МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

Высшего профессионального образования

**«Вятский государственный университет»**

**(ФГБОУ ВПО «ВятГУ»)**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Работа в графическом режиме.

Отчет по лабораторной работе №4 дисциплины

«Программирование»

Выполнил студент группы ИВТб-11\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Бурдукова А.А./

Проверил преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Чистяков Г.А./

Киров 2018

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

**Задание:**

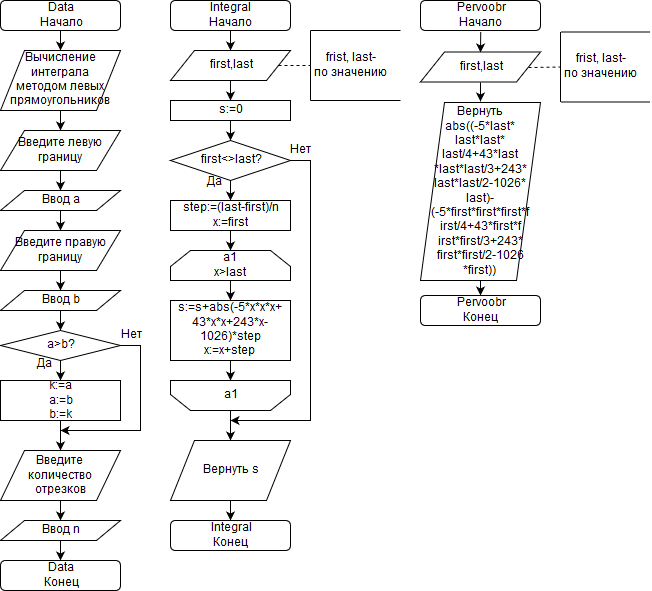
**1. Дополнить программу, реализованную в ходе предыдущей лабораторной работы, режимом визуализации**

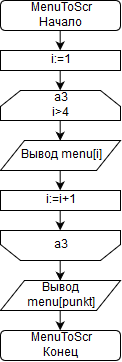
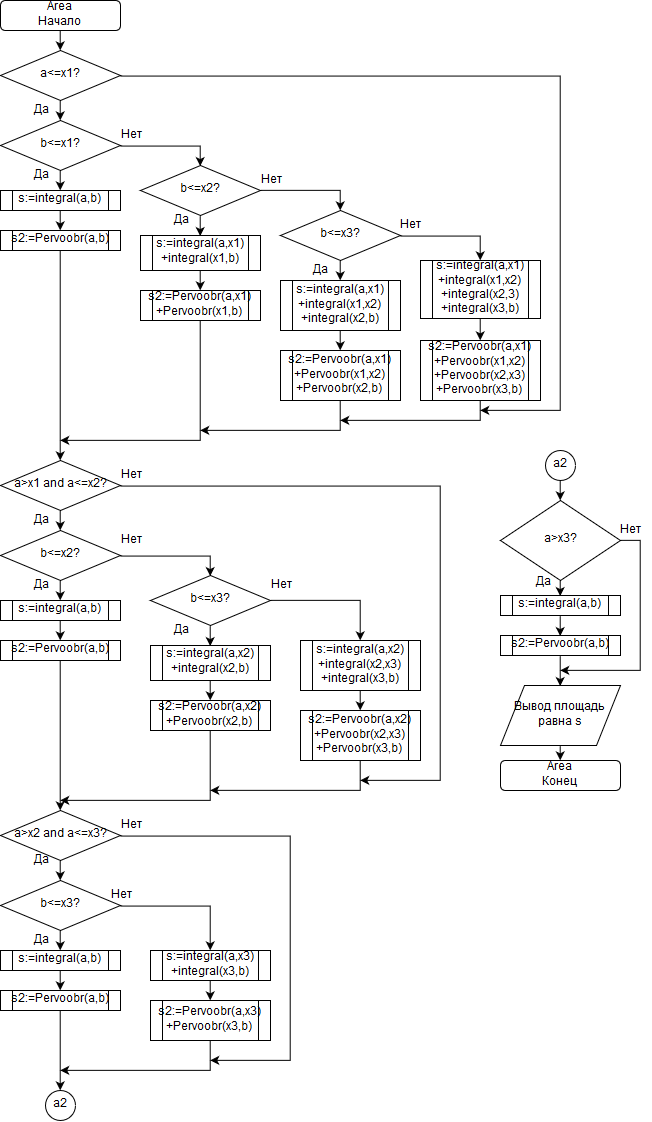
**2. Предусмотреть возможность вывода кривой, ограничивающей фигуру, на координатную плоскость.**

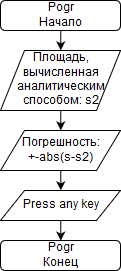
**3. Реализовать следующие возможности и элементы: масштабирование графика, подписи на осях, вывод информации о задании**

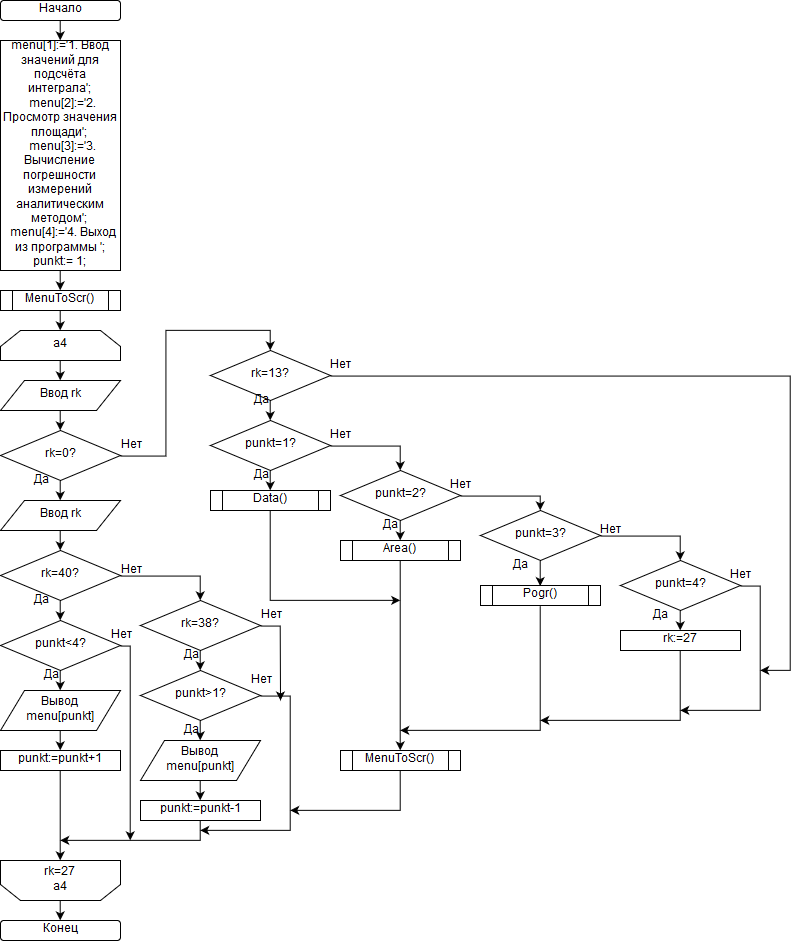
**4. Реализовать не менее двух возможностей из представленных: независимое масштабирование по осям, штриховка вычисляемой площади, визуализация численного расчёта интеграла**

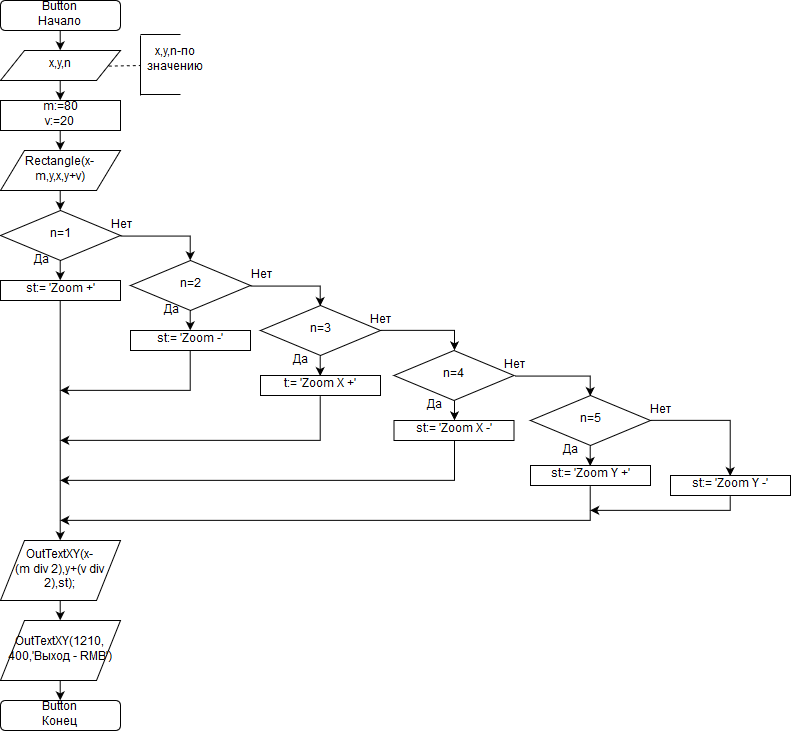
**Схема алгоритмов:**

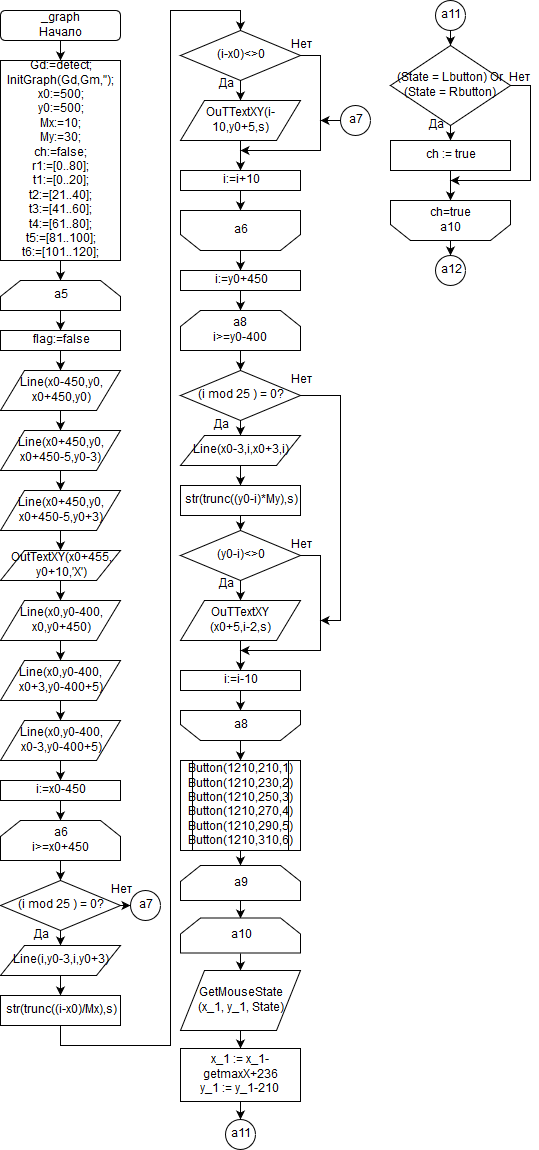
****

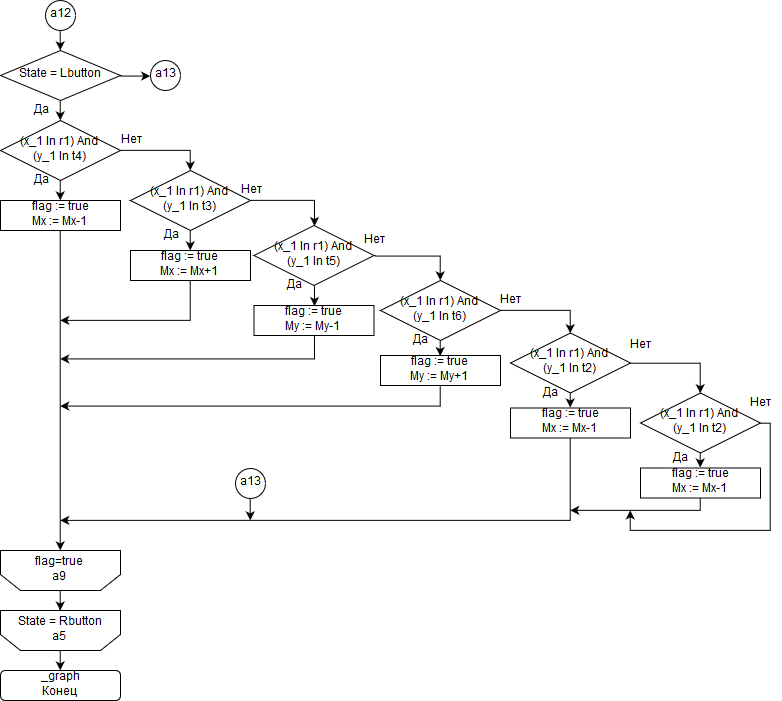
****

****

****

****

****

****

**Исходный текст программы:**

**uses** crt,graph,winmouse;

**const** x1=-6.053; x2=2.879; x3=11.774;

Sel=Green;

NSel=White;

**var** a,b,step,s,s2,k :real;

punkt: byte;

n : longint;

rk : char;

menu : **array**[1..5] **of** string;

buf : string;

err : integer;

**procedure** Data();

**begin**

clrscr;

Textcolor(NSel);

writeln ('Вычисление интеграла методом левых прямоугольников');

writeln();

write('Введите левую границу: ');

readln(buf);

val(buf, a, err);

**while** (err <> 0) **or** (length(buf) > 3) **do**

**begin**

writeln('Левая граница может быть только натуральным числом, не превосходщим 100');

write('Введите левую границу: ');

readln(buf);

val(buf, a, err);

**end**;

write('Введите правую границу: ');

readln(buf);

val(buf, b, err);

**while** (err <> 0) **or** (length(buf) > 3) **do**

**begin**

writeln('Правая граница может быть только натуральным числом, не превосходщим 100');

write('Введите правую границу: ');

readln(buf);

val(buf, b, err);

**end**;

**if** (a > b) **then**

**begin**

k:=a;

a:=b;

b:=k;

**end**;

write('Введите количество отрезков (только натуральное число, меньшее 1 000 000): ');

readln(buf);

val(buf, n, err);

**while** (err <> 0) **do**

**begin**

write('Количество отрезков может быть только натуральным числом, повторите ввод: ');

readln(buf);

val(buf, n, err);

**end**;

**end**;

**function** Integral(first,last:real):real;

**var** x:real;

**begin**

s:=0;

**if** (first <> last) **then**

**begin**

step:=(last-first)/n;

x:=first;

**while** (x<=last) **do**

**begin**

s:=s+abs(-5\*x\*x\*x+43\*x\*x+243\*x-1026)\*step;

x:=x+step;

**end**;

**end**;

Integral:=s;

**end**;

**function** Pervoobr(first,last:real):real;

**begin**

Pervoobr := abs((-5\*last\*last\*last\*last/4+43\*last\*last\*last/3+243\*last\*last/2-1026\*last)-(-5\*first\*first\*first\*first/4+43\*first\*first\*first/3+243\*first\*first/2-1026\*first));

**end**;

**procedure** Area();

**begin**

**if** (a<=x1) **then**

**begin**

**if** (b<=x1) **then**

**begin**

s:=Integral(a,b);

s2:=Pervoobr(a,b);

**end**

**else**

**if** (b<=x2) **then**

**begin**

s:=Integral(a,x1)+Integral(x1,b);

s2:=Pervoobr(a,x1)+Pervoobr(x1,b);

**end**

**else**

**if** (b<=x3) **then**

**begin**

s:=Integral(a,x1)+Integral(x1,x2)+Integral(x2,b);

s2:=Pervoobr(a,x1)+Pervoobr(x1,x2)+Pervoobr(x2,b);

**end**

**else**

**begin**

s:=Integral(a,x1)+Integral(x1,x2)+Integral(x2,x3)+Integral(x3,b);

s2:=Pervoobr(a,x1)+Pervoobr(x1,x2)+Pervoobr(x2,x3)+Pervoobr(x3,b);

**end**;

**end**;

**if** (a>x1) **and** (a<=x2) **then**

**begin**

**if** (b<=x2) **then**

**begin**

s:=Integral(a,b);

s2:=Pervoobr(a,b);

**end**

**else**

**if** (b<=x3) **then**

**begin**

s:=Integral(a,x2)+Integral(x2,b);

s2:=Pervoobr(a,x2)+Pervoobr(x2,b);

**end**

**else**

**begin**

s:=Integral(a,x2)+Integral(x2,x3)+Integral(x3,b);

s2:=Pervoobr(a,x2)+Pervoobr(x2,x3)+Pervoobr(x3,b);

**end**;

**end**;

**if** (a>x2) **and** (a<=x3) **then**

**begin**

**if** (b<=x3) **then**

**begin**

s:=Integral(a,b);

s2:=Pervoobr(a,b);

**end**

**else**

**begin**

s:=Integral(a,x3)+Integral(x3,b);

s2:=Pervoobr(a,x3)+Pervoobr(x3,b);

**end**;

**end**;

**if** (a>x3) **then**

**begin**

s:=Integral(a,b);

s2:=Pervoobr(a,b);

**end**;

clrscr();

writeln('Площадь равна: ',s:0:5);

write('Press any key');

Readkey();

**end**;

**procedure** Pogr();

**begin**

clrscr;

writeln('Площадь, вычисленная аналитическим методом: ',s2);

writeln('Погрешность: +-', abs(s-s2):0:5,' ');

write('Press any key');

rk:=Readkey();

**end**;

**procedure** MenuToScr;

**var** i:integer;

**begin**

clrscr;

**for** i:=1 **to** 5 **do**

**begin**

gotoxy(1,i);

writeln(menu[i]);

**end**;

Textcolor(Sel);

gotoxy(1,punkt);

write(menu[punkt]);

Textcolor(NSel);

**end**;

**function** F(x:integer):longint;

**Begin**

F:=(-1)\*(-5\*x\*x\*x+43\*x\*x+243\*x-1026);

**End**;

**Procedure** Button(x,y:integer;n:byte);

**const**

m= 80;

v=20;

**var**

st:string;

**begin**

SetColor(Red);

Rectangle(x-m,y,x,y+v);

**if** n=1 **then** st:= 'Zoom +'

**Else If** n=2 **then** st:= 'Zoom -'

**Else If** n=3 **then** st:= 'Zoom X +'

**Else if** n=4 **then** st:= 'Zoom X -'

**Else If** n=5 **then** st:= 'Zoom Y +'

**Else If** n=6 **then** st:= 'Zoom Y -';

Setcolor(White);

SetTextJustify(centertext,centertext);

OutTextXY(x-(m **div** 2),y+(v **div** 2),st);

OutTextXY(1210,400,'Выход - RMB');

**end**;

**Procedure** graph;

**var**

Gd,Gm,x0,y0,Mx,My,i,x,a1:integer;

x\_1,y\_1,x\_2,y\_2,x\_3,y\_3,x\_4,y\_4,x\_5,x\_6,state:longint;

s:string;

ch,flag:boolean;

r1,r2,t1,t2,t3,t4,t5,t6:**set of** byte;

**begin**

Gd:=detect;

InitGraph(Gd,Gm,'');

x0:=500;

y0:=500;

Mx:=10;

My:=30;

ch:=false;

r1:=[0..80];

t1:=[0..20];

t2:=[21..40];

t3:=[41..60];

t4:=[61..80];

t5:=[81..100];

t6:=[101..120];

**Repeat**

bar(0,0,getmaxX,getMaxY);

flag:=false;

SetTextStyle(1,0,1);

SetColor(White);

// axis X

Line(x0-450,y0,x0+450,y0);

Line(x0+450,y0,x0+450-5,y0-3);

Line(x0+450,y0,x0+450-5,y0+3);

SetColor(Yellow);

OutTextXY(x0+455,y0+10,'X');

SetColor(White);

// axis Y

Line(x0,y0-400,x0,y0+450);

Line(x0,y0-400,x0+3,y0-400+5);

Line(x0,y0-400,x0-3,y0-400+5);

SetColor(Yellow);

OuTTextXY(x0+10,y0-415,'Y');

SetColor(White);

// X

i:=x0-450;

**while** i<x0+450 **do**

**begin**

**if** (i **mod** 25 ) = 0 **then**

**begin**

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

str(trunc((i-x0)/Mx),s);

**if** (i-x0)<>0 **then**

OuTTextXY(i-10,y0+5,s);

**end**;

i:=i+10;

**end**;

// Y

i:=y0+450;

**while** i>y0-400 **do**

**begin**

**if** (i **mod** 25) = 0 **then**

**begin**

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

str(trunc((y0-i)\*My),s);

**if** (y0-i) <>0 **then**

OuTTextXY (x0+5,i-2,s);

**end**;

i:=i-10;

**end**;

// 0

SetColor(White);

OutTextXY(x0-10,y0+5,'0');

SetColor(Red);

// Func(graph)

**for** x:=-53 **to** 15 **do**

**begin**

x\_2:=(x0+(x-1)\*Mx);

x\_3:=(x0+x\*Mx);

y\_2:=(y0+(F(x-1)) **div** My);

y\_3:=(y0+(F(x) **div** My));

Line(x\_2,y\_2,x\_3,y\_3);

**end**;

// Informationя

Rectangle(800,50,1200,150);

Setcolor(White);

OutTextXY(950,75,'Метод левых прямоугольников');

OutTextXY(950,100,'F:=-5\*x^3+43\*x^2+274\*x-1026');

OutTextXY(950,120,'Корни x1=-6.053; x2=2.879; x3=11.774');

// Strokes

**if** (a <> 0) **and** (b <> 0) **then**

**begin**

Setcolor(yellow);

a1:=round(a);

**while** a1<=b **do**

**begin**

x\_4:=(x0+trunc(a1)\*Mx);

y\_4:=(y0+F(trunc(a1)) **div** My);

Line((x0+trunc(a1)\*Mx),y0,x\_4,y\_4);

a1:=a1+10;

**end**;

**end**;

// Borders

SetColor(green);

**if** (a <> 0) **and** (b <> 0) **then**

**begin**

x\_5:=(x0+trunc(a)\*Mx);

x\_6:=(x0+trunc(b)\*Mx);

Line(x\_5,50,x\_5,1000);

Line(x\_6,50,x\_6,1000);

**end**;

// Buttons

SetColor(white);

Button(1210,210,1);

Button(1210,230,2);

Button(1210,250,3);

Button(1210,270,4);

Button(1210,290,5);

Button(1210,310,6);

OutTextXY(1210,400,'Выход - RMB');

//Mouse

**Repeat**

**Repeat**

GetMouseState(x\_1, y\_1, State);

x\_1 := x\_1-getmaxX+236;

y\_1 := y\_1-210;

**If** (State = Lbutton) **Or** (State = Rbutton) **Then** ch := true;

**Until** ch = true;

**If** q **Then**

**Begin**

**If** (x\_1 **In** r1) **And** (y\_1 **In** t4) **Then**

**Begin**

flag := true;

Mx := Mx-1;

**If** Mx<5 **Then** Mx := Mx+1;

**End**

**Else If** (x\_1 **In** r1) **And** (y\_1 **In** t3) **Then**

**Begin**

flag := true;

Mx := Mx+1;

**If** Mx>35 **Then** Mx := Mx-1;

**End**

**Else If** (x\_1 **In** r1) **And** (y\_1 **In** t5) **Then**

**Begin**

flag := true;

My := My-1;

**If** My<5 **Then** My := My+1;

**End**

**Else If** (x\_1 **In** r1) **And** (y\_1 **In** t6) **Then**

**Begin**

flag := true;

My := My+1;

**If** My>35 **Then** My := My-1;

**End**

**Else If** (x\_1 **In** r1) **And** (y\_1 **In** t2) **Then**

**Begin**

flag := true;

Mx := Mx-1;

My := My+1;

**If** Mx<5 **Then** Mx := Mx+1;

**If** My>35 **Then** My := My-1;

**End**

**Else If** (x\_1 **In** r1) **And** (y\_1 **In** t1) **Then**

**Begin**

flag := true;

Mx := Mx+1;

My := My-1;

**If** My<5 **Then** My := My+1;

**If** Mx>35 **Then** Mx := Mx-1;

**End**

**End**

**Else If** state = Rbutton **Then break**;

**Until** flag = true;

**Until** State = Rbutton;

delay(100);

closegraph;

MenuToScr();

**end**;

**begin**

clrscr;

Textcolor(NSel);

writeln ('Функция для подсчета площади методом левых прямоугольников:');

writeln('F:=-5\*x^3+43\*x^2+274\*x-1026');

writeln('Корни данного уравнения: ',x1:0:4,' ',x2:0:4,' ',x3:0:4);

writeln ('Press any key');

Readkey();

Textcolor(NSel);

menu[1]:='1. Ввод значений для подсчёта интеграла';

menu[2]:='2. Просмотр значения площади';

menu[3]:='3. Вычисление погрешности измерений аналитическим методом';

menu[4]:='4. График функции';

menu[5]:='5. Выход из программы';

Readkey();

punkt:= 1;

MenuToScr();

**repeat**

rk := Readkey();

**if** (rk = #0) **then**

**begin**

rk := Readkey;

**case** rk **of**

#80:

**if** (punkt < 5) **then**

**begin**

gotoxy(1,punkt);

write(menu[punkt]);

inc(punkt);

Textcolor(Sel);

gotoxy(1,punkt);

write(menu[punkt]);

Textcolor(NSel);

**end**;

#72:

**if** (punkt > 1) **then**

**begin**

gotoxy(1,punkt);

write(menu[punkt]);

dec(punkt);

Textcolor(Sel);

gotoxy(1,punkt);

write(menu[punkt]);

Textcolor(NSel);

**end**;

**end**;

**end**

**else**

**if** (rk = #13) **then**

**begin**

**case** punkt **of**

1:

Data();

2:

Area();

3:

Pogr();

4:

graph();

5:

rk:=#27;

**end**;

MenuToScr();

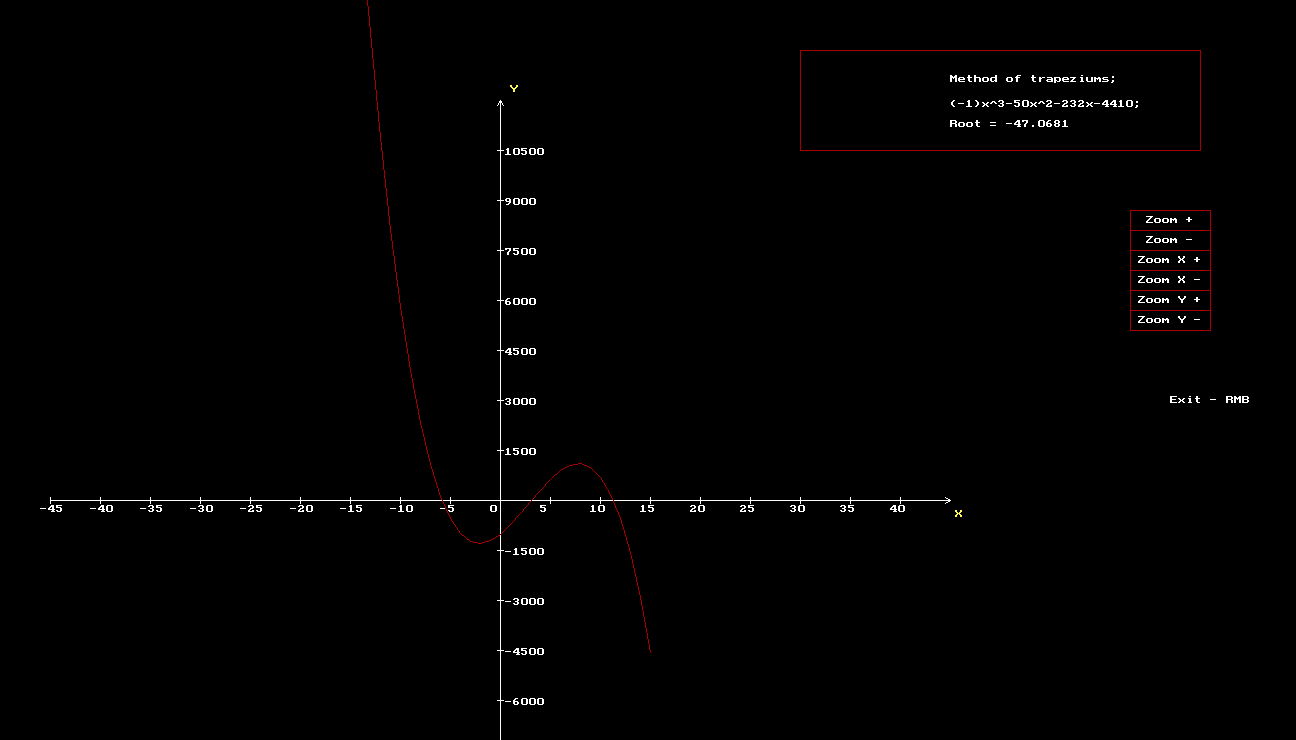
**end**

**until** rk=#27;

clrscr();

**end**.

**Скриншоты выполнения программы**

****

**Вывод**

В ходе данной лабораторной работы были получены знания для создания собственного графического окна. Также появились базовые навыки для взаимодействия с данным окном посредством графических примитивов. Например, процедура ClearWindow для очищения графического окна белым цветом, процедура setfontcolor для изменения цвета шрифта текста, процедура TextOut для вывода данном строки в данный прямоугольник с заданными координатами верхнего левого угла. Нельзя не отметить принцип работы с нажатиями клавиш мыши посредством готовых решений модуля winmouse. Данный метод позволяет считывать координаты экрана, в которых была нажата какая-либо клавиша мыши. Также были получены знания для вывода графика заданной функции. В дальнейшем к ней применялись такие методы, как масштабирование, масштабирование по отдельным осям Oy и Ox.

Лабораторная работа выполнялась в бесплатной среде разработки FreePascal. Данная среда предоставляет много возможностей, в числе которых есть отладчик, который применялся в процессе написания программы для её отладки. Таким образом, был получен навык работы с отладчиком и поиск ошибок в программе.