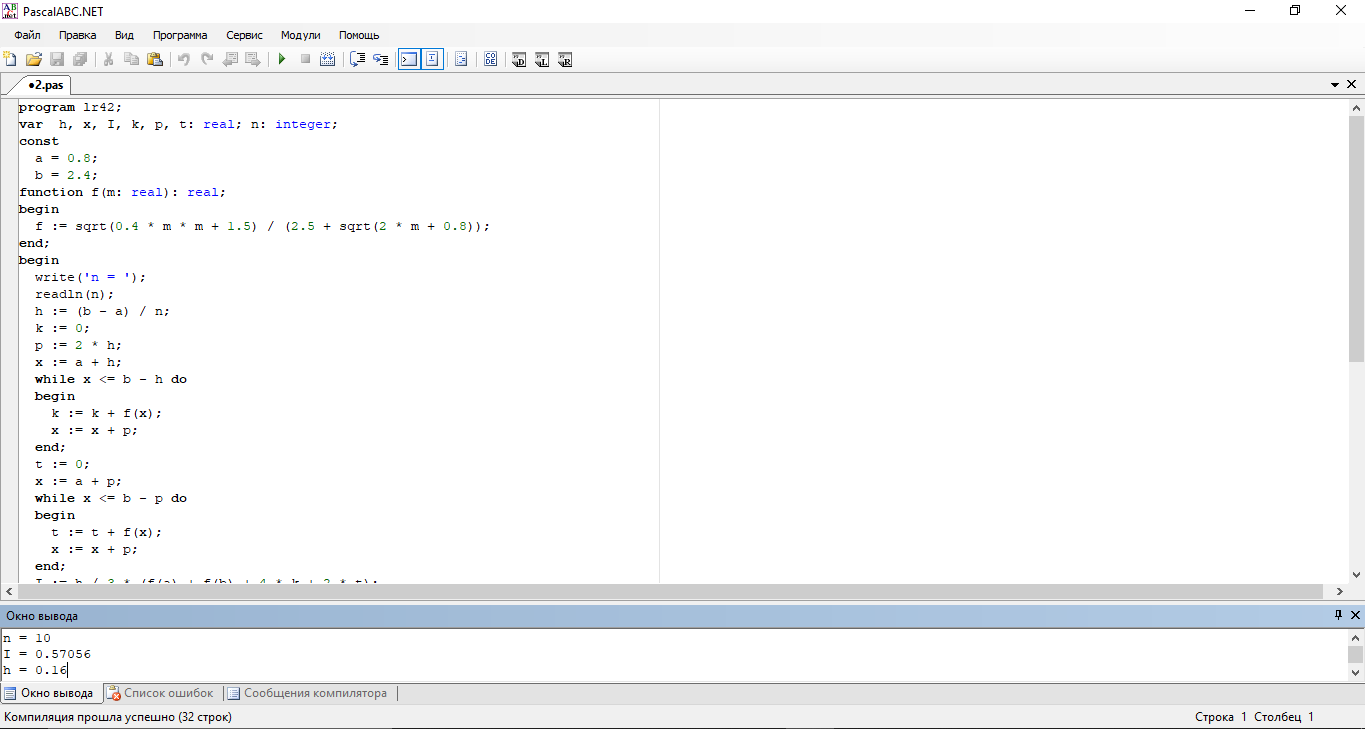
I := h / 3 \* (f(a) + f(b) + 4 \* k + 2 \* t);

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

writeln('h = ', h);

**end**.

**Результат выполненной работы:**



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 вычисления:**

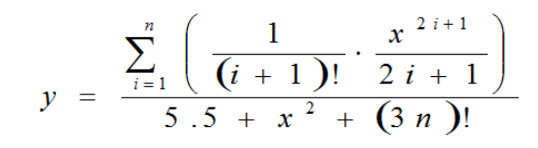
Для нахождения чётных и нечётных значений мы каждый повтор к накопительным переменным прибавляем значение функции в точке, равной параметру цикла в данный момент.

**Задача 3**

**Постановка задачи:**

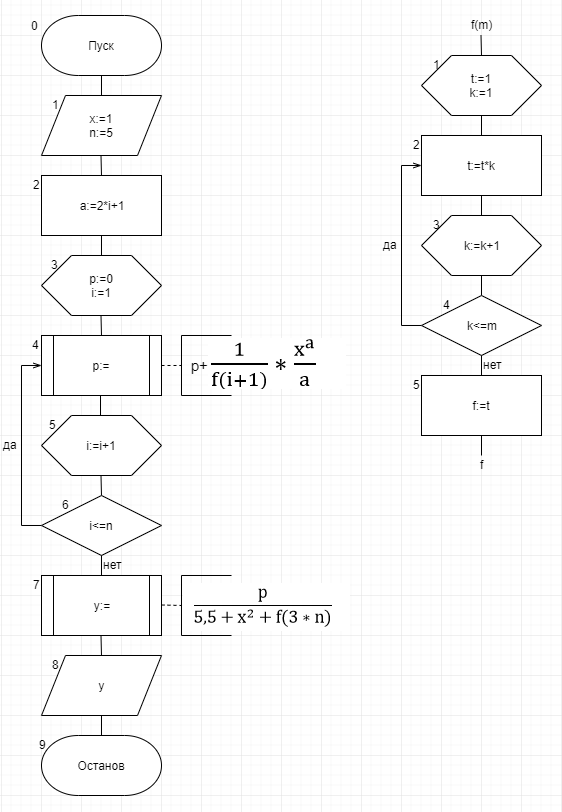
Вычислить

**Математическая модель:**



a=2i+1

**Блок-схема:**



**Список идентификаторов:**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| x | константа | integer |
| n | константа | integer |
| p | Накопительная переменная | real |
| i | Параметр цикла | integer |
| m | Переменная функции | integer |
| t | Накопительная переменная | longint |
| k | Параметр цикла | integer |
| f | Значение функции | longint |
| y | Результат | real |
| a | Промежуточная переменная | integer |

**Код программы:**

**program** lr43;

**var** i, a: integer; y, p: real;

**const**

x = 1;

n = 5;

**function** f(m: integer): longint;

**var** t: longint; k: integer;

**begin**

t := 1;

**for** k := 1 **to** m **do begin**

t := t \* k;

**end**;

f := t;

**end**;

**begin**

p := 0;

**for** i := 1 **to** n **do**

**begin**

a := 2 \* i + 1;

p := p + 1 / f(i + 1) \* (power(x , a) / a);

**end**;

y := p / (5.5 + x \* x + f(3 \* n));

writeln('y = ', y:1:20);

**end**.

**Результат выполненной работы:**

y = 0.00000000010327971580

**Анализ результатов вычисления:**

Для упрощения кода программы выводим промежуточную переменную **a=2i+1 ,** которая участвует в цикле. При помощи функции мы находим факториал любого числа и выражения.

**Вывод:**

Была изучена реализация вычисления определённого интеграла методами трапеций и парабол с использованием пользовательской функции при помощи компилятора Pascal.