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

Ситников Иван Владимирович

Преподаватель:

Кузьминых Ангелина Владимировна

Киров

2023

**Цель работы:** освоить принципы работы в графическом режиме; получить базовые навыки взаимодействия с графическими примитивами.

**Задание**

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.**

Алгоритм визуализирует расчёт интеграла, между графиком функции 2\*x^3+(-1)\*x^2+(1)\*x+(11) и осью OX (в положительной части оси OY).

1. Визуализация графика функции:

Сначала строится первая четверть координатной плоскости (положительные значения x и y).

1. Построение графика:

Построение графика функции в координатной плоскости.

1. Визуализация расчёта интеграла:

Под графиком функции строятся прямоугольники, визуализирующие вычисление интеграла.

1. Масштабирование графика:

Масштабирование графика функции относительно оси X и оси Y.

**Схема алгоритма**

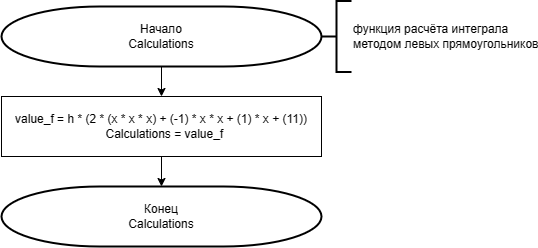


Рисунок 1.1 – Функция Calculations

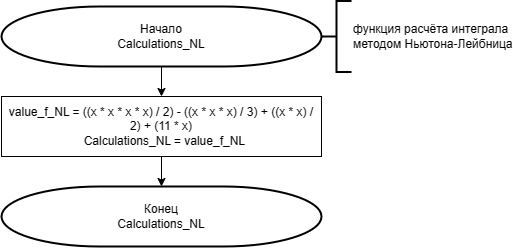


Рисунок 1.2 – Функция Calculations\_NL

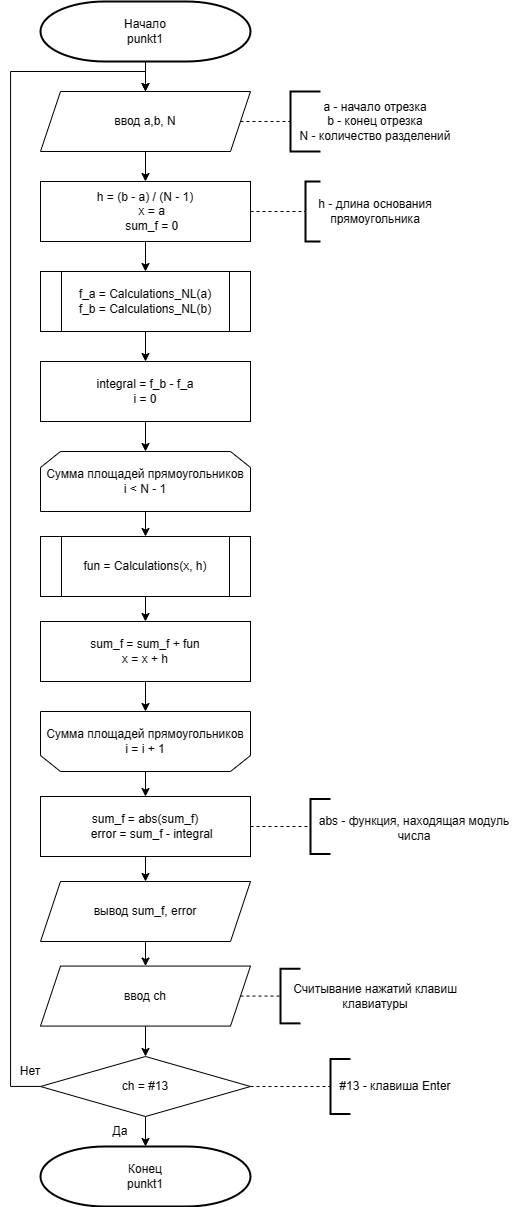


Рисунок 1.3 – Процедура punkt1

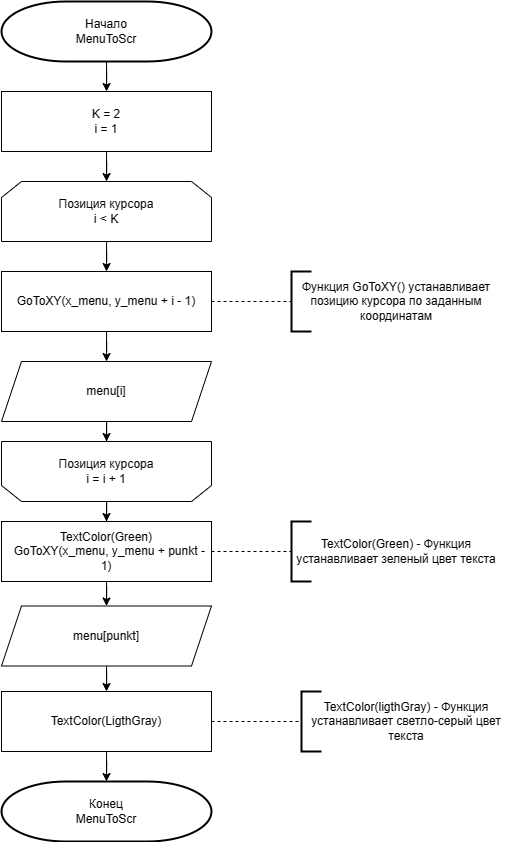


Рисунок 1.4 – Процедура MenuToScr

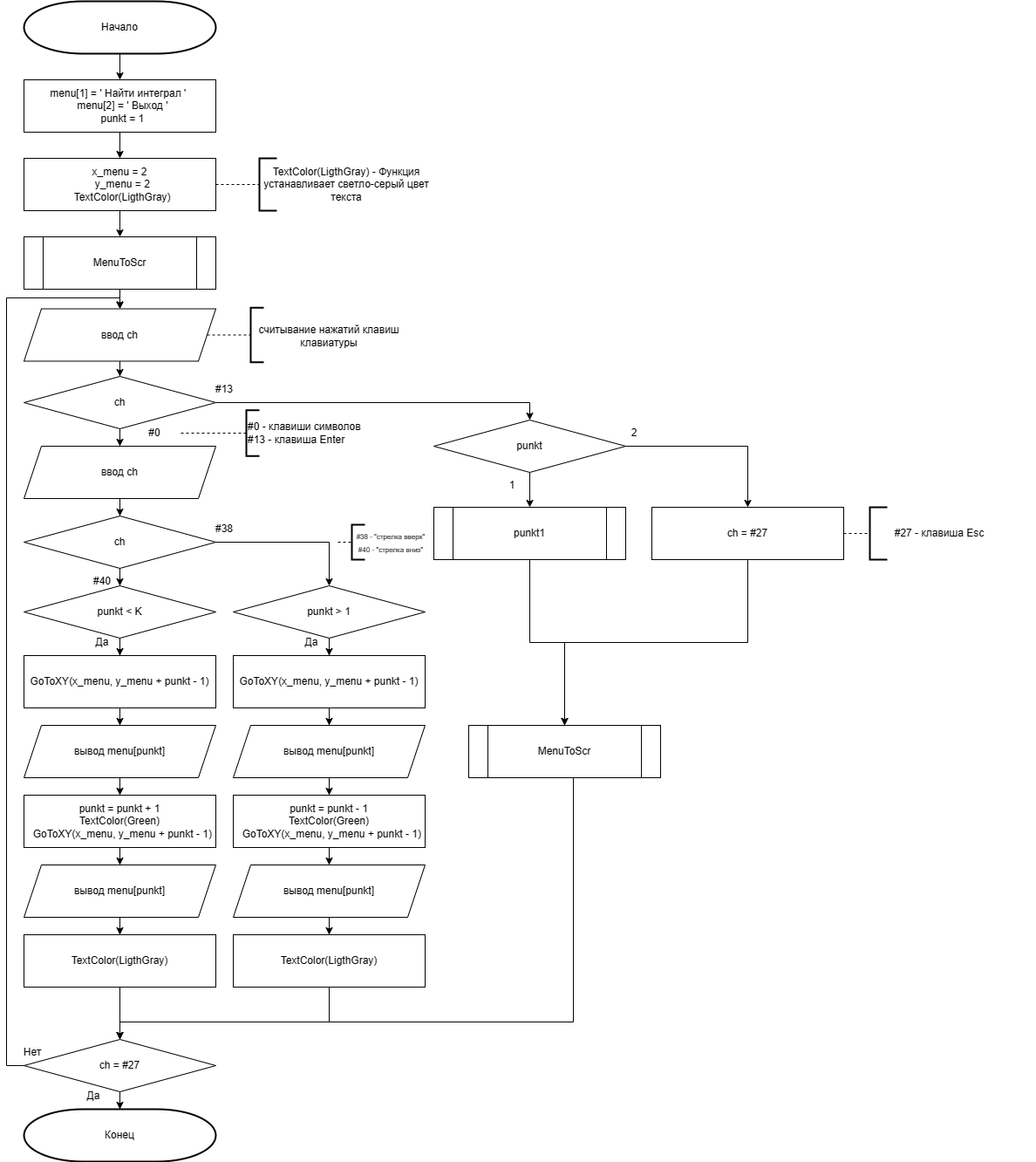


Рисунок 1.5 – Тело программы

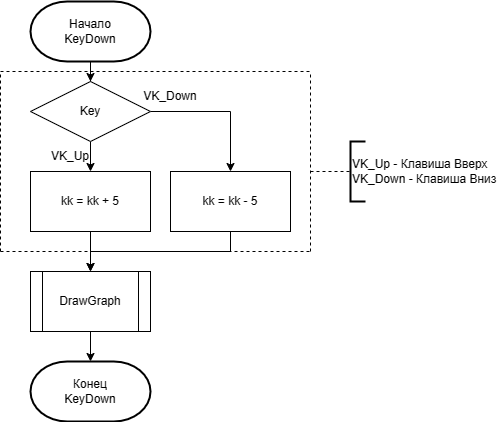
****

Рисунок 1.6 – Процедура KeyDown

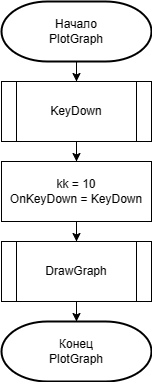


Рисунок 1.7 – Процедура PlotGraph

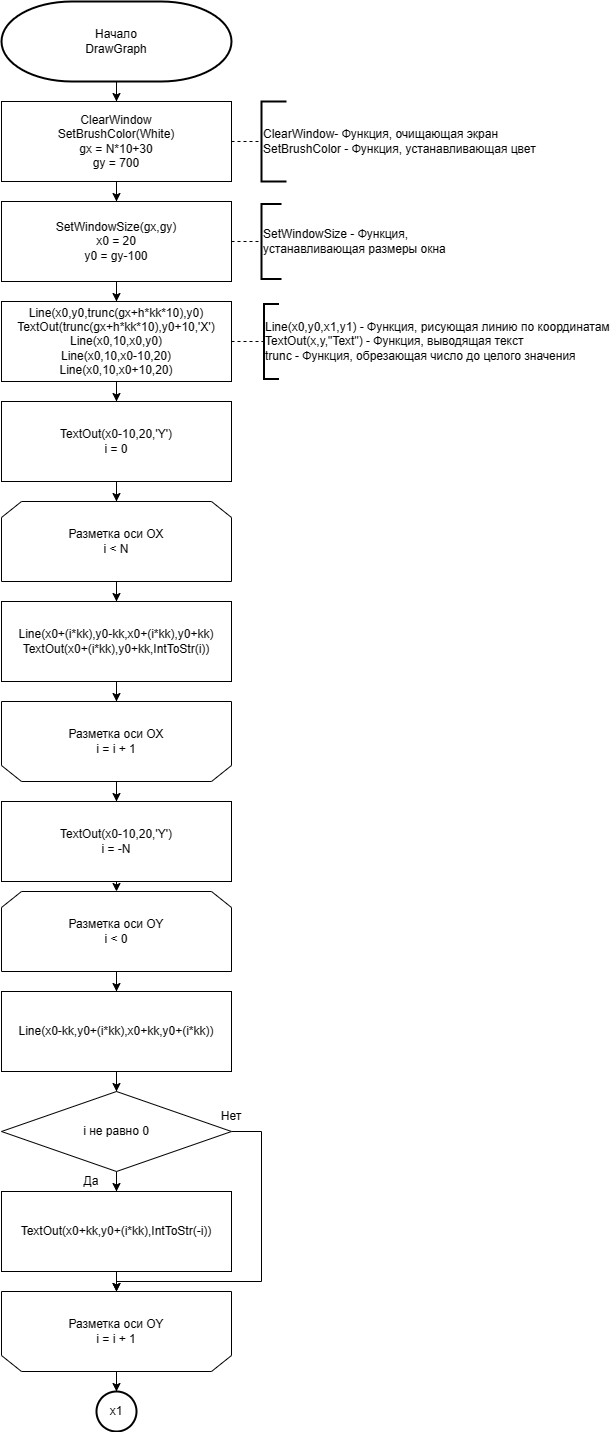


Рисунок 1.8.1 – Процедура DrawGraph

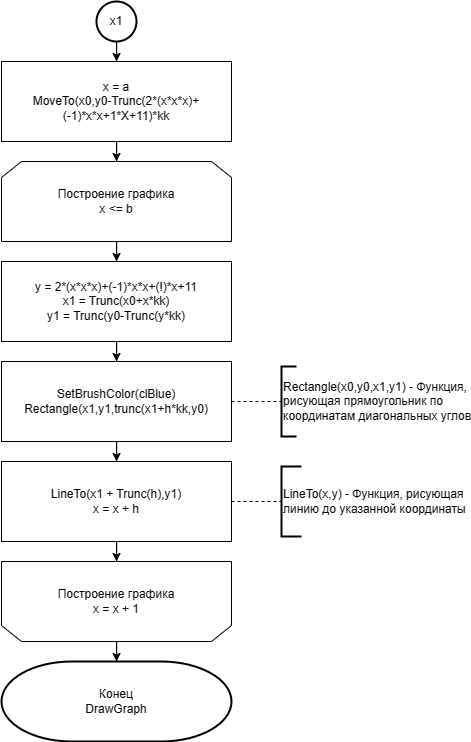
****

Рисунок 1.8.2 – Процедура DrawGraph

**Код программы**

program FindIntegral;

uses CRT,GraphABC;

const

NORM = LightGray;

SEL = Green;

K = 2;

var

menu: array[1..K] of string;

punkt: integer;

ch: char;

x\_menu, y\_menu: integer;

function Calculations(var x, h: real): real;

var

value\_f: real;

begin

value\_f := h \* (2 \* (x \* x \* x) + (-1) \* x \* x + (1) \* x + (11));

Calculations := value\_f;

end;

function Calculations\_NL(var x: real): real;

var

value\_f\_NL: real;

begin

value\_f\_NL := ((x \* x \* x \* x) / 2) - ((x \* x \* x) / 3) + ((x \* x) / 2) + (11 \* x);

Calculations\_NL := value\_f\_NL;

end;

var

x, h, a, b, sum\_f, fun, integral, f\_a, f\_b, error: real;

N,kk: integer;

procedure DrawGraph;

var

gx, gy, x0, y0, x1, y1, i: integer;

x, y: real;

begin

ClearWindow;

SetBrushColor(clWhite);

gx := N \* 10 + 30;

gy := 700;

SetWindowSize(gx, gy);

x0 := 20;

y0 := gy - 100;

Line(x0, y0, trunc(gx+h\*kk\*10), y0);

TextOut(trunc(gx+h\*kk\*10), y0 + 10, 'X');

Line(x0, 10, x0, y0);

Line(x0, 10, x0 - 10, 20);

Line(x0, 10, x0 + 10, 20);

TextOut(x0 - 10, 20, 'Y');

for i := 0 to N do

begin

Line(x0 + (i \* kk), y0 - kk, x0 + (i \* kk), y0 + kk);

TextOut(x0 + (i \* kk), y0 + kk, IntToStr(i));

end;

for i := -N to 0 do

begin

Line(x0 - kk, y0 + (i \* kk), x0 + kk, y0 + (i \* kk));

if i <> 0 then

TextOut(x0 + k, y0 + (i \* kk), IntToStr(-i));

end;

x := a;

MoveTo(x0, y0 - Trunc((2 \* (x \* x \* x) + (-1) \* x \* x + (1) \* x + 11) \* kk));

while x <= b do

begin

y := 2 \* (x \* x \* x) + (-1) \* x \* x + (1) \* x + 11; { функция }

x1 := Trunc(x0 + x \* kk);

y1 := Trunc(y0 - Trunc(y \* kk)); { экранные координаты }

SetBrushColor(clBlue);

Rectangle(x1, y1, trunc(x1+h\*kk), y0);

LineTo(x1 + Trunc(h), y1);

x := x + h;

end;

end;

procedure KeyDown(Key: integer);

begin

case Key of

VK\_Up: kk := kk + 5;

VK\_Down: kk := kk - 5;

end;

DrawGraph;

end;

procedure PlotGraph(a, b, h: real; N: integer);

begin

kk := 10;

OnKeyDown := KeyDown;

DrawGraph;

end;

procedure punkt1;

begin

ClrScr;

var i: integer;

repeat

begin

writeln('Введите пределы интегрирования через пробел:');

readln(a, b);

CLRSCR;

writeln('Введите количество разделений:');

readln(N);

CLRSCR;

h := (b - a) / (N - 1);

x := a;

sum\_f := 0;

f\_a := Calculations\_NL(a);

f\_b := Calculations\_NL(b);

integral := f\_b - f\_a;

for i := 0 to (N - 1) do

begin

fun := Calculations(x, h);

sum\_f := sum\_f + fun;

x := x + h;

end;

sum\_f := abs(sum\_f);

error := sum\_f - integral;

writeln('Результат вычислений:', sum\_f);

writeln('Абсолютная погрешность: ', error);

PlotGraph(a,b,h,N);

writeln('<<Нажмите Enter чтобы выйти в меню>>');

ch := readkey;

end;

until ch = #13;

end;

procedure MenuToScr;

var

i: integer;

begin

ClrScr;

for i := 1 to K do

begin

GoToXY(x\_menu, y\_menu + i - 1);

write(menu[i]);

end;

TextColor(SEL);

GoToXY(x\_menu, y\_menu + punkt - 1);

write(menu[punkt]);

TextColor(NORM);

end;

begin

SetConsoleIO;

menu[1] := ' Найти интеграл ';

menu[2] := ' Выход ';

punkt := 1;

x\_menu := 2;

y\_menu := 2;

TextColor(NORM);

MenuToScr;

repeat

ch := ReadKey;

if ch = #0 then

begin

ch := ReadKey;

case ch of

#40: {Вниз}

if punkt < K then

begin

GoToXY(x\_menu, y\_menu + punkt - 1);

write(menu[punkt]);

punkt := punkt + 1;

TextColor(SEl);

GoToXY(x\_menu, y\_menu + punkt - 1);

write(menu[punkt]);

TextColor(NORM);

end;

#38: {Вверх}

if punkt > 1 then

begin

GoToXY(x\_menu, y\_menu + punkt - 1);

write(menu[punkt]);

punkt := punkt - 1;

TextColor(SEl);

GoToXY(x\_menu, y\_menu + punkt - 1);

write(menu[punkt]);

TextColor(NORM);

end;

end;

end

else

if ch = #13 then

begin

case punkt of

1: punkt1;

2: ch := #27;

end;

MenuToScr;

end;

until ch = #27;

end.

**Результаты выполнения программы**

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 1 10  10 | Результат вычислений:5830  Абсолютная погрешность: 1015 |

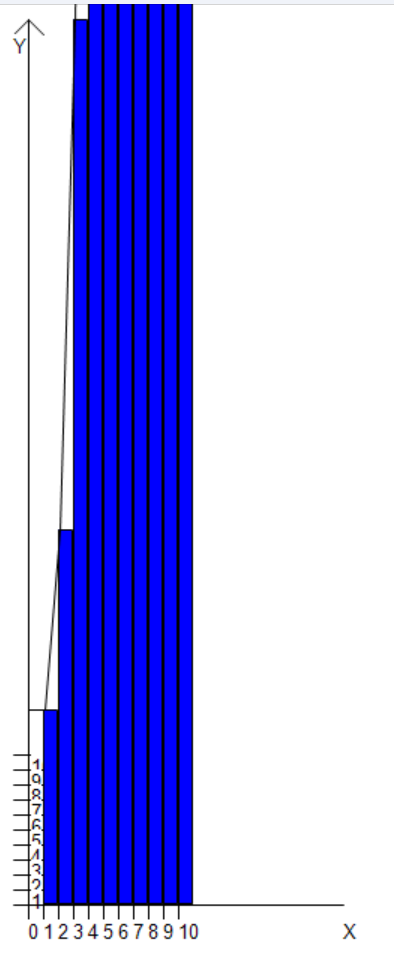
****

Рисунок 2.1 – Визуализация расчёта интеграла

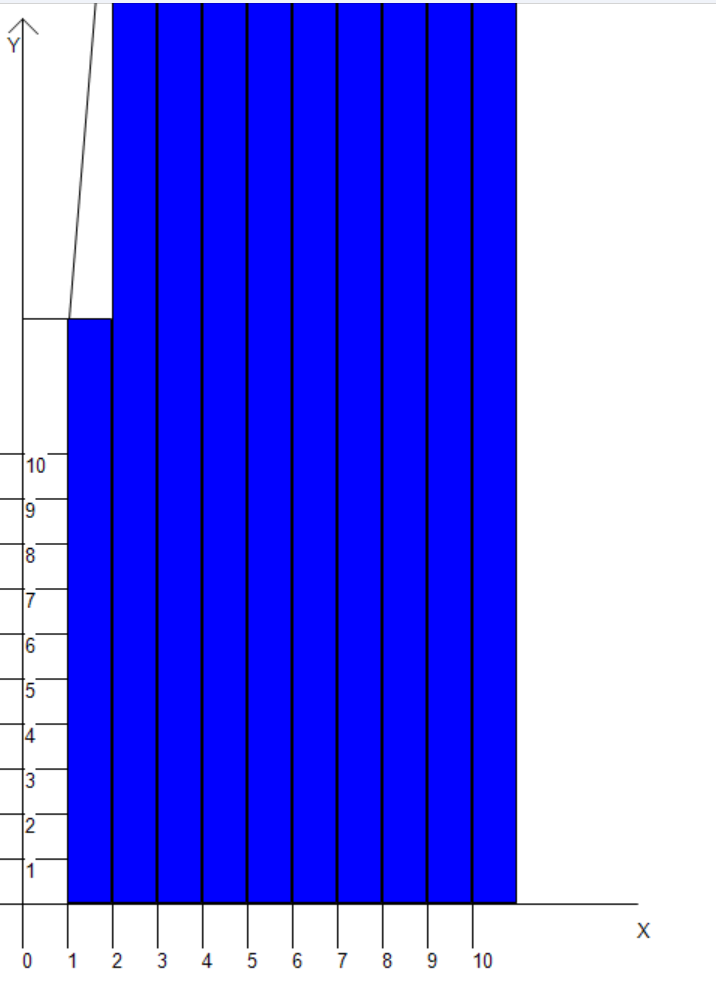
****

Рисунок 2.2 – Масштабирование графика функции

**Вывод**

Программы и алгоритмы, разработанные в ходе работы, представляют собой пример работы с графикой на языке программирования Pascal.

В ходе написания программы для визуализации графика функции и расчёта интеграла были изучены основные принципы работы с графикой в языке программирования Pascal и теория по вычислению интеграла методом «левых прямоугольников».

В итоге, была разработана программа, визуализирующая расчёт интеграла функции методом «левых прямоугольников», с возможностью масштабирования графика. А также пример работы с функциями, процедурами, и подключением библиотеки Crt для разработки минимального пользовательского интерфейса.