МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«**Вятский государственный университет**»

**(ФГБОУ ВО «ВятГУ»)**

Факультет автоматики и вычислительной техники

Кафедра ЭВМ

Отчёт

### Лабораторная работа № 4 по дисциплине

«Программирование»

«Работа в графическом режиме»

Выполнил студент группы ИВТб-1301 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Жеребцов К. А.

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

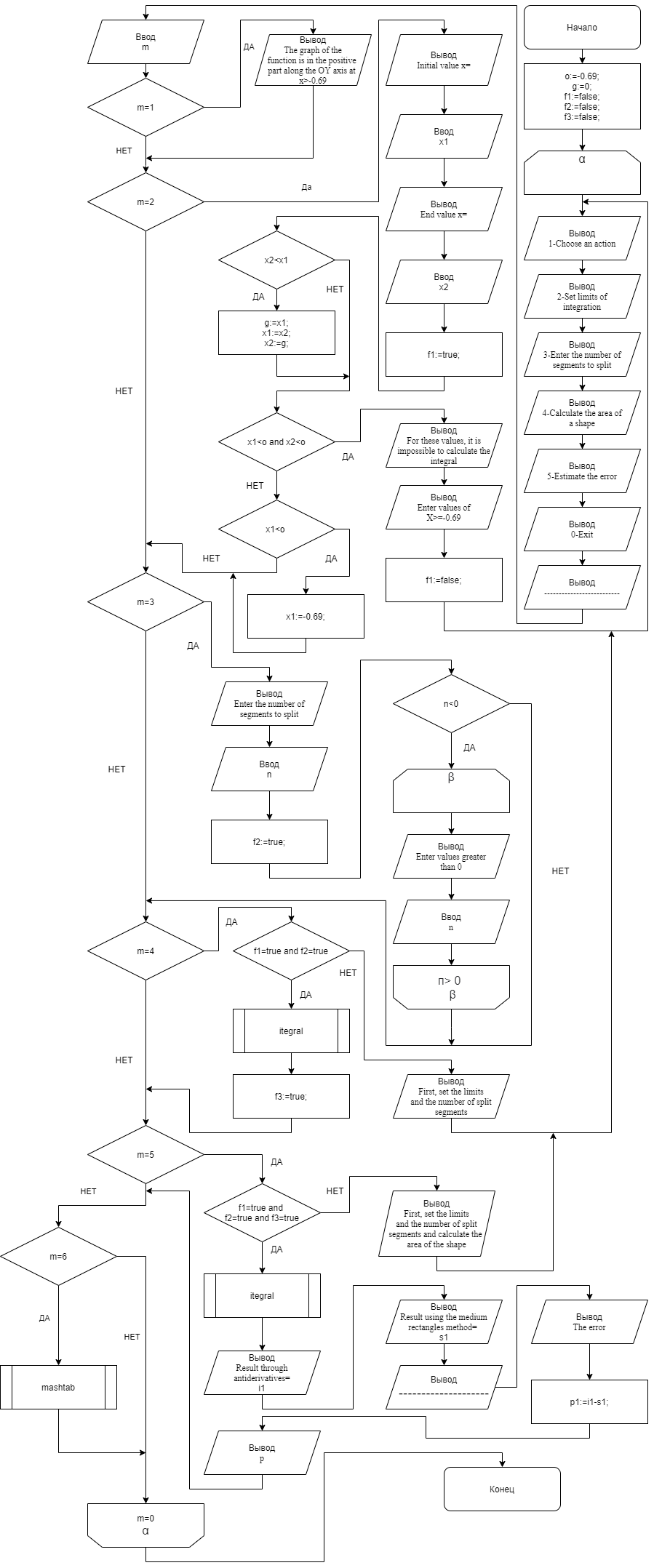
Киров 2020

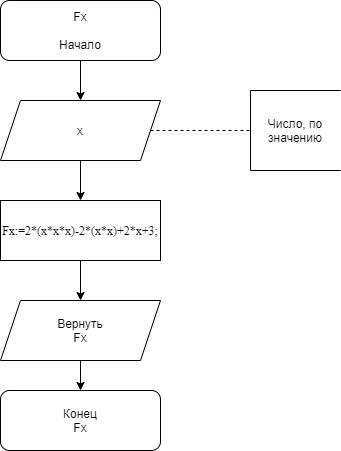
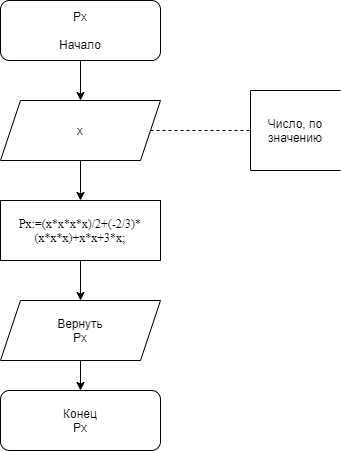
***Цель работы:*** освоить принципы работы в графическом режиме, получить базовые навыки взаимодействия с графическими примитивами.

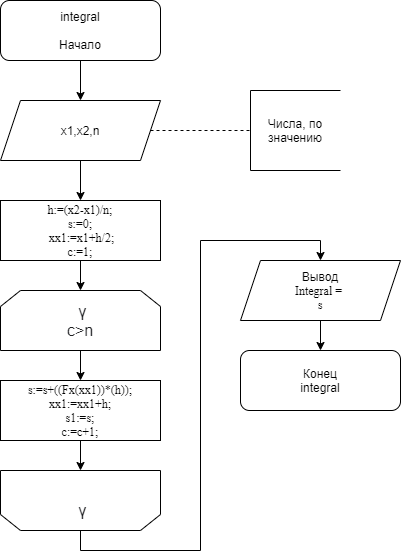
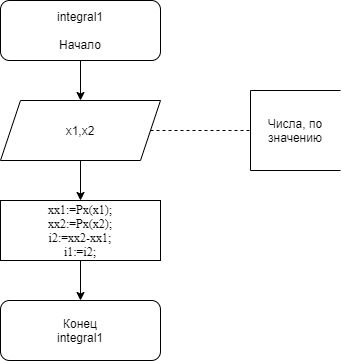
***Задания:***

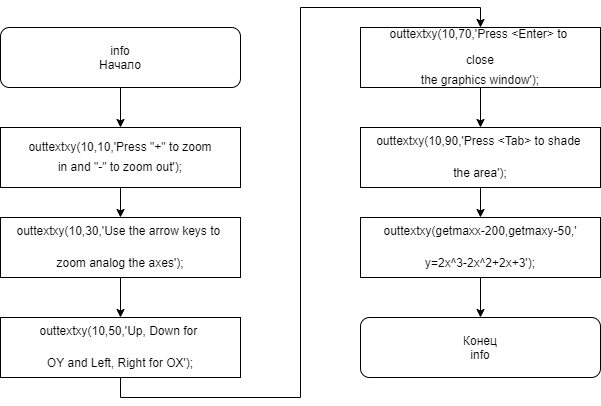
1. Дополнить программу, реализованную в ходе предыдущей лабораторной работы, режимом визуализации.
2. Предусмотреть возможность вывода кривой, ограничивающей фигуру, на координатную плоскость.
3. Реализовать следующие возможности и элементы: масштабирование графика, подписи на осях, вывод информации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

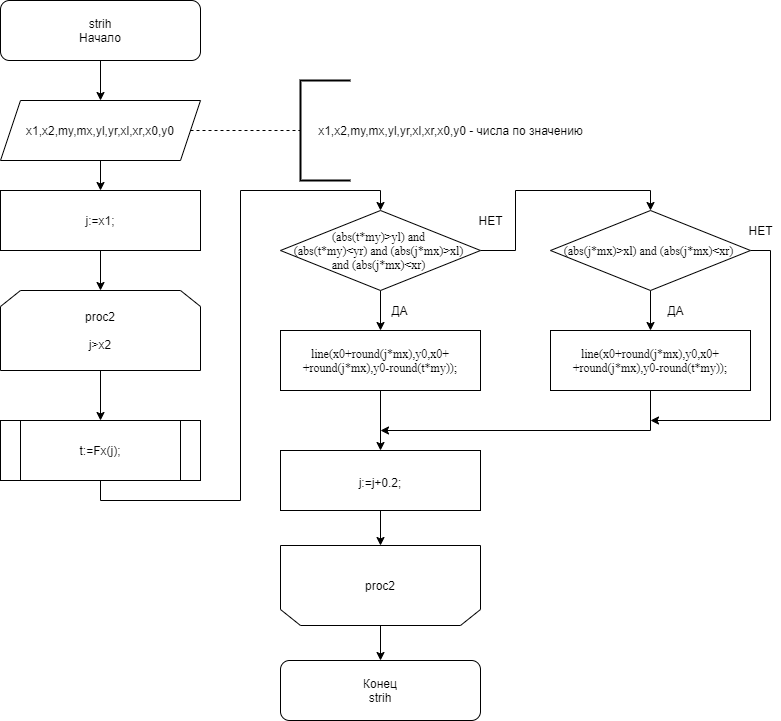
***Схема алгоритма:***

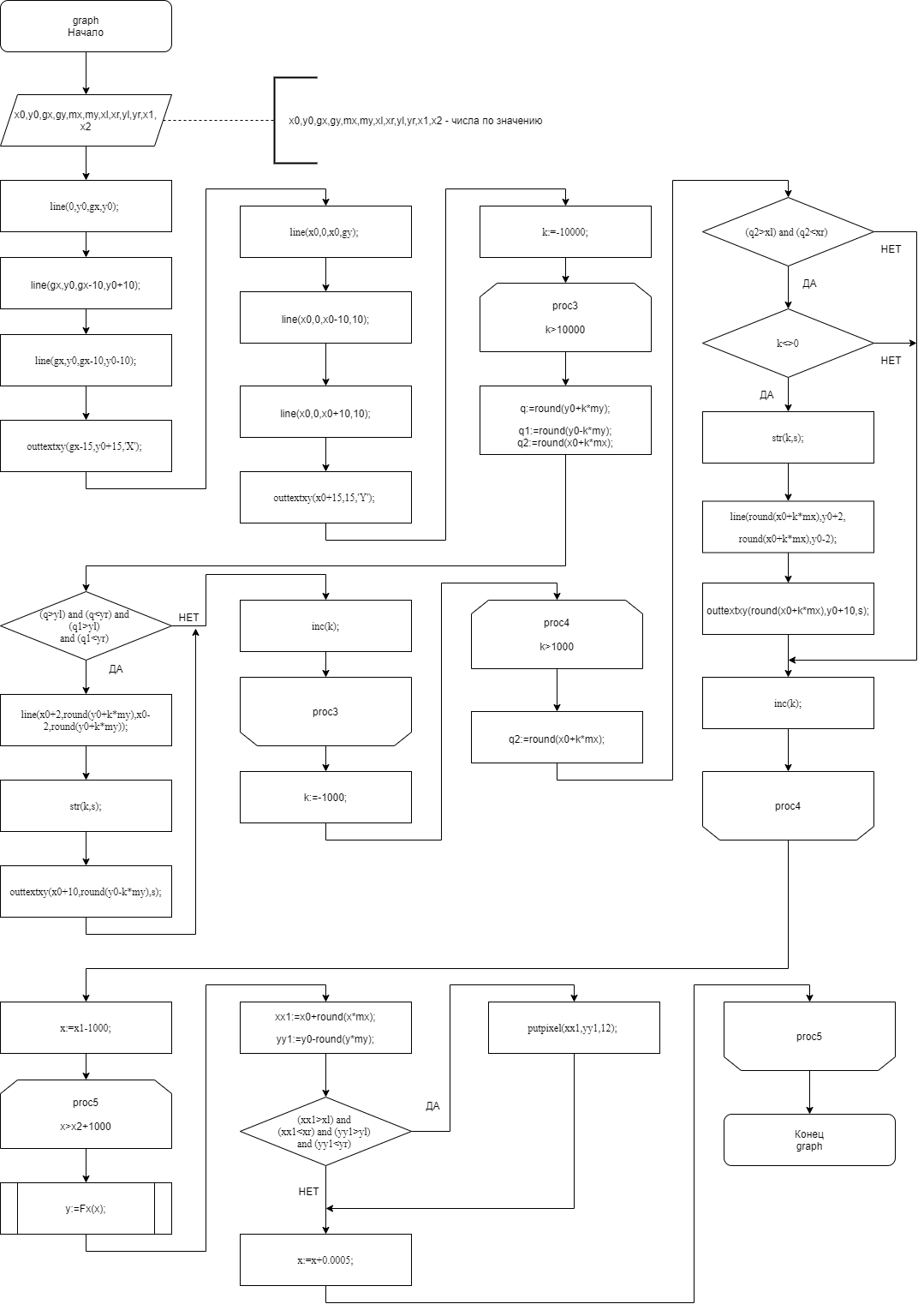


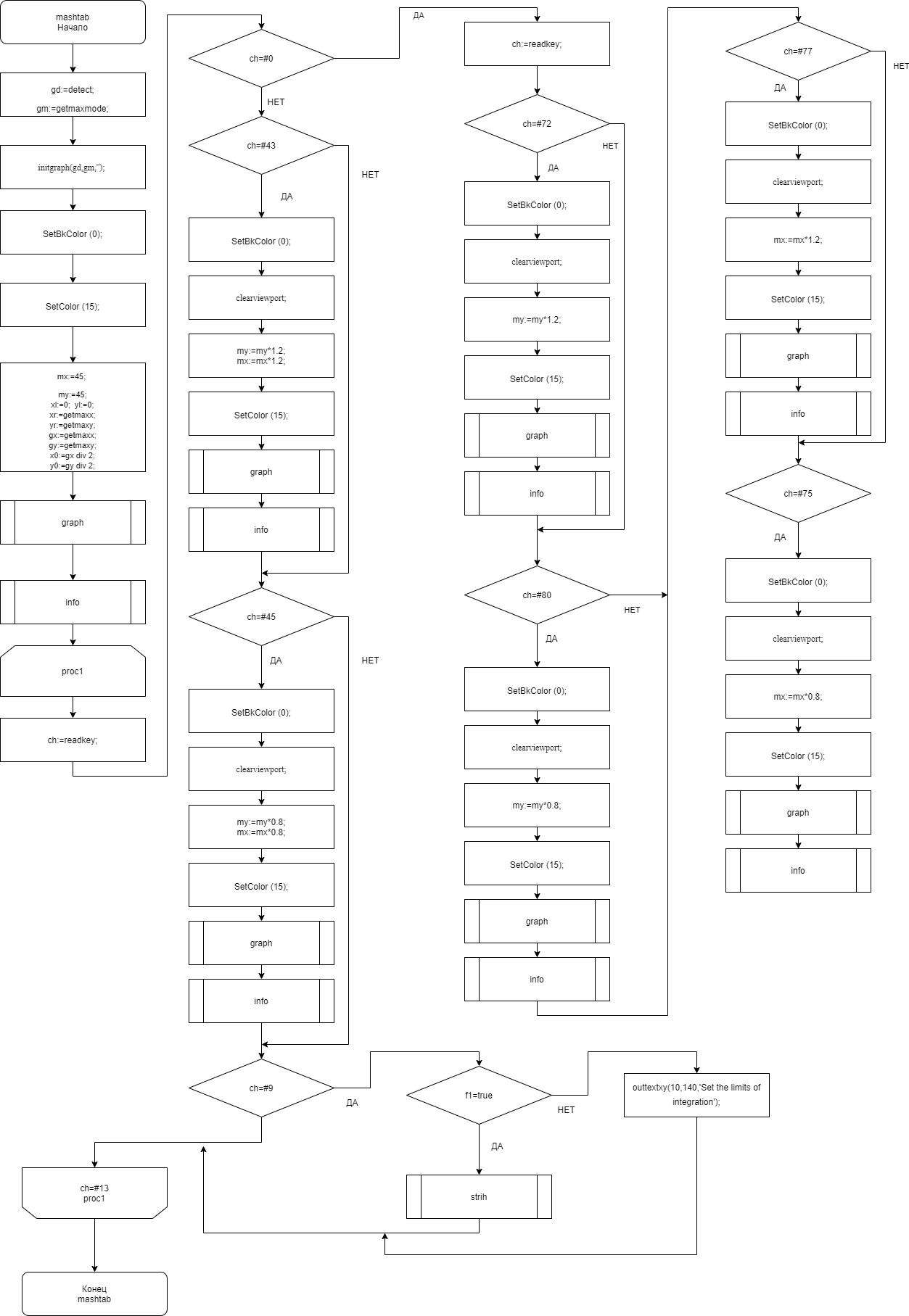
 









***Листинг:***

**program** lab4; // Название программы

**uses**

Crt, graph, wincrt // Подключение консоли

**var** // Объявление переменных

m,xl,yl,xr,yr,n,gx,gy,x0,y0:int64;

x1,x2,g,o,p,i1,s1,h,i,mx,my:extended;

f1,f2,f3:boolean;

gm,gd:integer;

**function** Fx(x:extended):extended; // Объявление исходной функции

**begin**

Fx:=2\*(x\*x\*x)-2\*(x\*x)+2\*x+3;

**end**;

**function** Px(x:extended):extended; // Объявление первообразной исходной функции

**begin**

Px:=(x\*x\*x\*x)/2+(-2/3)\*(x\*x\*x)+x\*x+3\*x;

**end**;

**procedure** integral; // Процедура, вычисляющая значение интеграла через средние прямоугольники

**var** // Объявление локальных переменных

xx1,s: extended;

c:integer;

**begin**

h:=(x2-x1)/n;

s:=0;

xx1:=x1+h/2;

**for** c:=1 **to** n **do**

**begin**

s:=s+((Fx(xx1))\*(h));

xx1:=xx1+h;

s1:=s;

**end**;

writeln('Integral = ',s:6:4); // Вывод полученного значения

**end**;

**procedure** integral1; // Процедура, вычисляющая интеграл через первообразные

**var**

xx1,xx2,i2:extended;

**begin**

xx1:=Px(x1);

xx2:=Px(x2);

i2:=xx2-xx1;

i1:=i2;

**end**;

procedure graph; // Процедура вывода на экран осей и графика функции

var

xx1,yy1,k,q,q1,q2:int64;

x,y:extended;

s:string;

begin

line(0,y0,gx,y0); // Процедура отрисовки линии по 2 точкам, отрисовка оси Х

line(gx,y0,gx-10,y0+10);

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

outtextxy(gx-15,y0+15,'X'); // Процедура вывода подписи оси Х

line(x0,0,x0,gy); // Отрисовка оси Y

line(x0,0,x0-10,10);

line(x0,0,x0+10,10);

outtextxy(x0+15,15,'Y'); // Подпись оси Y

k:=-10000;

while k<=10000 do // Цикл для отрисовки делений на оси Y

begin

q:=round(y0+k\*my);

q1:=round(y0-k\*my);

q2:=round(x0+k\*mx);

if (q>yl) and (q<yr) and (q1>yl) and (q1<yr) then

begin

line(x0+2,round(y0+k\*my),x0-2,round(y0+k\*my));

str(k,s);

outtextxy(x0+10,round(y0-k\*my),s); // Подпись делений

end;

inc(k);

end;

k:=-1000;

while k<1000 do // Цикл для отрисовки делений на оси Х

begin

q2:=round(x0+k\*mx);

if (q2>xl) and (q2<xr) then

begin

if k<>0 then

begin

str(k,s);

line(round(x0+k\*mx),y0+2,round(x0+k\*mx),y0-2);

outtextxy(round(x0+k\*mx),y0+10,s);

end;

end;

inc(k);

end;

x:=x1-1000;

while x<x2+1000 do // Цикл для отрисовки графика функции

begin

y:=Fx(x);

xx1:=x0+round(x\*mx);

yy1:=y0-round(y\*my);

if (xx1>xl) and (xx1<xr) and (yy1>yl) and (yy1<yr) then

begin

putpixel(xx1,yy1,12); // Процедура отрисовки точки по координатам заданного цвета

end;

x:=x+0.0005;

end;

end;

procedure strih; // Процедура для штриховки площади

var

j,t,t1:extended;

begin

j:=x1;

while j<=x2 do

begin

t:=Fx(j);

if (abs(t\*my)>yl) and (abs(t\*my)<yr) and (abs(j\*mx)>xl) and (abs(j\*mx)<xr) then

begin

line(x0+round(j\*mx),y0,x0+round(j\*mx),y0-round(t\*my));

j:=j+0.2;

end

else

begin

if (abs(j\*mx)>xl) and (abs(j\*mx)<xr) then

begin

line(x0+round(j\*mx),y0,x0+round(j\*mx),0);

j:=j+0.2;

end

else

j:=j+0.2;

end;

end;

end;

procedure info; // Процедура вывода информации об управлении

begin

outtextxy(10,10,'Press "+" to zoom in and "-" to zoom out');

outtextxy(10,30,'Use the arrow keys to zoom analog the axes');

outtextxy(10,50,'Up, Down for OY and Left, Right for OX');

outtextxy(10,70,'Press <Enter> to close the graphics window');

outtextxy(10,90,'Press <Tab> to shade the area');

outtextxy(getmaxx-200,getmaxy-50,'y=2x^3-2x^2+2x+3');

end;

procedure mashtab; // Процедура изменения масштаба

var

ch:char;

begin

clrscr;

gd:=detect;

gm:=getmaxmode;

initgraph(gd,gm,''); // Запуск графического режима

SetBkColor (0);

SetColor (15);

mx:=45; // Задание начального масштаба

my:=45;

xl:=0;

yl:=0;

xr:=getmaxx;

yr:=getmaxy;

gx:=getmaxx;

gy:=getmaxy;

x0:=gx div 2;

y0:=gy div 2;

graph; // Отрисовка осей и графика

info; // Вывод информации об управлении

repeat // Цикл для изменения масштаба

ch:=readkey; // считывание нажатой клавиши

if ch=#0 then // Проверка на несимвольную клавишу

begin

ch:=readkey;

case ch of

#72:

begin

SetBkColor (0);

clearviewport;

my:= my\*1.2; // увеличение масштаба по OY

setcolor(15);

graph;

info;

end;

#80:

begin

SetBkColor (0);

clearviewport;

my:= my\*0.8; // уменьшение масштаба по OY

setcolor(15);

graph;

info;

end;

#77:

begin

SetBkColor (0);

clearviewport;

mx:=mx\*1.2; // увеличение масштаба по OX

setcolor(15);

graph;

info;

end;

#75:

begin

SetBkColor (0);

clearviewport;

mx:=mx\*0.8; // уменьшение масштаба по OX

setcolor(15);

graph;

info;

end;

end;

end

else

case ch of

#43:

begin

SetBkColor (0);

clearviewport;

mx:= mx\*1.2; // увеличение масштаба

my:=my\*1.2;

setcolor(15);

graph;

info;

end;

#45:

begin

SetBkColor (0);

clearviewport;

mx:=mx\*0.8; // уменьшение масштаба

my:=my\*0.8;

setcolor(15);

graph;

info;

end;

#9:

begin

if f1 then

begin

strih; // выполнение штриховки

end

else

outtextxy(10,140,'Set the limits of integration');

end;

end;

until ch = #13;

closegraph; // Закрытие графического режима

end;

**begin**

o:=-0.69; // Корень функции

g:=0;

f1:=false; // Установка «флагов»

f2:=false;

f3:=false;

**repeat** // Цикл, с помощью которого меню повторно выводиться на экран

clrscr; // Очистка экрана

writeln('Choose an action'); // Вывод возможных действий

writeln('1-Task Information');

writeln('2-Set limits of integration');

writeln('3-Enter the number of segments to split');

writeln('4-Calculate the area of a shape');

writeln('5-Estimate the error');

writeln('6-Graph');

writeln('0-Exit');

writeln('-----------------------------------------------------------------------');

readln(m);

**case** m **of** // Создание case-меню

1: **begin** // Выводит информацию о задании

write('The graph of the function is in the positive part along the OY axis at x>-0.69');

readln;

**end**;

2:**begin** // Считывает Пределы интегрирования

write('Initial value x=');

readln(x1);

write('End value x=');

readln(x2);

f1:=true;

**if** x2<x1 **then** // Проверка верности введенных данных

**begin**

g:=x1;

x1:=x2;

x2:=g;

**end**;

**if** (x1<o) **and** (x2<o) **then**

**begin**

writeln('For these values, it is impossible to calculate the integral');

writeln('Enter values of X>=-0.69');

f1:=false;

readln;

**end**

**else**

**if** x1<o **then begin**

x1:=-0.69;

**end**;

**end**;

3:**begin** // Считывает количество отрезков разбиения

writeln('Enter the number of segments to split');

readln(n);

f2:=true;

**if** n<0 **then begin** // Проверка верности введенного значения

**repeat**

writeln('Enter values greater than 0');

readln(n);

**until** n>0;

**end**;

**end**;

4:**begin** // Вычисление площади фигуры через средние прямоугольники

**if** (f1) **and** (f2) **then** // Проверка выполнения 2 и 3 пунктов меню

**begin**

integral; // Вызов процедуры, вычисляющей интеграл через средние прямоугольники

readln;

f3:=true;

**end**

**else**

**begin**

write('First, set the limits and the number of split segments');

readln;

**end**;

**end**;

5: **begin** // Оценка погрешности

**if** (f1) **and** (f2) **and** (f3) **then //**Проверка выполнения 2, 3 и 4 пунктов меню

**begin**

integral1; // Вызов процедуры, вычисляющей интеграл через первообразные

writeln('Result through antiderivatives= ',i1:6:4);

writeln('Result using the medium rectangles method= ',s1:6:4);

writeln('-----------------------------------------------------------------------');

writeln('The error');

p:=i1-s1;

writeln(p:6:4); // Вывод погрешности

readln;

**end**

**else**

**begin**

write('First, set the limits and the number of split segments and calculate the area of the shape');

readln;

**end**;

**end**;

6:begin

mashtab; // отрисовка графика с осями и возможностью мастабирования

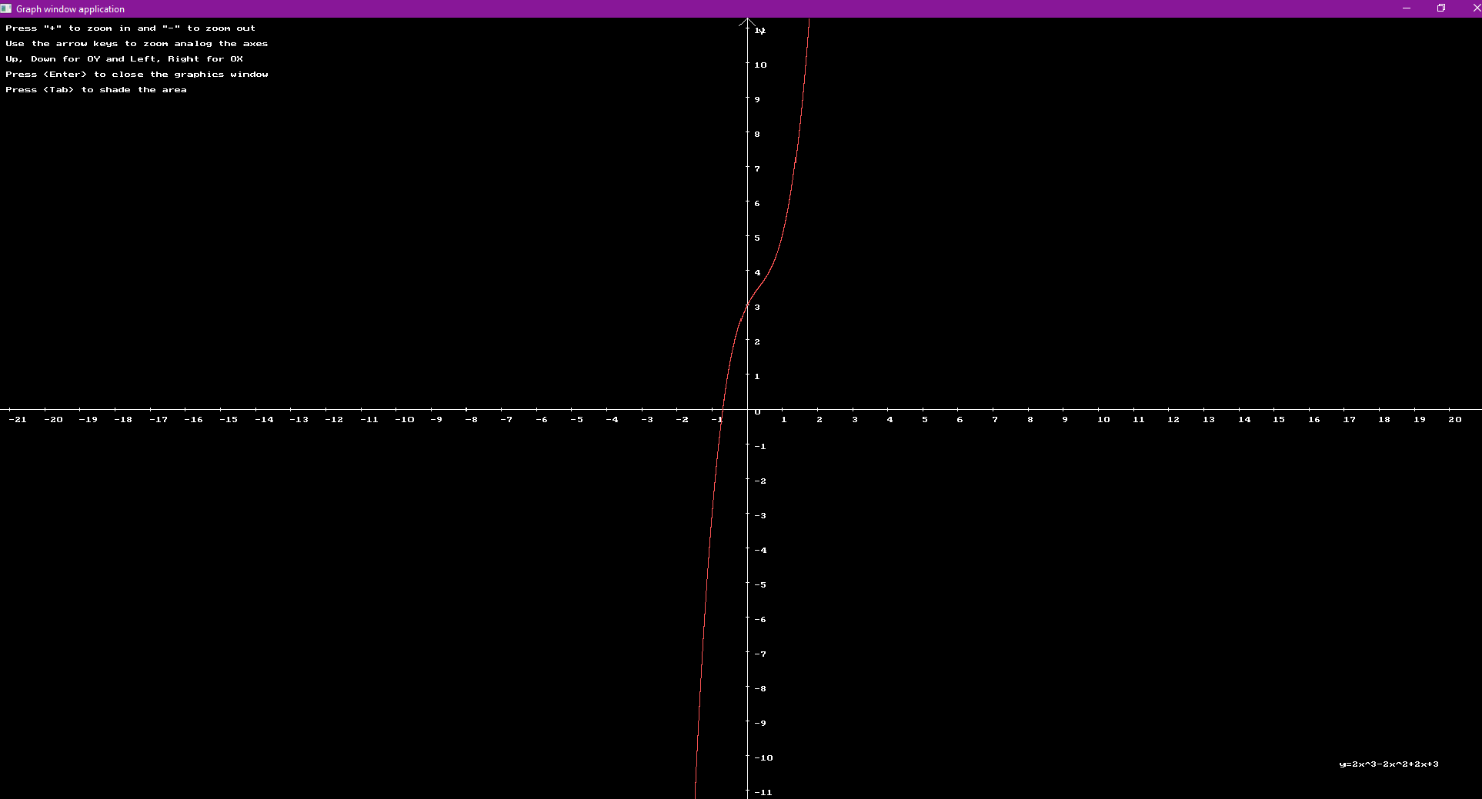
end;

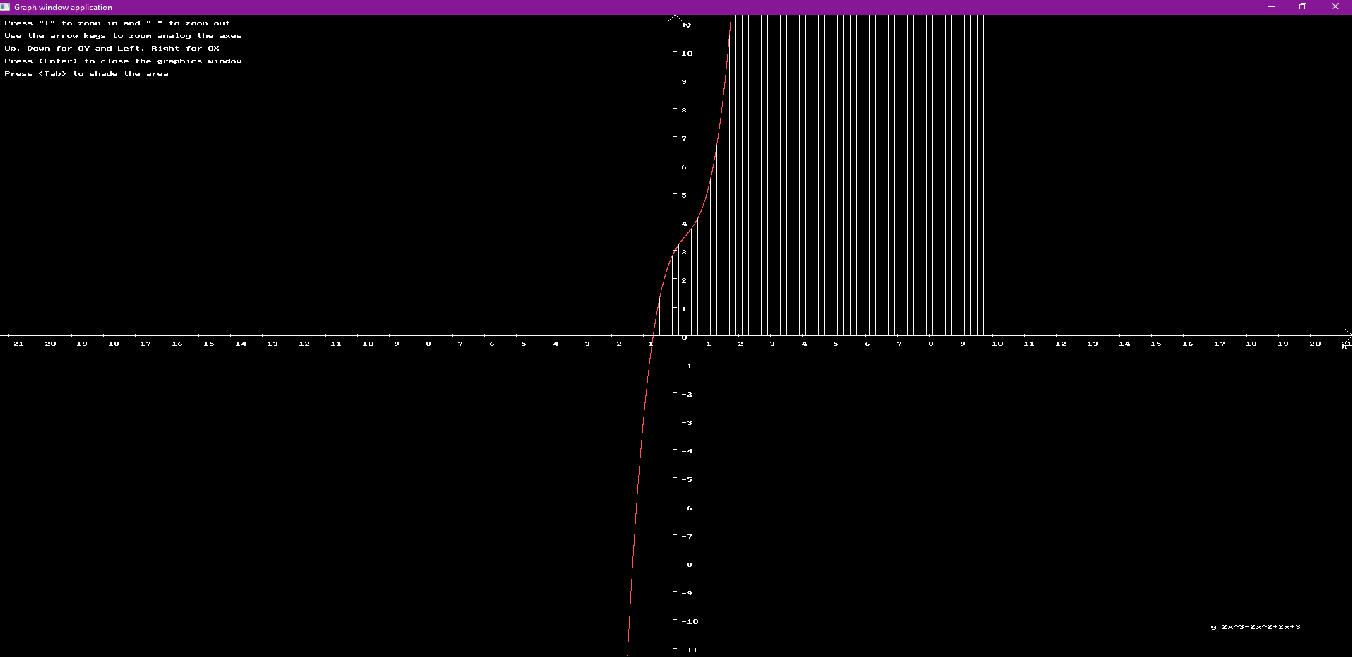
**end**;

**until** m = 0; // Значение переменной m, при котором программа завершит работу

**end**.

***Экранные формы:***







***Вывод.***

Данная лабораторная работа помоглаосвоить принципы работы в графическом режиме, получить базовые навыки взаимодействия с графическими примитивами.

В ходе выполнения данной лабораторной работы использовался графический модуль «graph», который инициализируется с помощью процедуры «initgraph» и закрывается с помощью процедуры «closegraph». Данный модуль позволил работать в графическом режиме.

Для отрисовки линий и точек при изображении на экране координатных осей и графика функции использовались процедуры «line» и «putpixel» соответственно. Для подписей осей использовалась процедура «outtextxy».

Возможность масштабировать график реализована при помощи функции «readkey». Она позволила назначить на определенную клавишу свое действие, такое как увеличение или уменьшение масштаба.

Так четвертая лабораторная работа позволила освежить и закрепить знания об языке программирования Pascal, а также познакомила с новым материалом.