Лабораторная работа №3.

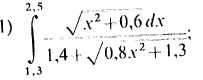
**Тема:**

**Цель:**

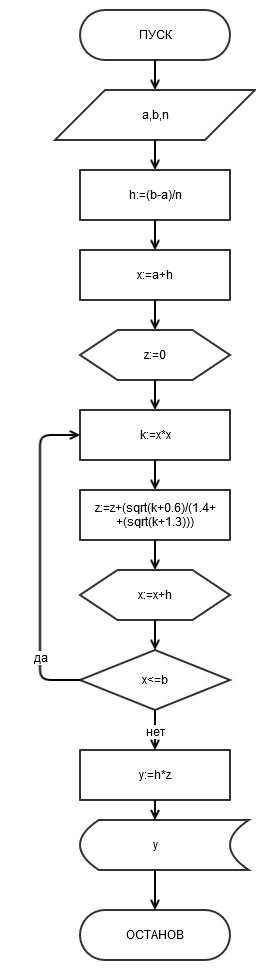
**Используемое оборудование: ПК, среда программирования Lazarus.**

**Задание 1.**

1. Постановка задачи: Вычислить интеграл из индивидуального задания с помощью метода правых частей прямоугольника.
2. Математическая модель:



1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| A | Real | Нижний предел интегрирования |
| B | Real | Верхний предел интегрирования |
| H | Real | Шаг |
| z | Real | Сумма |
| K | Real | Вспомогательная переменная |
| N | Real | Количество отрезков |
| X | Real | Переменная |
| Y | real | Функция |

1. Код программы:

program metod\_pravyh\_chastey;

Var

a,b,z,h,k,n,x,y:real;

begin

writeln('vvedite n');

readln(n);

writeln('vvedite a');

readln(a);

writeln('vvedite b');

readln(b);

h:=(b-a)/n;

writeln(h:2:4);

x:=a+h;

z:=0;

while x <= b do

begin

k:=x\*x;

z:=z+(sqrt(k+0.6)/(1.4+sqrt(k+1.3)));

x:=x+h;

end;

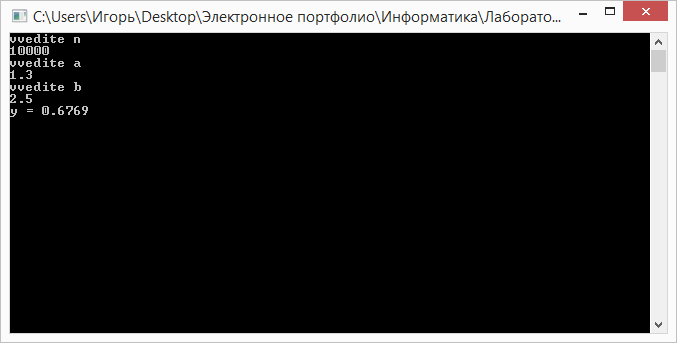
y:=h\*z;

writeln('y = ', y:2:4);

readln()

end.

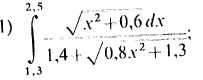
1. Результат выполнения программы:



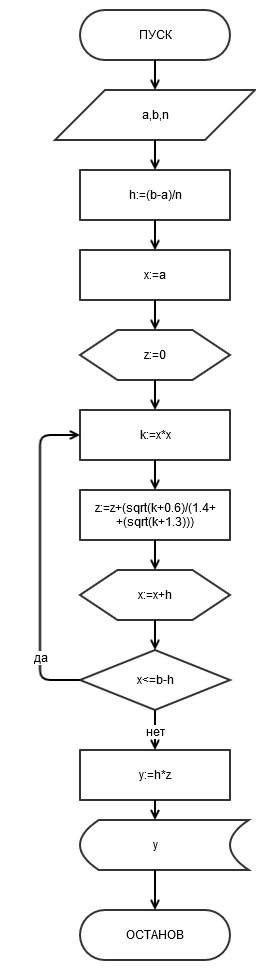
1. Анализ результатов вычисления: При разбиении на 10000 отрезков интеграл равен 0,6769
2. Вывод: Программа работает корректно

**Задание 2.**

1. Постановка задачи: Вычислить интеграл из индивидуального задания с помощью метода левых частей прямоугольника.
2. Математическая модель:



1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| A | Real | Нижний предел интегрирования |
| B | Real | Верхний предел интегрирования |
| H | Real | Шаг |
| z | Real | Сумма |
| K | Real | Вспомогательная переменная |
| N | Real | Количество отрезков |
| X | Real | Переменная |
| y | Real | Функция |

1. **Код программы:**

program metod\_levyh\_chastei;

Var

a,b,z,h,n,k,x,y:real;

begin

writeln('vvedite n');

readln(n);

writeln('vvedite a');

readln(a);

writeln('vvedite b');

readln(b);

h:=(b-a)/n;

x:=a;

z:=0;

while x <= b-h do

begin

k:=x\*x;

z:=z+(sqrt(k+0.6)/(1.4+sqrt(k+1.3)));

x:=x+h;

end;

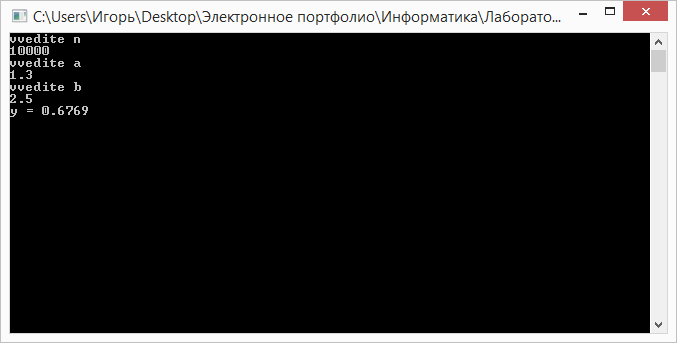
y:=h\*z;

writeln('y = ', y:2:4);

readln()

end.

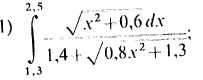
1. Результат выполнения программы:



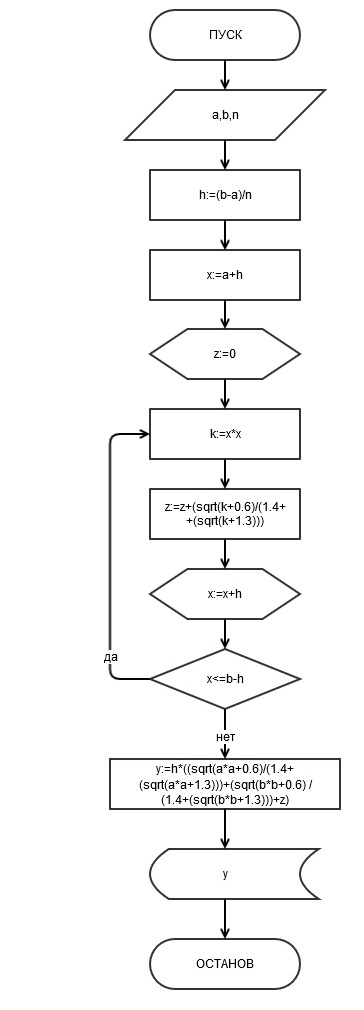
1. Анализ результатов вычисления: При разбиении на 10000 отрезков интеграл равен 0,6769
2. Вывод: Программа работает корректно

**Задание 3.**

1. Постановка задачи: Вычислить интеграл из индивидуального задания с помощью метода трапеций.
2. Математическая модель:



1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| A | Real | Нижний предел интегрирования |
| B | Real | Верхний предел интегрирования |
| H | Real | Шаг |
| z | Real | Сумма |
| K | Real | Вспомогательная переменная |
| N | Real | Количество отрезков |
| X | Real | Переменная |
| Y | Real | Функция |

1. Код программы:

program metod\_trapetsyi;

Var

a,b,z,h,k,n,x,y:real;

begin

writeln('vvedite n');

readln(n);

writeln('vvedite a');

readln(a);

writeln('vvedite b');

readln(b);

h:=(b-a)/n;

x:=a+h;

z:=0;

while x <= b-h do

begin

k:=x\*x

z:=z+(sqrt(k+0.6)/(1.4+sqrt(k+1.3)));

x:=x+h;

end;

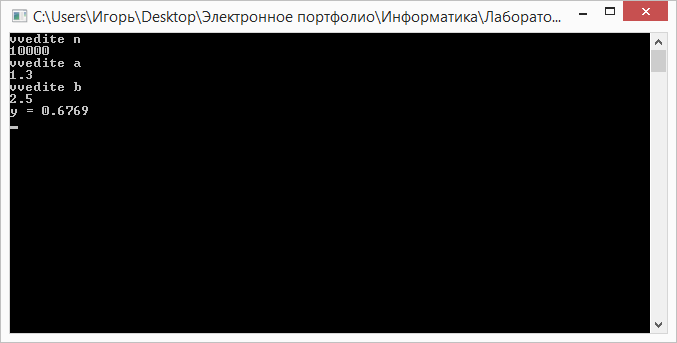
y:=h\*((sqrt(a\*a+0.6)/(1.4+sqrt(a\*a+1.3)))+(sqrt(b\*b+0.6)/(1.4+sqrt(b\*b+1.3)))+z);

writeln('y = ', y:2:4);

readln()

end.

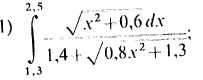
1. Результат выполнения программы:



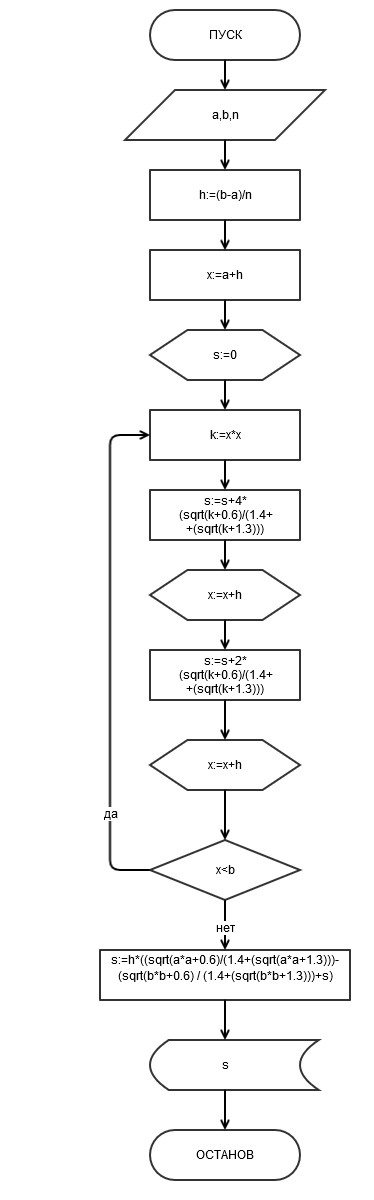
1. Анализ результатов вычисления: При разбиении на 10000 отрезков интеграл равен 0,6769
2. Вывод: Программа работает корректно

**Задание 4.**

1. Постановка задачи: Вычислить интеграл из индивидуального задания с помощью метода парабол.
2. Математическая модель:



1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| A | Real | Нижний предел интегрирования |
| B | Real | Верхний предел интегрирования |
| H | Real | Шаг |
| s | Real | Сумма |
| K | Real | Вспомогательная переменная |
| N | Real | Количество отрезков |
| X | Real | Переменная |
| Y | Real | Функция |

1. Код программы:

program metod\_parabol;

var

x,a,b,h,s,k,y:real;

n:integer;

begin

writeln('vvedite a');

readln(a);

writeln('vvedite b');

readln(b);

writeln('vvedite n');

readln(n);

h:=(b-a)/n;

s:=0;

x:=a+h;

while x<b do

begin

k:=x\*x;

s:=s+4\*(sqrt(k+0.6)/(1.4+sqrt(k+1.3)));

x:=x+h;

s:=s+2\*(sqrt(k+0.6)/(1.4+sqrt(k+1.3)));

x:=x+h;

end;

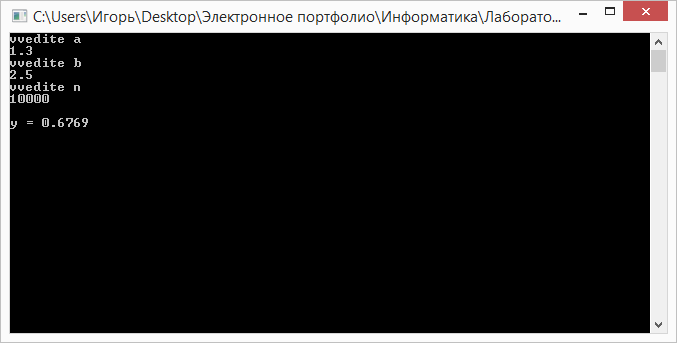
y:=h/3\*(s+(sqrt(a\*a+0.6)/(1.4+sqrt(a\*a+1.3)))-(sqrt(b\*b+0.6)/(1.4+sqrt(b\*b+1.3))));

writeln('y = ', s:2:4);

readln;

end.

1. Результат выполнения программы:



1. Анализ результатов вычисления: При разбиении на 10000 отрезков интеграл равен 0,6769
2. Вывод: Программа работает корректно

**Результаты вычислений**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| N  Количество разбиений | H  шаг | Y  Метод левых частей прямоугольников | I  Метод правых частей прямоугольников | I  Метод трапеций | I  Метод Парабол |
| 10 | 0,12 | 0,5936 | 0,6099 | 0,6693 | 0,6711 |
| 100 | 0,012 | 0,6684 | 0,6702 | 0,6760 | 0,6763 |
| 1000 | 0,0012 | 0,6768 | 0,6769 | 0,6775 | 0,6768 |
| 10000 | 0,00012 | 0,6769 | 0,6769 | 0,6769 | 0,6769 |

Из данных результатов можно сделать вывод, что для увеличения точности вычисления интеграла необходимо увеличить количество отрезков, на которые разбивается интеграл. Для вычисления данного интеграла лучше всего использовать метод парабол, как самый точный.