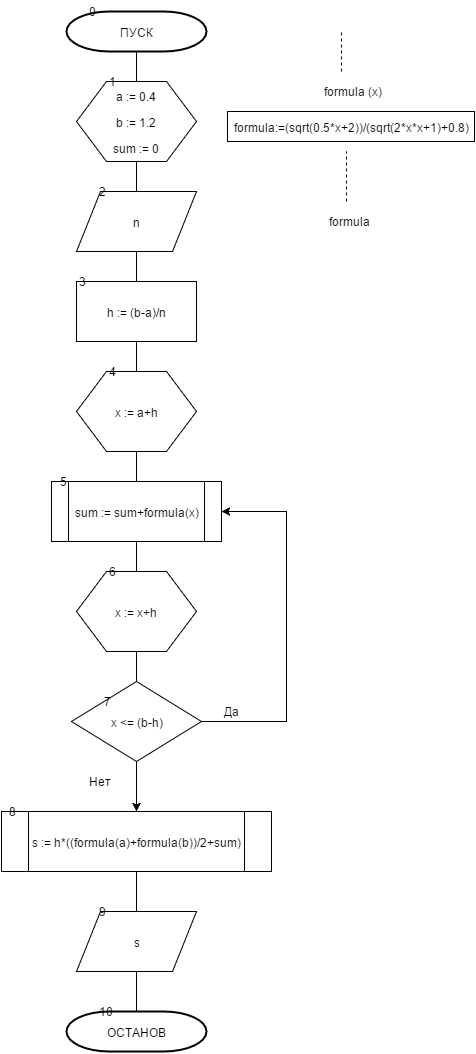
Лабораторная работа №4.

Задание 1.

1. Тема: детерминированные циклические вычислительные процессы с управлением по аргументу. Численное интегрирование.
2. Цель: научиться реализовывать алгоритмы численного интегрирования посредством детерминированных циклических вычислительных процессов с управлением по аргументу и использованием функций с помощью Free Pascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: Реализовать вычисление определенного интеграла из индивидуального задания) методом трапеций с использованием пользовательской функции.
5. Математическая модель:
6. Блок-схема: См. на следующей странице.



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| a | real | Пределы интегрирования |
| b | real | Пределы интегрирования |
| h | real | Шаг для проведения выч. |
| n | real | Кол-во отрезков разбиения |
| sum | real | Сумма |
| x | real | Параметр цикла |
| s | real | Площадь |
| Formula(x) | real | Функция |

1. Код программы:

**program** lr4\_task1;

**var**

a,b,

n,

h,

sum, x, s : real;

**function** formula(x:real):real;

**begin**

formula:=(sqrt(0.5\*x+2))/(sqrt(2\*x\*x+1)+0.8);

**end**;

**begin**

a := 0.4;

b := 1.2;

writeln('Vvedite kol-vo otrezkov razbieniy: '); readln(n);

h := (b-a)/n;

x := a+h;

sum := 0;

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

sum := sum+formula(x);

x := x+h;

**end**;

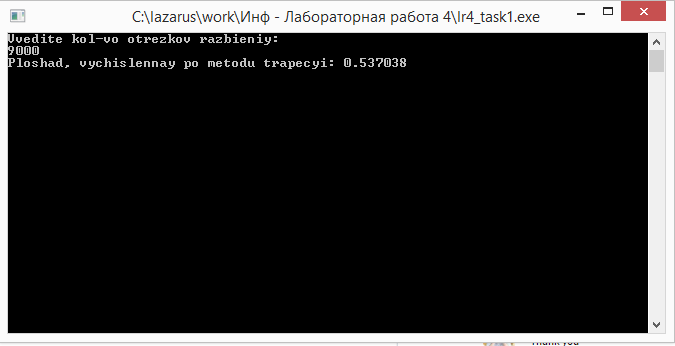
s := h\*((formula(a)+formula(b))/2+sum);

writeln('Ploshad, vychislennay po metodu trapecyi: ',s:3:6);

readln();

**end**.

1. Результат работы программы:



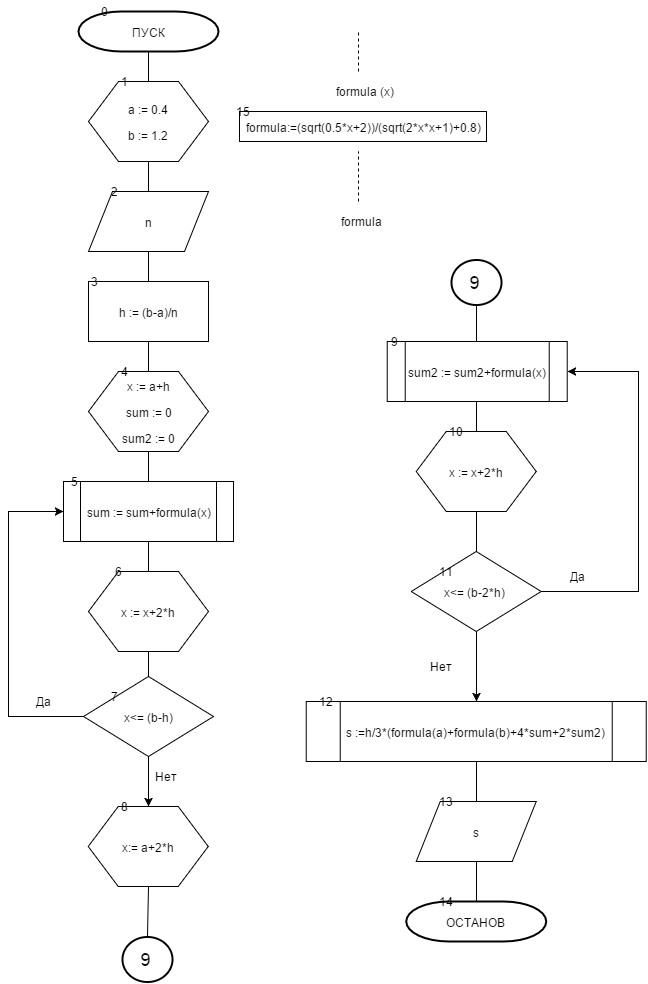
1. Анализ результатов вычисления:

Мы задаем функцию, к которой мы будем впоследствии обращаться. Задаем пределы интегрирования и кол-во отрезков разбиения. Вычисляем интеграл по методу трапеций, обращаясь к выполнению функции.

1. Вывод: мы написали программу, которая вычисляет определенный интеграл из индивидуального задания методом трапеций с использованием пользовательской функции.

Задание 2.

1. Тема: детерминированные циклические вычислительные процессы с управлением по аргументу. Численное интегрирование.
2. Цель: научиться реализовывать алгоритмы численного интегрирования посредством детерминированных циклических вычислительных процессов с управлением по аргументу и использованием функций с помощью Free Pascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: Написать программу для вычисления определенного интеграла из индивидуального задания методом парабол с использованием пользовательской функции.
5. Математическая модель:
6. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| a | real | Пределы интегрирования |
| b | real | Пределы интегрирования |
| h | real | Шаг для проведения выч. |
| n | real | Кол-во отрезков разбиения |
| sum | real | Сумма |
| Sum2 | real | Сумма |
| x | real | Параметр цикла |
| s | real | Площадь |
| Formula(x) | real | Функция |

1. Код программы:

program lr4\_task2;

var

a,b,

n,

h,

sum, sum2, x, s : real;

function formula(x:real):real;

begin

formula := (sqrt(0.5\*x+2))/(sqrt(2\*x\*x+1)+0.8);

end;

begin

a := 0.4;

b := 1.2;

writeln('Vvedite kol-vo otrezkov razbieniy: '); readln(n);

h := (b-a)/n;

x := a+h;

sum := 0;

sum2 := 0;

while x <= (b-h) do begin

sum := sum+formula(x);

x := x+2\*h;

end;

x := a+2\*h;

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

sum2 := sum2+formula(x);

x := x+2\*h;

end;

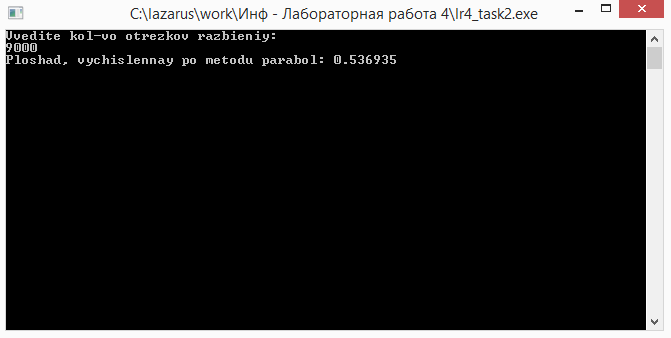
s := h/3\*(formula(a)+formula(b)+4\*sum+2\*sum2);

writeln('Ploshad, vychislennay po metodu parabol: ',s:3:6);

readln();

end.

1. Результат работы программы:



1. Анализ результатов вычисления:

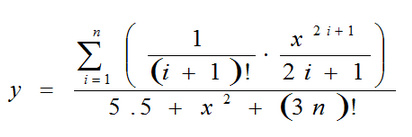
Мы задаем функцию, к которой мы будем впоследствии обращаться. Задаем пределы интегрирования и кол-во отрезков разбиения. Определяем шаг, обнуляем суммы. Вычисляем интеграл по методу парабол, обращаясь к выполнению функции.

1. Вывод:

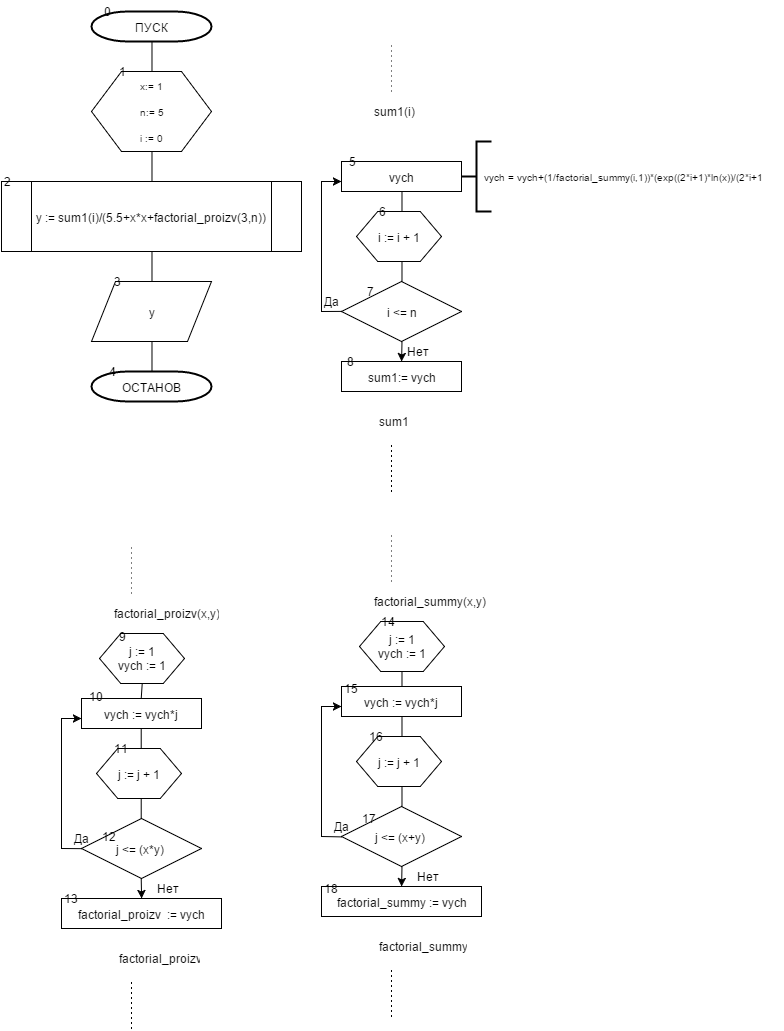
мы написали программу, которая вычисляет определенный интеграл из индивидуального задания методом парабол с использованием пользовательской функции.

Задание 3.

1. Тема: детерминированные циклические вычислительные процессы с управлением по аргументу. Численное интегрирование.
2. Цель: научиться реализовывать алгоритмы численного интегрирования посредством детерминированных циклических вычислительных процессов с управлением по аргументу и использованием функций с помощью Free Pascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: Написать программу для вычисления заданного выражения с использованием пользовательских функций.
5. Математическая модель:



1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| y | real | Искомое значение |
| n | integer | Входные данные, параметр цикла |
| i | integer | Параметр цикла |
| x | integer | Входные данные |
| factorial\_summy(x,y) | longint | Функц. Вычисления факториала для выражения (x+y) |
| factorial\_proizv(x,y) | longint | Функц. Вычисления факториала для выражения (x\*y) |
| sum1(I,x) | real | Функц. Вычисления суммы |
| j | longint | Локальная переменная функции. Параметр внутреннего цикла |
| vych | longint | Локальная переменная функции. Промежуточный результат вычислений. |

1. Код программы:

**program** lr4\_task3;

**var**

y : real;

n,i,x : integer;

**function** factorial\_summy(x,y:longint):longint;

**var** j,vych: longint;

**begin**

vych := 1;

**for** j := 1 **to** (x+y) **do**

vych := vych\*j;

factorial\_summy := vych;

**end**;

**function** factorial\_proizv(x,y:integer):longint;

**var** j,vych: longint;

**begin**

vych := 1;

**for** j := 1 **to** (x\*y) **do**

vych := vych\*j;

factorial\_proizv := vych;

**end**;

**function** sum1(i:integer):real;

**var**

vych : real;

**begin**

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

vych := vych+(1/factorial\_summy(i,1))\*(exp((2\*i+1)\*ln(x))/(2\*i+1));

sum1:= vych;

**end**;

**begin**

x:= 1;

n:= 5;

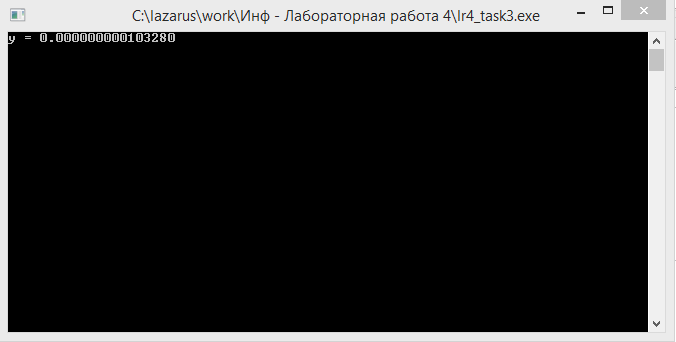
y := sum1(i)/(5.5+x\*x+factorial\_proizv(3,n));

writeln('y = ',y:2:15);

readln();

**end**.

1. Результат работы программы:



1. Анализ результатов вычисления:

Мы задаем функцию, к которым будем впоследствии обращаться: вычисление факториала суммы, произведения, а также подсчёт суммы. Затем присваиваем переменным x и n нужные значения, после чего производим вычисления по формуле, используя заданные функции.

1. Вывод: мы написали программу для вычисления заданного выражения с использованием пользовательских функций.