Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №4**

**«Работа в графическом режиме»**

**ПО ДИСЦИПЛИНЕ «Основы алгоритмизации и программирования»**

Выполнил: студент учебной группы

ИСПк-202-52-00

Лаптев Владимир Антонович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

**Цель работы**

Освоить принципы работы в графическом режиме; получить базовые навыки взаимодействия с графическими примитивами.

**Формулировка задания**

1. Дополнить программу, реализованную в ходе предыдущей лабораторной работы, режимом визуализации.

2. Предусмотреть возможность вывода кривой, ограничивающей фигуру, на координатную плоскость.

3. Реализовать следующие возможности и элементы: масштабирование графика, подписи на осях, вывод информации о задании.

4. Реализовать не менее двух возможностей из представленных независимое масштабирование по осям, штриховка вычисляемой площади, визуализация численного расчета интеграла.

**Описание алгоритма**

Была написана процедура, которая высчитывает значение интеграла функции. Была написана процедура, которая высчитывает значение площади функции по методу средних прямоугольников. С помощью GraphABC была сделана графическая визуализация функции, также было графически показано разбиение функции по методу средних прямоугольников. Также была совершена штриховка прямоугольников.

**Схема алгоритма с комментариями**

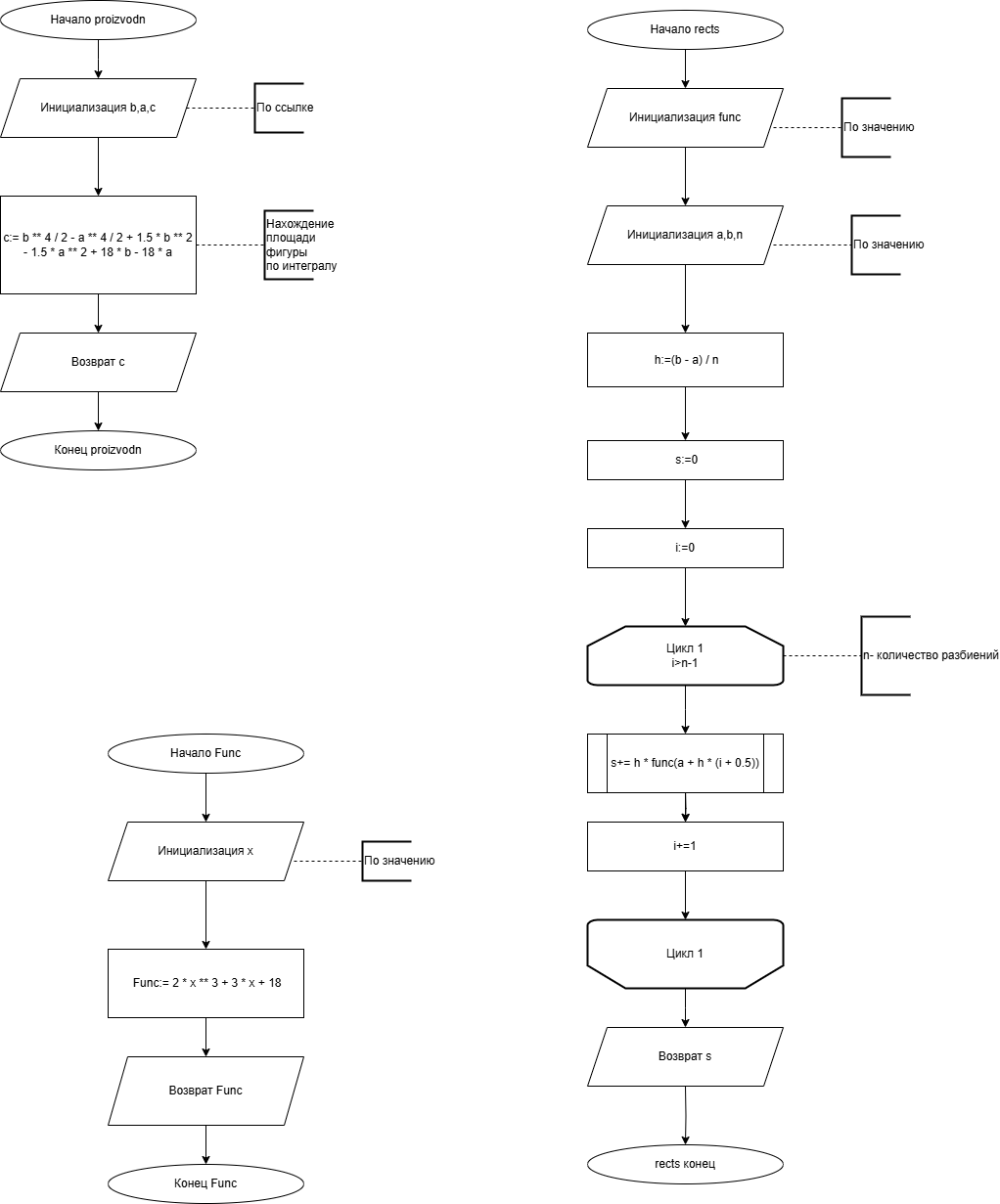


Рисунок 1 — Алгоритм решения задачи

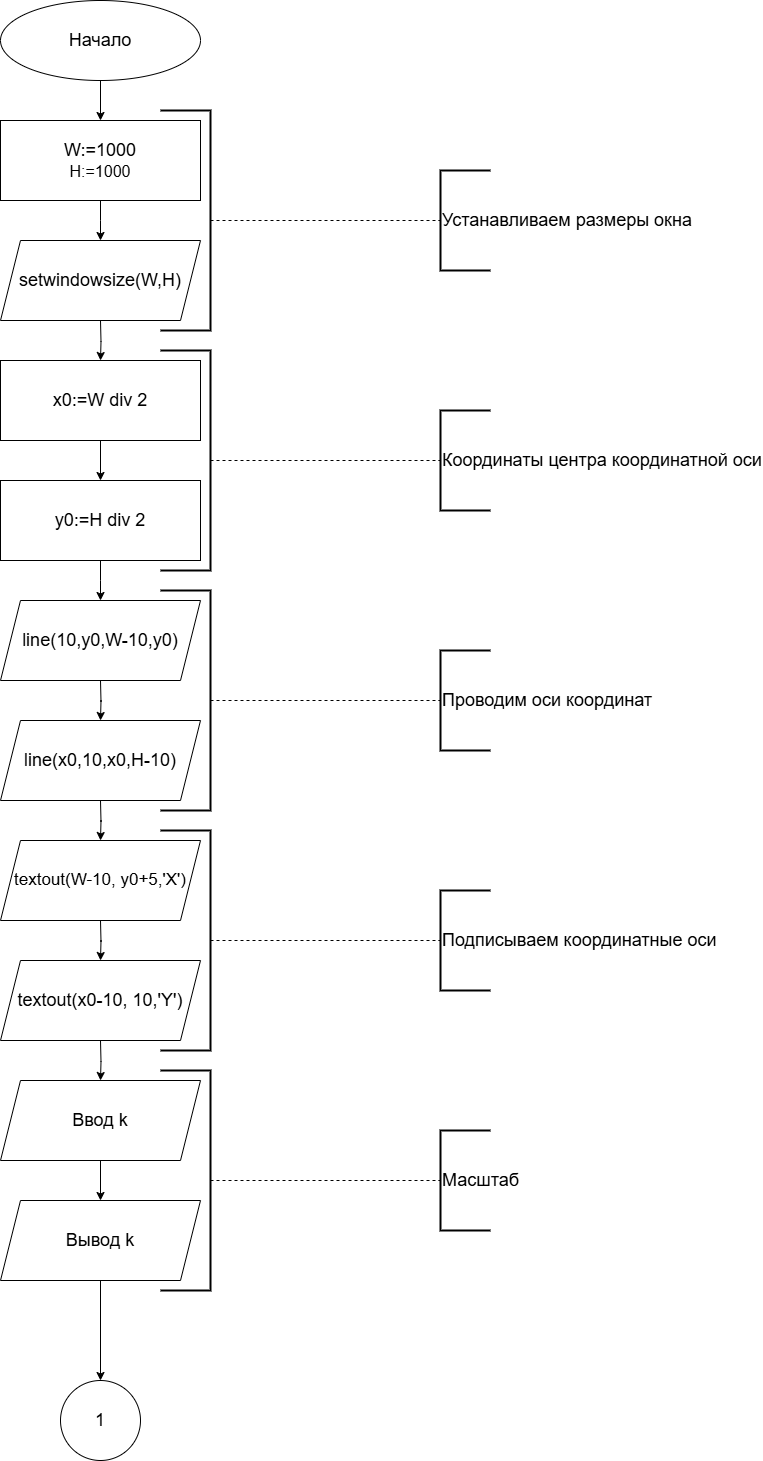


Рисунок 2 — Алгоритм решения задачи

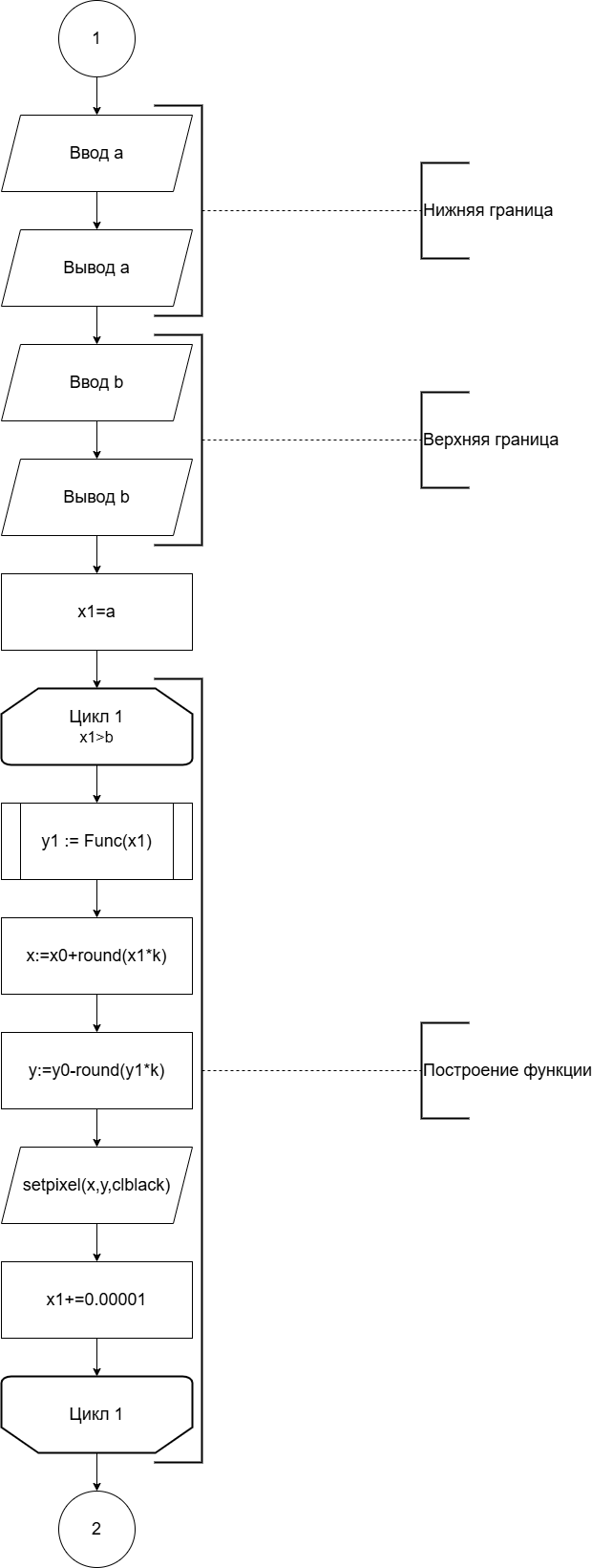


Рисунок 3 — Алгоритм решения задачи

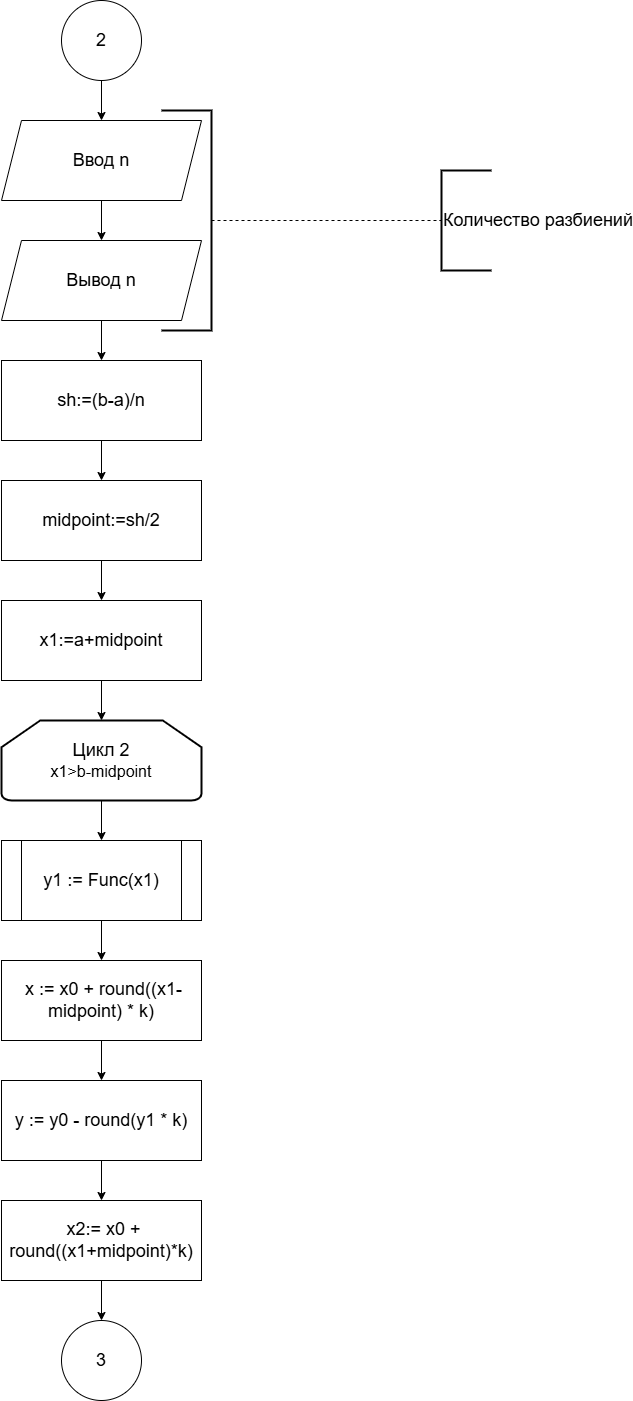


Рисунок 4 — Алгоритм решения задачи

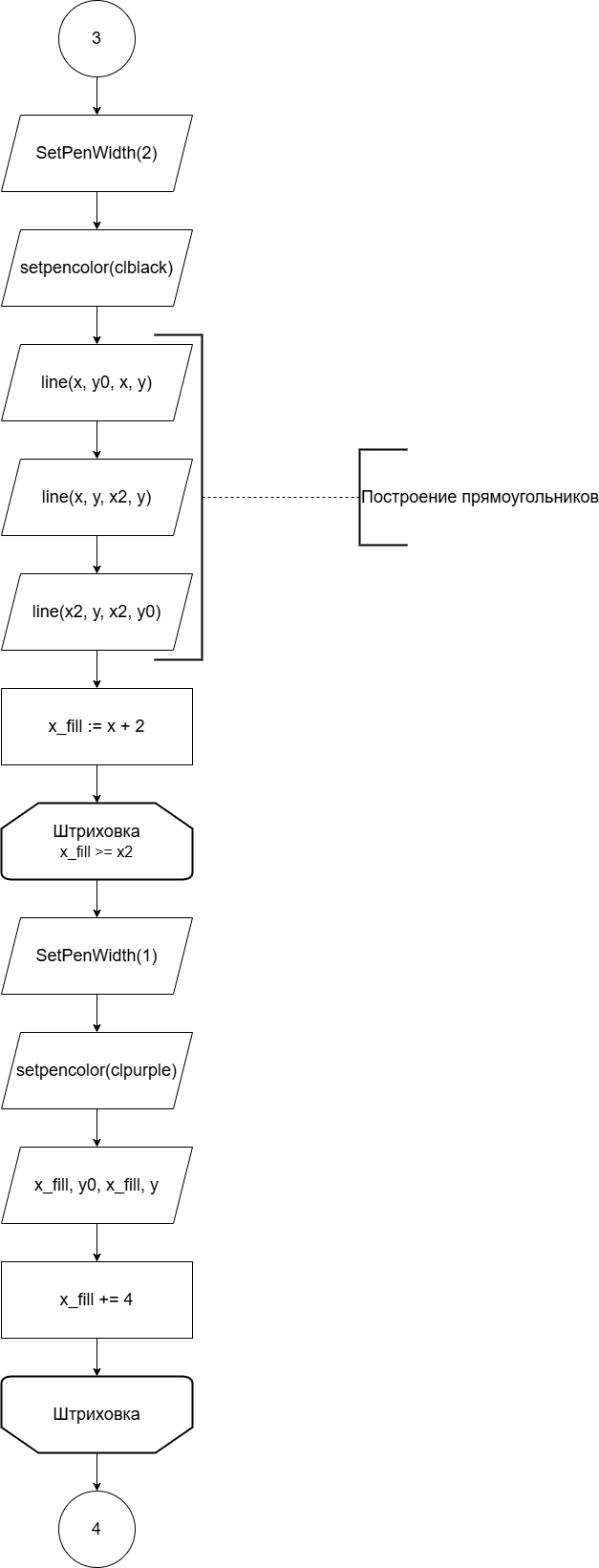


Рисунок 5 — Алгоритм решения задачи

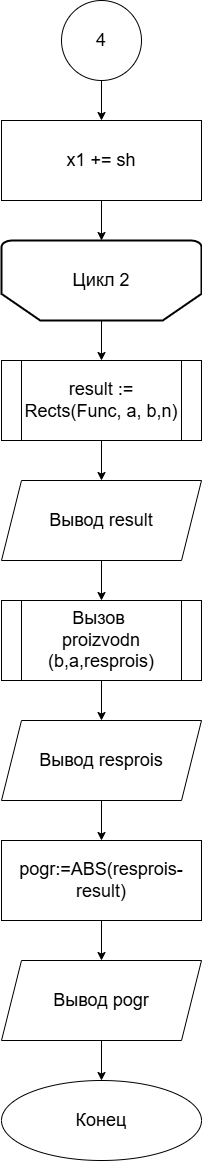


Рисунок 6 — Алгоритм решения задачи

**Код программы**

program qq;

uses GraphABC;

function Rects(func: function(x: real): real; a, b: real; n: integer): real;

var

h, s: real;

i: integer;

begin

h := (b - a) / n;

s := 0;

for i := 0 to n-1 do

begin

s := s + h \* func(a + h \* (i + 0.5));

end;

Rects := s;

end;

procedure proizvodn(var b:real; var a:real; var c:real);

begin

c:=b\*\*4/2-a\*\*4/2+1.5\*b\*\*2-1.5\*a\*\*2+18\*b-18\*a;

end;

var W,H,x0,y0,x,y,n,x2,x\_fill:integer;

x1,y1,a,b,k,sh,midpoint,result,resprois, pogr:real;

function Func(x: real): real;

begin

Func := 2\*x\*x\*x+3\*x+18;

end;

begin

W:=1000;

H:=1000;

setwindowsize(W,H);

x0:=W div 2;

y0:=H div 2;

line(10,y0,W-10,y0);

line(x0,10,x0,H-10);

textout(W-10, y0+5,'X');

textout(x0-10, 10,'Y');

write('Введите масштаб ');

read(k);

writeln(k);

write('Введите нижнюю границу ');

read(a);

writeln(a);

write('Введите верхнюю границу ');

read(b);

writeln(b);

x1:=a;

while x1<=b do

begin

y1:=func(x1);

x:=x0+round(x1\*k);

y:=y0-round(y1\*k);

setpixel(x,y,clblack);

x1+=0.00001;

end;

write('Введите количество разбиений ');

read(n);

writeln(n);

sh:=(b-a)/n;

midpoint:=sh/2;

x1:=a+midpoint;

while x1 <= (b-midpoint) do

begin

y1 := Func(x1);

x := x0 + round((x1-midpoint) \* k);

y := y0 - round(y1 \* k);

x2:= x0 + round((x1+midpoint)\*k);

SetPenWidth(2);

setpencolor(clblack);

line(x, y0, x, y);

line(x, y, x2, y);

line(x2, y, x2, y0);

x\_fill := x + 2;

while x\_fill < x2 do

begin

SetPenWidth(1);

setpencolor(clpurple);

line(x\_fill, y0, x\_fill, y);

x\_fill += 4;

end;

x1 += sh;

end;

result := Rects(Func, a, b,n);

writeln('Результат интегрирования по методу средних прямоугольников: ', result:0:6);

proizvodn(b,a,resprois);

writeln('Результат по производной: ',resprois);

pogr:=ABS(resprois-result);

writeln('Погрешность: ',pogr)

end.

**Результат выполнения программы**

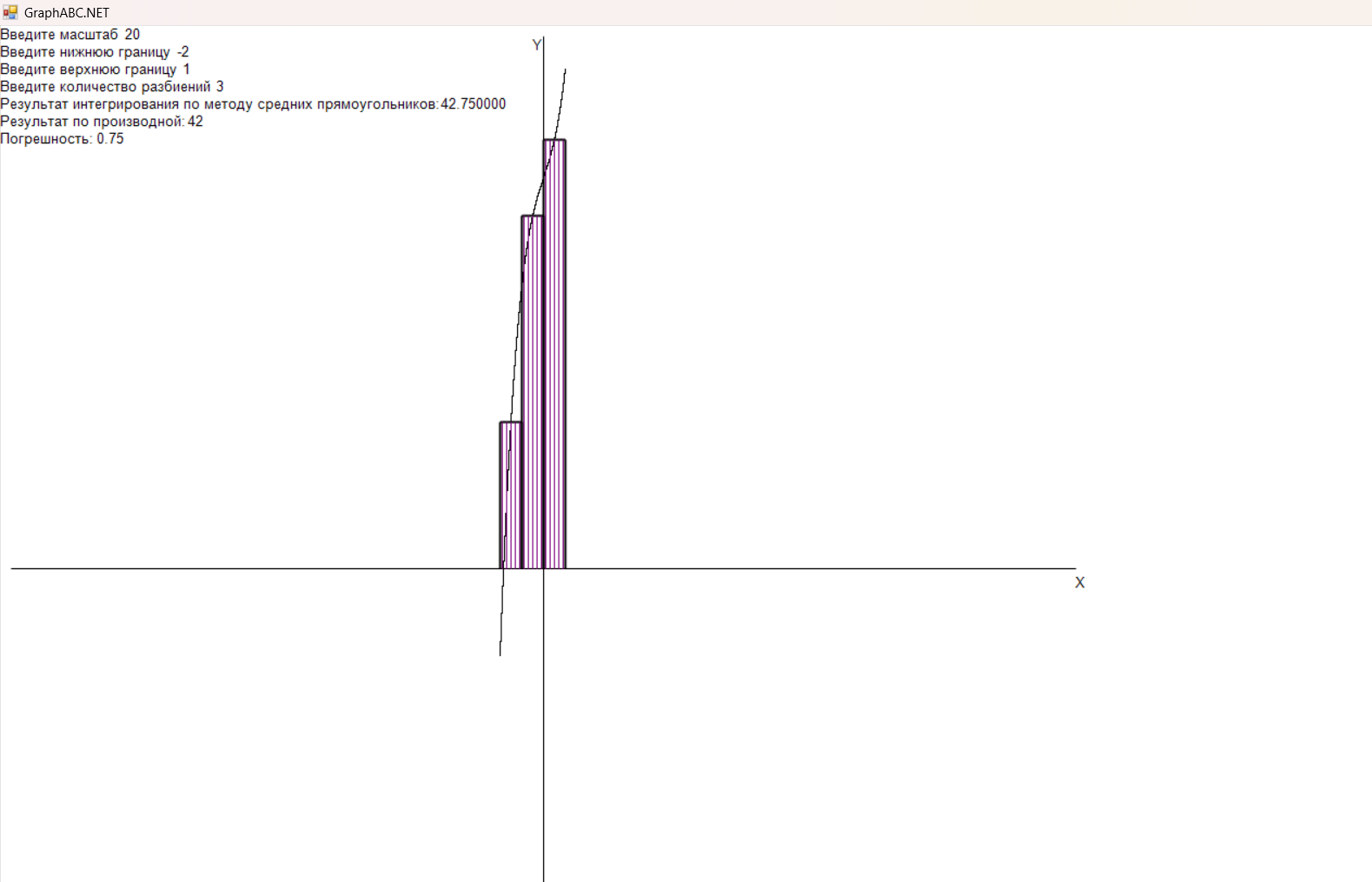


Рисунок 7 — Результат выполнения программы

**Вывод**

В ходе выполнения контрольной работы на тему "Работа в графическом режиме Pascal" были изучены основные принципы графического программирования в языке Pascal.

Мы рассмотрели:

1. Создание графического окна - использование методов и функций для инициализации графического режима, а также настройки окна для отображения графики.

2. Рисование графических объектов - освоены базовые примитивы, такие как линии, круги, и прямоугольники, а также применение различных цветов и режимов заливки.

3. Обработка событий - обсуждены методы взаимодействия с пользователем, включая обработку нажатий клавиш и мыши, что позволяет создавать более динамичные и интерактивные приложения.

В результате проведенной работы было понятно, что графический режим Pascal предоставляет множество возможностей для создания различных приложений, включая игры и учебные программы. Полученные знания о графической библиотеке, такие как Graph, являются полезными для дальнейшего изучения программирования и разработки графических интерфейсов.

Таким образом, контрольная работа позволила не только углубить теоретические знания, но и получить практические навыки работы в графическом режиме, что будет полезно в будущем.