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-205-52-00

Веселов Алексей Денисович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

1 Цель работы

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

2 Формулировка задания (с вариантом)

Вариант: 4

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.

3 Описание алгоритма

Задача не из простых, для удобства разделим её на 6 частей. Одной для взаимодействия с пользователем (case меню), и 5 вычислительных частей.

Первая часть, она же продпрограмма, вычисление y = x ^ 3 + (2) \* x ^ 2 + (-2) \*x + (8). В дальнейшем она будет использоваться в нашем методе.

Вторая подпрограмма – вычисление первообразной от функции.

Третья подпрограмма находит крайнюю левую точку пересечения графика с осью.

Четвёртая подпрограмма — вычисление Формулы Ньютона-Лейбница.

Пятая подпрограмма — вычисление методом Симпсона, тут нужно ввести количество промежутков, чем их больше, тем точнее значение получиться. Далее путём не хитрых махинаций “рисуем” параболы и считаем их площадь.

Шестая подпрограмма — это кейс меню для взаимодействия с пользователем. В нем есть всё вышеперечисленные подпрограммы.

Добавлена подпрограмма для вывода графика

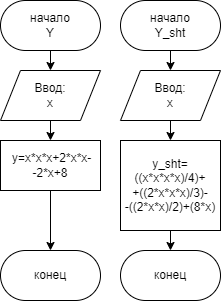
4. Схема алгоритма с комментарием

Рисунок 1 Процедуры, Y и Y\_sht

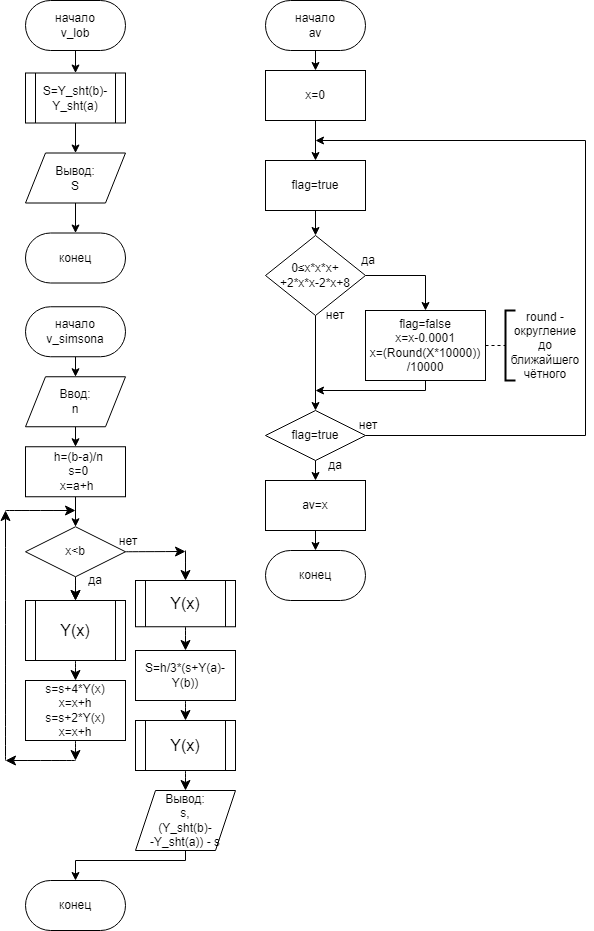


Рисунок 2 Процедуры v\_lob, av, v\_simsona

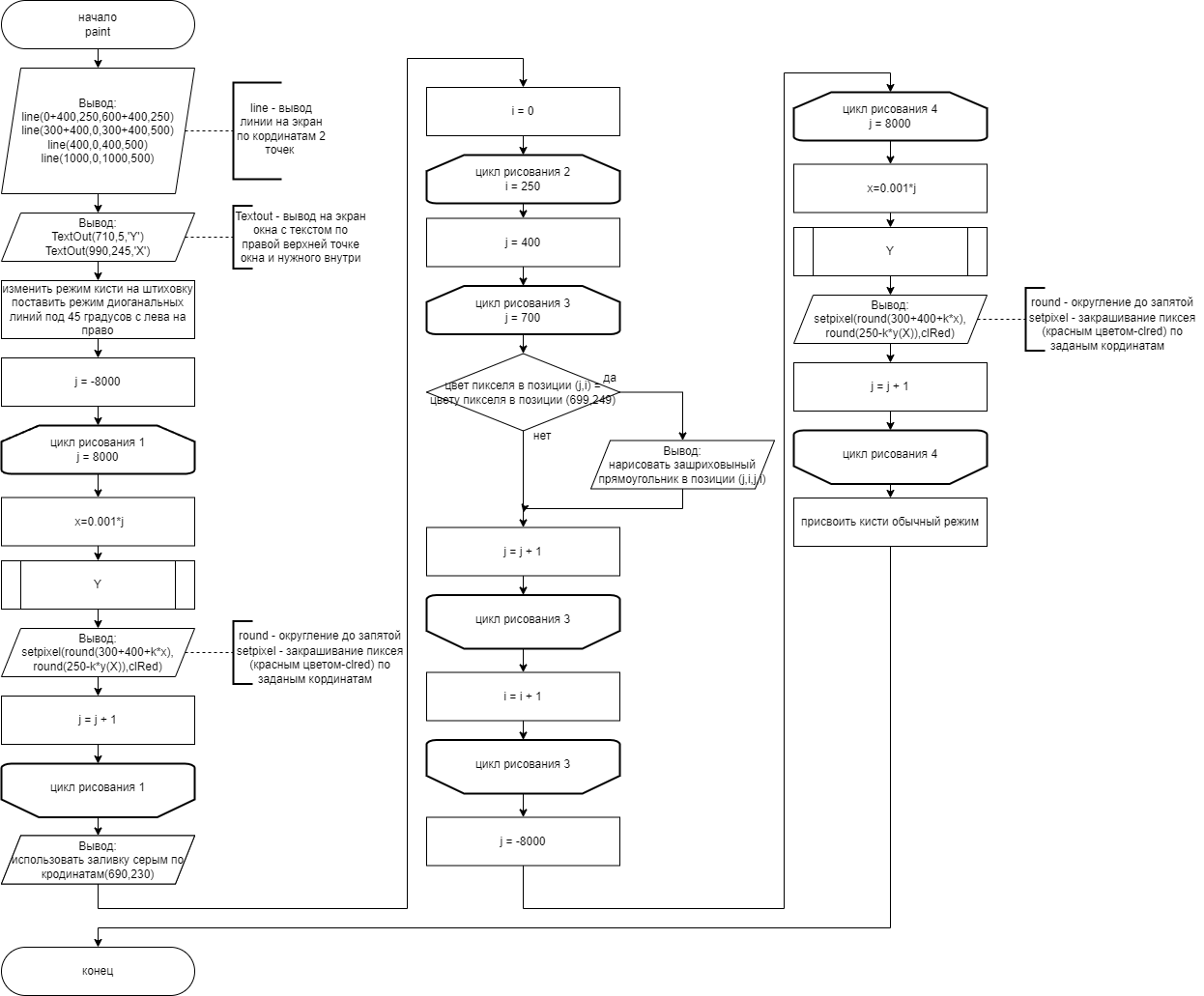


Рисунок 3 продпрограмма для рисования

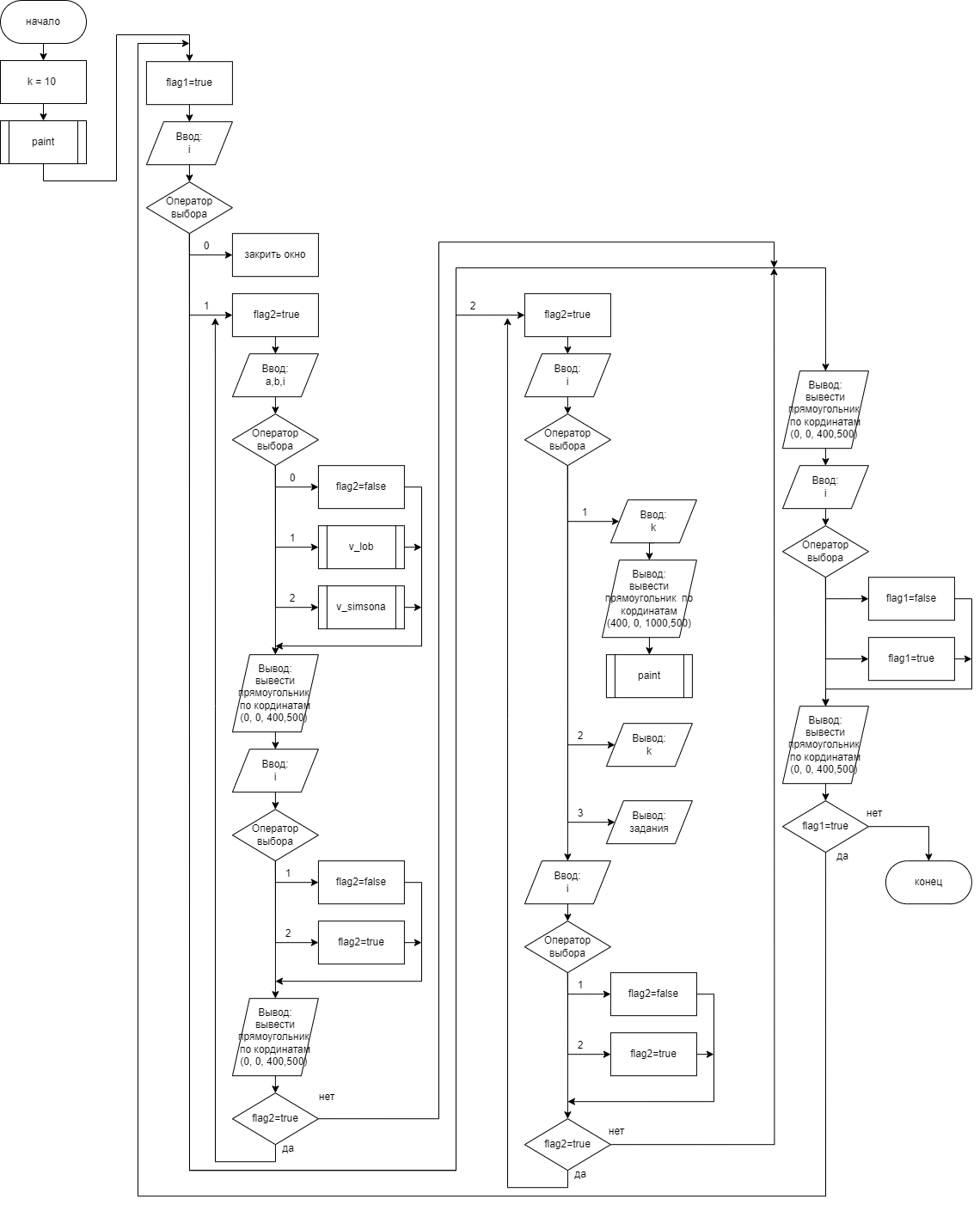


Рисунок 4 case меню

5 Код программы

**program** z1; ///f=x\*x\*x+2\*x\*x-2\*x+8

**uses** graphabc;

**var**

a,b,S : Real; // подсчёт базовым методом

h,x : real; // не базовым

i : integer; // кейс меню

flag2 : boolean; //

j,k : integer; // дкр 4, рисование

flag1 : boolean; // кейс меню

a1,b1 : integer;

////////////////////////////////дкр 3//////////////////////////////////////////

**function** Y (x:real) :real; //функция

**begin**

y:=x\*x\*x+2\*x\*x-2\*x+8;

**end**;

**function** Y\_sht (x:real) :real; //первообразная от функции

**begin**

y\_sht:=((x\*x\*x\*x)/4)+((2\*x\*x\*x)/3)-((2\*x\*x)/2)+(8\*x);

**end**;

**function** av:real; // возможно непотребуеться, это вычисление точки пересечения с осью Y

**var** flag:boolean;

x:real;

**begin**

x:=0;

**repeat**

flag:=true;

**if** 0<=x\*x\*x+2\*x\*x-2\*x+8 **then begin**

flag:=false;

x-=0.0001;

X:=(Round(X\*10000))/10000;

**end**;

**until** flag=true;

av:=x;//-3.3251 , это её числовое значение

**end**;

{procedure pregel(var a,b :real); // ввод пределов

var flag:boolean;

begin

repeat

flag:=true;

writeln('Введите пределы интегрирования ф-ии [-3.3251;0]');

read(a,b);

writeln('[',a,';',b,']');

if (av<a) and (a>0) then begin

flag:=false;

writeln('предел а недопустим');

end;

if (av<b) and (b>0) then begin

flag:=false;

writeln('предел b недопустим');

end;

if a>b then begin

flag:=false;

writeln('предел a и b недопустим');

end;

until flag=true;

end;}

**procedure** v\_lob; // поиск по формуле: интеграл нижний a предел верхний b предел f(x)=F(b)-F(a)

**begin**

writeln('Вы выбрали базовый метод');

//в лоб

S:=Y\_sht(b)-Y\_sht(a);

writeln('Интеграл = ',s);

**end**;

**procedure** v\_simsona;

**var** n : integer;

**begin**

//в симсонa

writeln('Вы выбрали метод Симпсона');

writeln('На сколько частей делим промежуток?');

Readln(n);

h:=(b-a)/n;// мб спросить на сколько частей?(1000=n)

s:=0;

x:=a+h;

**while** x<b **do begin**

s:=s+4\*Y(x);

x:=x+h;

s:=s+2\*Y(x);

x:=x+h;

**end**;

S:=h/3\*(s+Y(a)-Y(b));

writeln('Интеграл = ',s);

writeln('Погрешность = ', (Y\_sht(b)-Y\_sht(a)) - s);

**end**;

**procedure** paint;

**begin**

**var** i,j:integer;

setWindowSize(1500,500);//длина окна

CenterWindow;

//setpixel(400,250,clRed);

line(0+400,250,600+400,250);// ось Х

line(300+400,0,300+400,500);// ось Y

line(400,0,400,500);// поле ввода

line(1000,0,1000,500);

TextOut(710,5,'Y'); // ось Y

TextOut(990,245,'X');// ось Х

setbrushstyle(bsHatch);

Brush.Hatch:=bhBackwardDiagonal;

**for** j:=-8000 **to** 8000 **do** // вывод графика отмечая точки

**begin**

x:=0.001\*j;

setpixel(round(300+400+k\*x),round(250-k\*y(X)),clRed);

**end**;

FloodFill(690,230,clDimGray);

**for** i:=0 **to** 250 **do**

**for** j:=400 **to** 700 **do**

**if** GetPixel(j,i) = GetPixel(699,249) **then** Rectangle(j,i,j,i);

**for** j:=-8000 **to** 8000 **do** // вывод графика отмечая точки

**begin**

x:=0.001\*j;

setpixel(round(300+400+k\*x),round(250-k\*y(X)),clRed);

**end**;

setbrushstyle(bsSolid);

**end**;

**begin**

k:=10;// измениение маштаба графика

paint;

**begin**// это кейс меню

**repeat**

flag1:=true;

writeln('0:закрыть окно');

writeln('1:работа с интегралом');

writeln('2:работа с графиком');

readln(i);

writeln(i);

**case** i **of**

0:CloseWindow;

1:**begin**// это кейс меню дкр 3

**repeat**

flag2:=true;

writeln('Нахождение интеграла y=x^(3)+2x^(2)-2\*x+8');

writeln('Введите пределы интегрирования ф-ии [-3.3251;∞]');

read(a,b);

writeln('Выберете метод интегрирования');

writeln('0:ввести заново');

writeln('1:базовый метод интегрирования');

writeln('2:метод Симпсона');

readln(i);

writeln(i);

**case** i **of**

0:flag2:=false;

1:v\_lob;

2:v\_simsona;

**end**;

writeln('Погрешность = ',Y(a)+Y(b)-s);

Rectangle(0, 0, 400,500);

writeln('Хотите повторить ввод?');

writeln('1:Да');

writeln('2:Нет');

readln(i);

writeln(i);

**case** i **of**

1:flag2:=false;

2:flag2:=true;

**end**;

Rectangle(0, 0, 400,500);

**until** flag2=true;

**end**;

2:**begin**

**repeat**

flag2:=true;

writeln('маштабирование графика');

writeln('текущий размер ',k);

writeln('1:Да');

writeln('2:Нет');

writeln('3:Вывести задание');

readln(i);

writeln(i);

**case** i **of**

1:**begin**

writeln('Введите размер [4-35]');

Read(k);// измениение маштаба графика

Rectangle(400, 0, 1000,500);// очистка экрана

paint;

**end**;

2:writeln('текущий размер остался ',k);

3:**begin**

writeln('Реализовать не менее двух возможностей из представленных:');

writeln('независимое масштабирование по осям,');

writeln('штриховка вычисляемой площади,');

writeln('визуализация численного расчета интеграла.');

**end**;

**end**;

writeln('Хотите повторить ввод?');

writeln('1:Да');

writeln('2:Нет');

readln(i);

writeln(i);

**case** i **of**

1:flag2:=false;

2:flag2:=true;

**end**;

**until** flag2=true;

**end**;

**end**;

Rectangle(0, 0, 400,500);

writeln('Хотите повторить ввод?');

writeln('1:Да');

writeln('2:Нет');

readln(i);

writeln(i);

**case** i **of**

1:flag1:=false;

2:flag1:=true;

**end**;

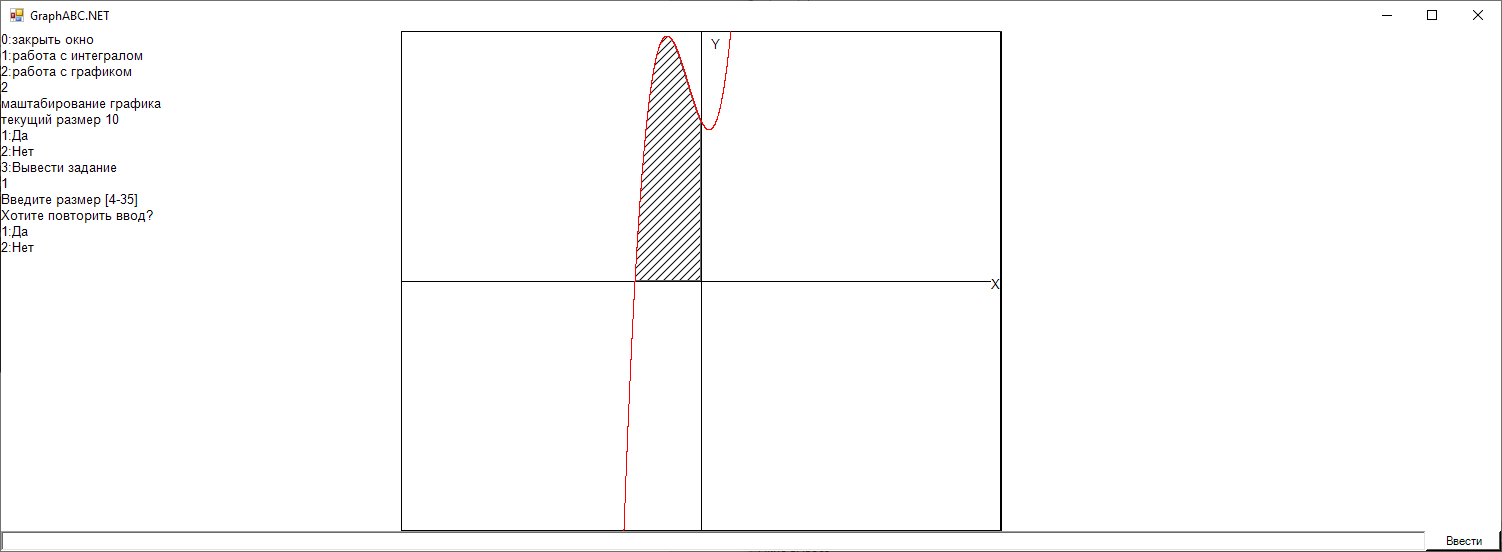
Rectangle(0, 0, 400,500);

**until** flag1=true;

**end**;

**end**.

6 Результат выполнения программы



7 Вывод

Данная работа оказалась сложной.

Был изучен метод Симпсона также называемый метод парабол. Изначально было не понятно, как реализовать вычисление этим методом. Но решение было найдено благодаря сети интернет.

Так же была реализована, затем удалена подпрограмма для ввода корректных ограничений, но так как такого задания не дано, было принято решение удалить данную часть кода (но она ещё осталась в виде пояснения).

Изначально была идея не писать количество промежутков, но как выяснилось это потребуется в следующей домашней контрольной работе.

В новой процедуре сложностей не было

Само собой цель работы была достигнута.