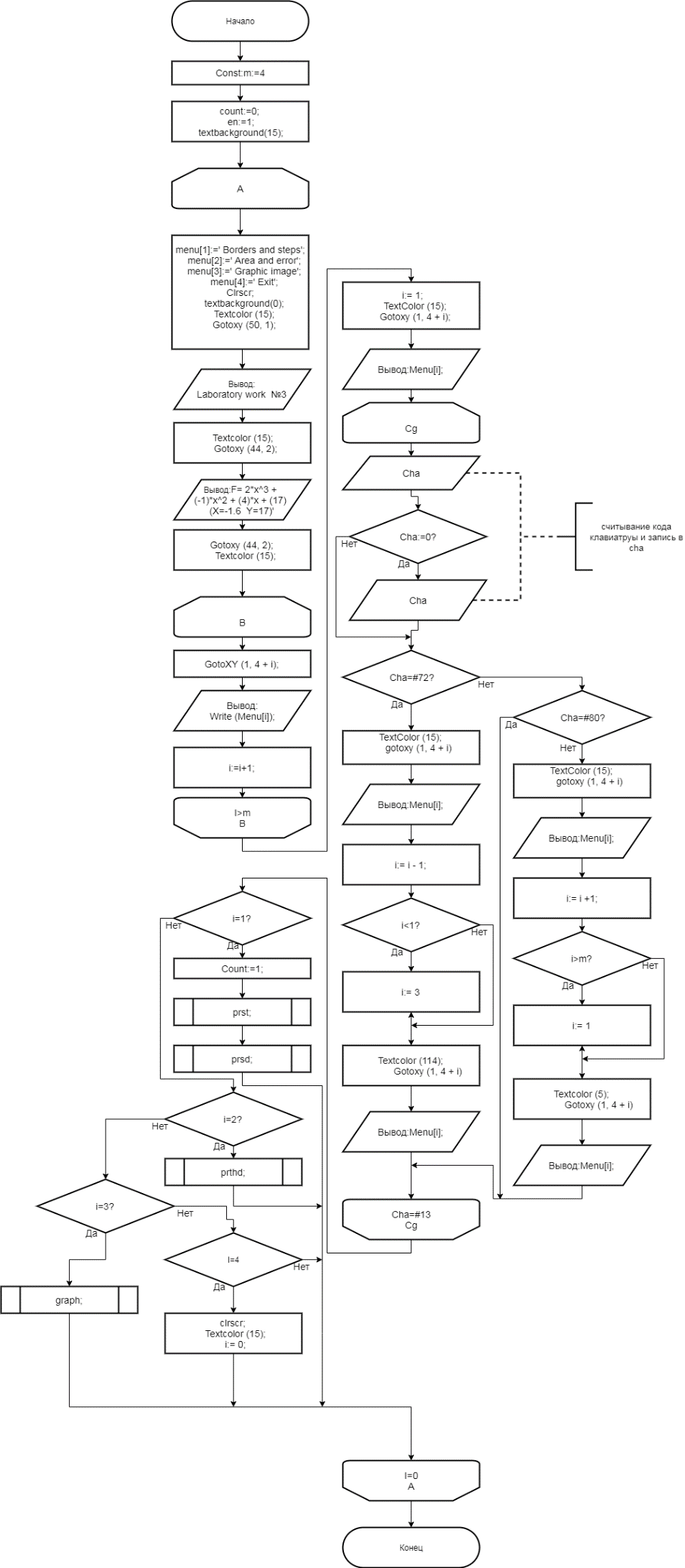
Вариант: 380

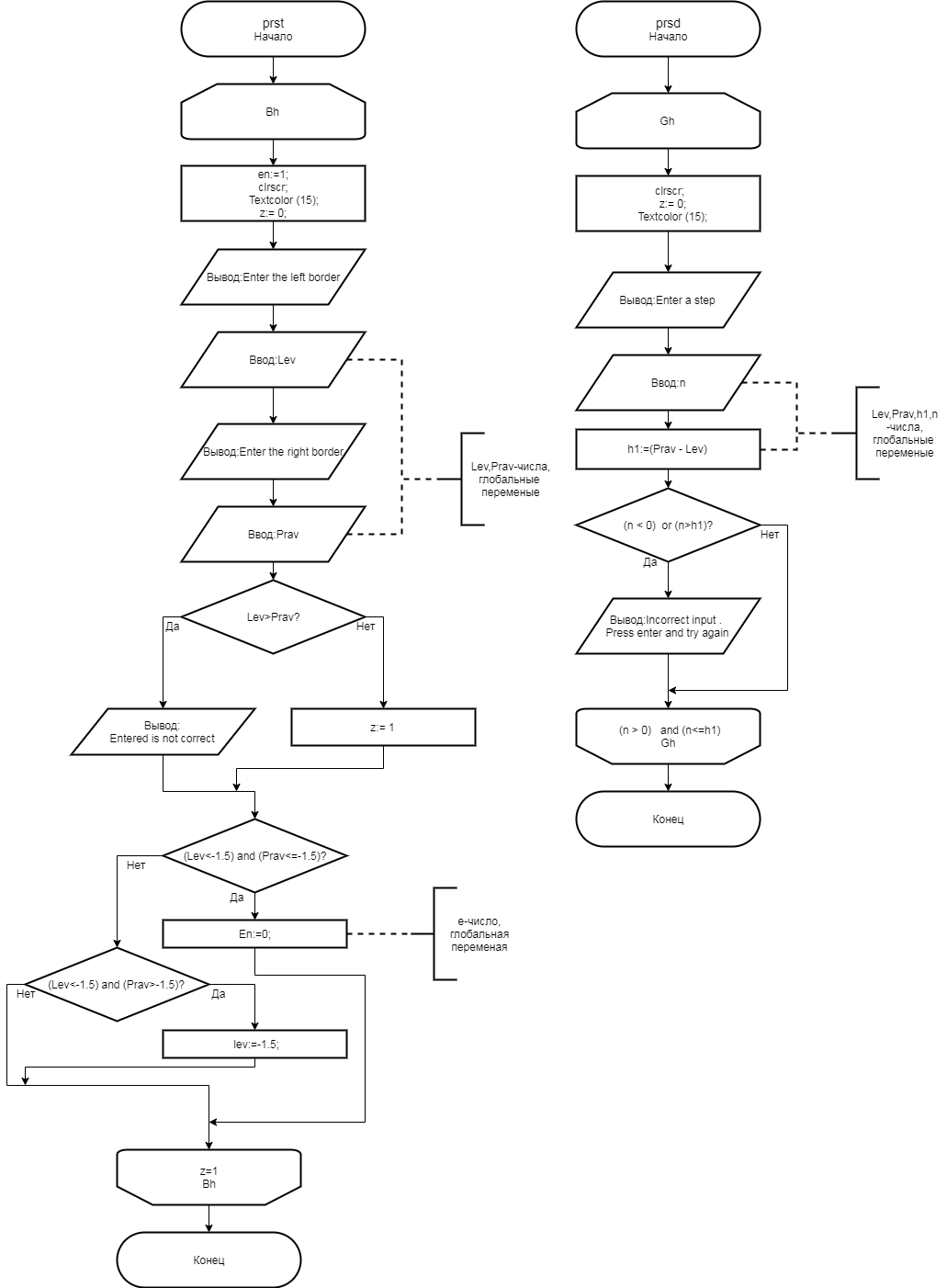
Цель работы: освоить принципы работы в графическом режиме; получить базовые навыки взаимодействия с графическими примитивами.

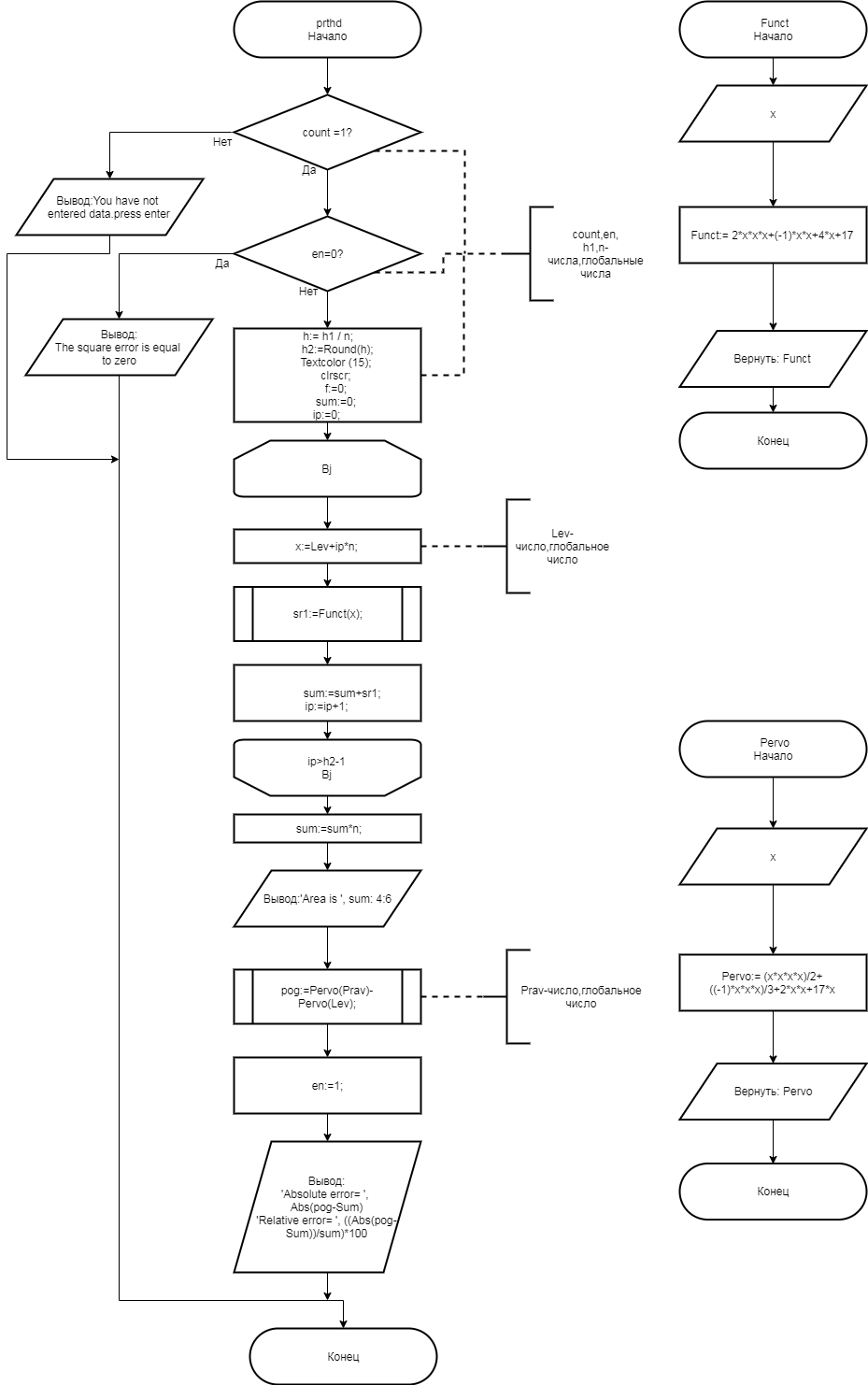
Задание:

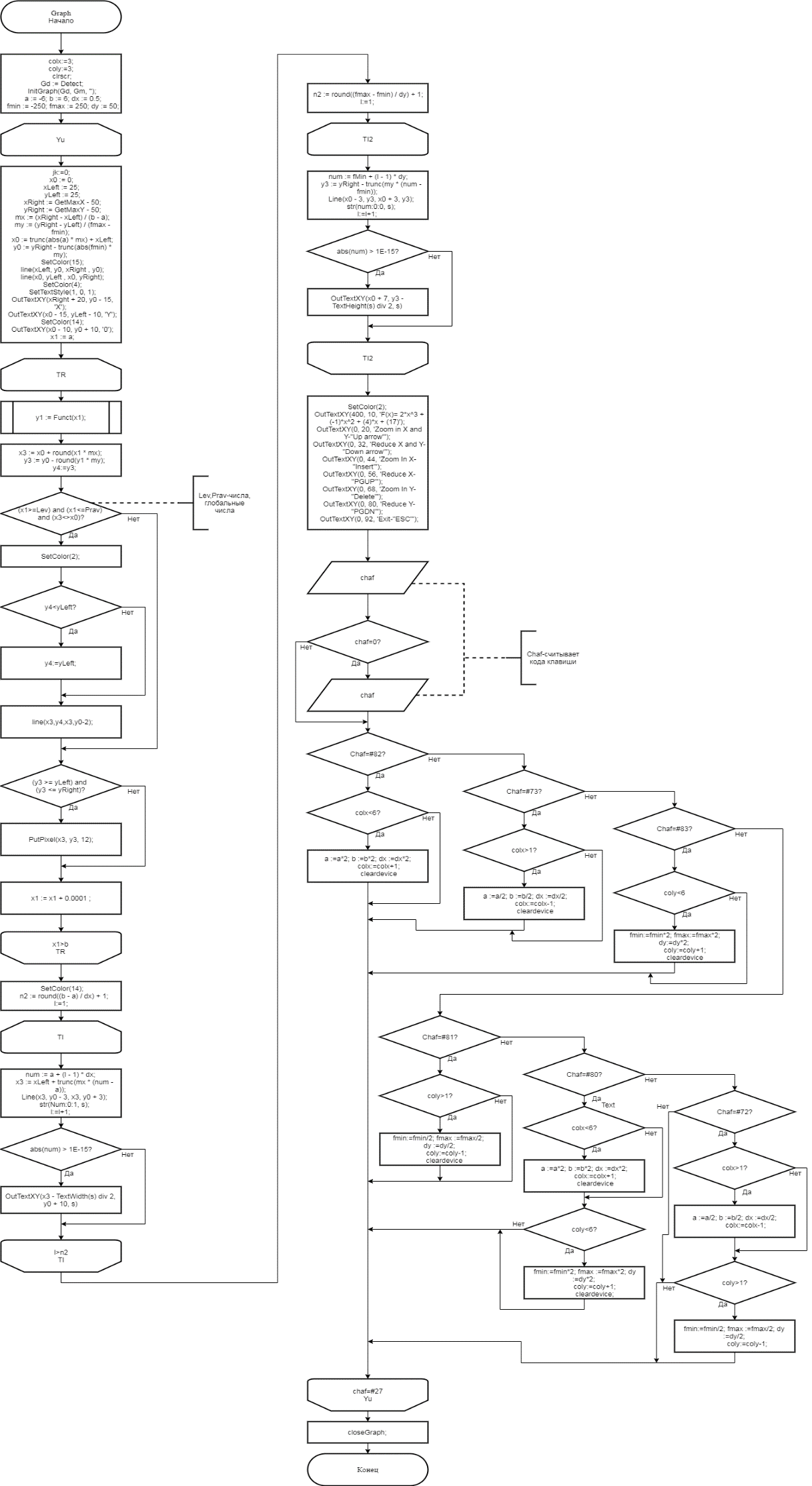
1. Дополнить программу, реализованную в ходе предыдущей лабораторной работы, режимом визуализации.
2. Предусмотреть возможность вывода кривой, ограничивающей фигуру, на координатную плоскость
3. Реализовать следующие возможности и элементы: масштабирование графика, подписи на осях, вывод информации о задание.
4. Реализовать не менее двух возможностей из представленных: независимое масштабирование по осям, штриховка вычисляемой площади, визуализация численного расчета интеграла.

Схема алгоритма:









Исходный код:

**program** laba3;

{$mode delphi}

**uses** wincrt,graph,crt;

**const**

m= 4;

**var** Gd,Gm:smallint;

Lev, Prav,en, h, sum, pog,x,f,h1:real;

z, i,h2,count,chaf:longint;

n:real;

cha:char;

Menu:**array**[1..m]**of** string;

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

**begin**

Funct:= 2\*x\*x\*x+(-1)\*x\*x+4\*x+17

**end**;

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

**begin**

Pervo:= (x\*x\*x\*x)/2+((-1)\*x\*x\*x)/3+2\*x\*x+17\*x

**end**;

**procedure** graph;

**var**

x0, y0, x3, y3, xLeft, yLeft, xRight, yRight, n2,colx,coly,jk,y4: integer;

a, b, fmin, fmax, x1, y1, mx, my, dx, dy, num: real;

l: byte;

s: string;

chaf:char;

**begin**

colx:=3;

coly:=3;

clrscr;

Gd := Detect;

InitGraph(Gd, Gm, '');

a := -6; b := 6; dx := 0.5;

fmin := -250; fmax := 250; dy := 50;

**repeat**

jk:=0;

x0 := 0;

xLeft := 25;

yLeft := 25;

xRight := GetMaxX - 50;

yRight := GetMaxY - 50;

mx := (xRight - xLeft) / (b - a);

my := (yRight - yLeft) / (fmax - fmin);

x0 := trunc(abs(a) \* mx) + xLeft;

y0 := yRight - trunc(abs(fmin) \* my);

line(xLeft, y0, xRight , y0);

line(x0, yLeft , x0, yRight);

SetColor(4);

SetTextStyle(1, 0, 1);

OutTextXY(xRight + 20, y0 - 15, 'X');

OutTextXY(x0 - 15, yLeft - 10, 'Y');

SetColor(14);

OutTextXY(x0 - 10, y0 + 10, '0');

x1 := a;

**while** x1 <= b **do**

**begin**

y1 := Funct(x1);

x3 := x0 + round(x1 \* mx);

y3 := y0 - round(y1 \* my);

y4:=y3;

**if** (y3 >= yLeft) **and** (y3 <= yRight) **then** PutPixel(x3, y3, 12);

**if** (x1>=Lev) **and** (x1<=Prav) **and** (x3<>x0) **then**

**begin**

SetColor(2);

**if** y4<yLeft **then** y4:=yLeft;

line(x3,y4,x3,y0-2);

**end**;

**if** (y3 >= yLeft) **and** (y3 <= yRight) **then** PutPixel(x3, y3, 12);

x1 := x1 + 0.0001 ;

**end**;

SetColor(14);

n2 := round((b - a) / dx) + 1;

**for** l := 1 **to** n2 **do**

**begin**

num := a + (l - 1) \* dx;

x3 := xLeft + trunc(mx \* (num - a));

Line(x3, y0 - 3, x3, y0 + 3);

str(Num:0:1, s);

**if** abs(num) > 1E-15 **then**

OutTextXY(x3 - TextWidth(s) **div** 2, y0 + 10, s)

**end**;

n2 := round((fmax - fmin) / dy) + 1;

**for** l := 1 **to** n2 **do**

**begin**

num := fMin + (l - 1) \* dy;

y3 := yRight - trunc(my \* (num - fmin));

Line(x0 - 3, y3, x0 + 3, y3);

str(num:0:0, s);

**if** abs(num) > 1E-15 **then**

OutTextXY(x0 + 7, y3 - TextHeight(s) **div** 2, s)

**end**;

SetColor(15);

OutTextXY(400, 10, 'F(x)= 2\*x^3 + (-1)\*x^2 + (4)\*x + (17)');

OutTextXY(0, 20, 'Zoom in X and Y-"Up arrow"');

OutTextXY(0, 32, 'Reduce X and Y-"Down arrow"');

OutTextXY(0, 44, 'Zoom In X-"Insert"');

OutTextXY(0, 56, 'Reduce X-"PGUP"');

OutTextXY(0, 68, 'Zoom In Y-"Delete"');

OutTextXY(0, 80, 'Reduce Y-"PGDN"');

OutTextXY(0, 92, 'Exit-"ESC"');

chaf:=wincrt.readkey;

**IF** chaf=#0 **then** chaf:=wincrt.readkey;

**case** chaf **of**

#82: **begin**

**if** (colx<6) **then**

**begin**

a :=a\*2; b :=b\*2; dx :=dx\*2;

colx:=colx+1;

cleardevice

**end**;

**end**;

#73 :**begin**

**if** (colx>1) **then**

**begin**

a :=a/2; b :=b/2; dx :=dx/2;

colx:=colx-1;

cleardevice

**end**;

**end**;

#83 :**begin**

**if** (coly<6) **then**

**begin**

fmin:=fmin\*2; fmax:=fmax\*2; dy:=dy\*2;

coly:=coly+1;

cleardevice

**end**;

**end**;

#81 :**begin**

**if** (coly>1) **then**

**begin**

fmin:=fmin/2; fmax :=fmax/2; dy :=dy/2;

coly:=coly-1;

cleardevice

**end**;

**end**;

#80 :**begin**

**if** (colx<6) **then**

**begin**

a :=a\*2; b :=b\*2; dx :=dx\*2;

colx:=colx+1;

cleardevice

**end**;

**if** (coly<6) **then**

**begin**

fmin:=fmin\*2; fmax :=fmax\*2; dy :=dy\*2;

coly:=coly+1;

cleardevice;

**end**;

**end**;

#72 :**begin**

**if** (colx>1) **then**

**begin**

a :=a/2; b :=b/2; dx :=dx/2;

colx:=colx-1;

**end**;

**if** (coly>1) **then**

**begin**

fmin:=fmin/2; fmax :=fmax/2; dy :=dy/2;

coly:=coly-1;

**end**;

cleardevice;

**end**;

**end**;

**until** chaf=#27;

closeGraph;

**end**;

**procedure** prst;

**begin**

**repeat**

en:=1;

clrscr;

Textcolor (15);

z:= 0;

writeln ('Enter the left border');

readln (Lev);

writeln ('Enter the right border');

readln (Prav);

**if** Lev>Prav **then**

**begin**

writeln ('Entered is not correct');

readln;

**end**

**else**

z:= 1;

**if** (Lev<-1.55) **and** (Prav<=-1.55) **then**

**begin**

en:=0;

**end**

**else**

**if** (Lev<-1.55) **and** (Prav>-1.55)**then**

lev:=-1.55;

**until** z = 1;

**end**;

**procedure** prsd;

**begin**

**repeat**

z:= 0;

Textcolor (15);

writeln ('Enter a step');

readln (n);

h1:=(Prav - Lev);

**if** (n < 0) **or** (n>h1) **then**

**begin**

writeln ('Incorrect input . Press enter and try again');

readln;

**end**;

**until** (n > 0) **and** (n<=h1);

writeln ('Press enter to exit the menu.');

readln;

**end**;

**procedure** prthd;

**var** ip:integer;

sr1:real;

**begin**

**if** count =1 **then**

**begin**

**if** en=0 **then**

**begin**

clrscr;

Writeln('The square error is equal to zero');

**end**

**else**

**begin**

h:= (Prav - Lev) / n;

h2:=Round(h);

Textcolor (15);

clrscr;

f:=0;

sum:=0;

**for** ip:= 0 **to** h2-1 **do**

**begin**

x:=Lev+ip\*n;

sr1:=Funct(x);

sum:=sum+sr1;

**end**;

sum:=sum\*n;

writeln ('Area is ', sum: 0:6);

**begin**

pog:=Pervo(Prav)-Pervo(Lev);

en:=1;

**end**;

Writeln ('Absolute error= ', Abs(pog-Sum): 5:6);

Writeln ('Relative error= ', ((Abs(pog-Sum))/sum)\*100: 3:6,'%');

writeln ('Press enter to exit the menu');

**end**

**end**

**else**

**begin**

clrscr;

Writeln('You have not entered data.press enter.');

**end**;

readln;

**end**;

**begin**

count:=0;

en:=1;

textbackground(0);

**repeat**

menu[1]:=' Borders and steps';

menu[2]:=' Area and error';

menu[3]:=' Graphic image';

menu[4]:=' Exit';

Clrscr;

textbackground(0);

Textcolor (15);

Gotoxy (50, 1);

Writeln ('Laboratory work number 3');

Textcolor (15);

Gotoxy (44, 2);

Writeln ('F= 2\*x^3 + (-1)\*x^2 + (4)\*x + (17)(X=-1.6 Y=17)');

Gotoxy (44, 2);

Textcolor (15);

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

**begin**

GotoXY (1, 4 + i);

Write (Menu[i]);

**end**;

i:= 1;

TextColor (5);

Gotoxy (1, 4 + i);

Write (Menu[i]);

**repeat**

cha:=readkey;

**if** cha = #0 **then**

cha:=readkey;

**case** cha **of**

#72:

**begin**

TextColor (15);

Gotoxy (1, 4 + i);

Write (Menu[i]);

i:= i - 1;

**if** i <1 **then** i:= 4;

Textcolor (14);

Gotoxy (1, 4 + i);

Write (Menu[i]);

**end**;

#80:

**begin**

Textcolor (15);

Gotoxy (1, 4 + i);

Write (Menu[i]);

i:= i + 1;

**if** i >m **then** i:= 1;

Textcolor (5);

gotoxy (1, 4 + i);

Write (Menu[i]);

**end**;

**end**;

**until** cha =#13;

**case** i **of**

1:

**begin**

count:=1;

prst;

prsd;

**end**;

2: prthd;

3: **begin**

graph;

menu[1]:=' Borders and steps';

menu[2]:=' Area and error';

menu[3]:=' Graphic image';

menu[4]:=' Exit';

**end**;

4:

**begin**

clrscr;

Textcolor (15);

writeln ('End of the program...');

i:= 0;

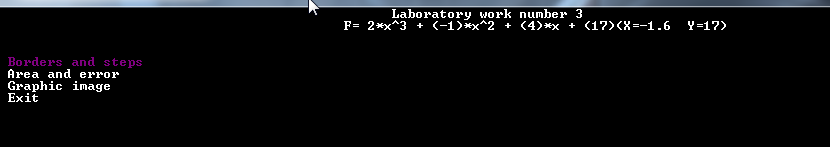
**end**;

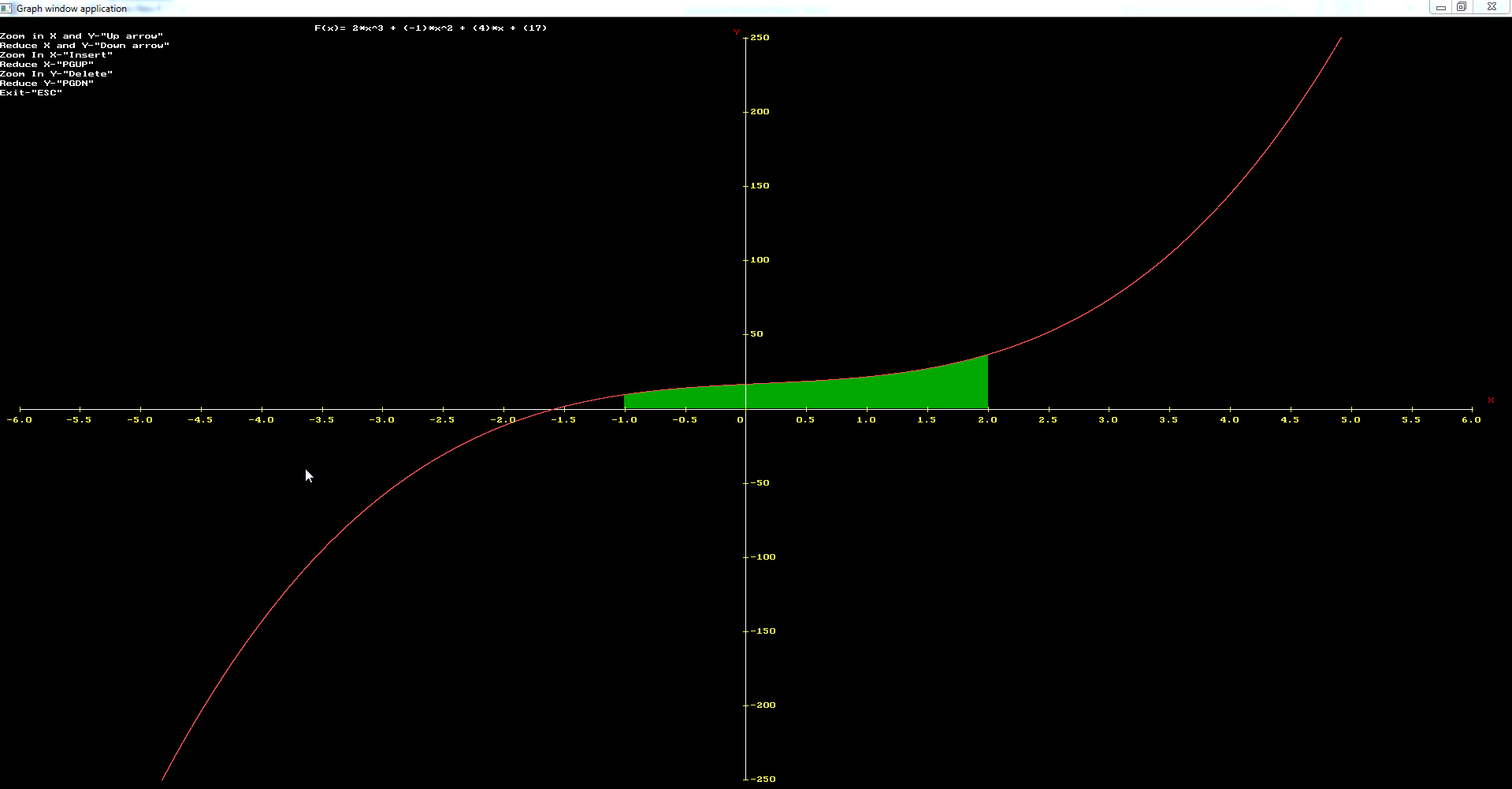
**end**;

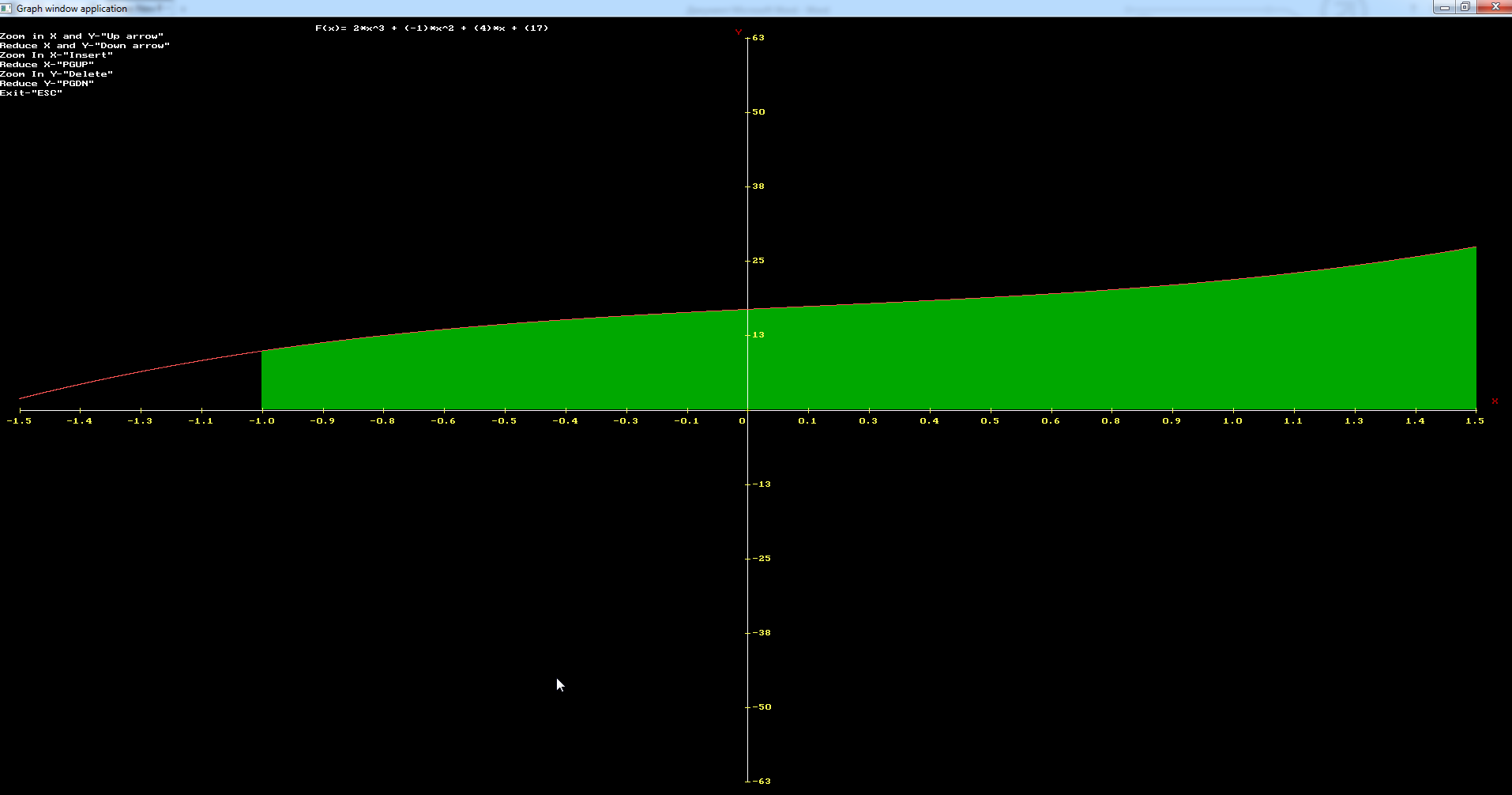
**until** i = 0;

**end**.

Фото отчет выполненой программы:







Вывод: В ходе выполнение лабораторной работы по программированию были получены:

* Навыки работы графическим модулем
* Навыки работы с функциями графического модуля
* Навык проверки вывода функции и изменение масштаба.

При выполнение лабораторной работы были выявлены сложности такие как:

* Работа с графическим модулем.
* Работа с зависимым масштабом и независимым масштабом, увеличение/уменьшение его с помощьюклавиш.
* Выделение площади на графике.