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-203-52-00

Гагаринов Владислав Максимович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2022

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

**program** DKR4;

**uses** graphABC;

**var** input: byte; m1, m2, p, u: real; n: integer;

**function** kos(**var** v: byte): byte;

**begin**

readln(v);

kos := v;

writeln;

**end**;

**function** kos1(**var** x: real): real;

**var**

f, s: real;

**begin**

f:=1 \* power(x,3) + (-2) \* power(x,2) +(-5)\*x +(17);

kos1 := f;

**end**;

**function** kos2(**var** x: real): real;

**begin**

**var** f1: real;

f1:=(x\*(3\*power(x,3)-8\*sqr(x)-30\*x+204))/12;

kos2 := f1;

**end**;

**function** graph(**var** s1, s2, h: real; ras: integer): integer;

**var**

x, mx, my: real; a, b, x0, y0, i: integer;

**begin**

MaximizeWindow;

clearwindow;

setpencolor(clblack);

a := -5;

b := 100;

x0 := windowwidth **div** 2;

y0 := windowheight **div** 2;

mx := m1;

my := m2;

line(0, y0, windowwidth, y0);

line(x0, 0, x0, windowheight);

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

**begin**

line(x0 + round(i \* mx), y0 - 3, x0 + round(i \* mx), y0 + 3);

line(x0 - round(i \* mx), y0 - 3, x0 - round(i \* mx), y0 + 3);

line(x0 - 3, y0 + round(i \* my), x0 + 3, y0 + round(i \* my));

line(x0 - 3, y0 - round(i \* my), x0 + 3, y0 - round(i \* my));

textout(x0 + round(i \* mx), y0 + 10, inttostr(i));

textout(x0 - round(i \* mx), y0 + 5, inttostr(-i));

textout(x0 - 25, y0 - round(i \* my), inttostr(i));

textout(x0 - 20, y0 + round(i \* my), inttostr(-i));

**end**;

textout(x0 + 5, y0 + 10, '0');

textout(windowwidth - 10, y0 - 15, 'X');

textout(x0 + 5, 10, 'Y');

x := a;

setpencolor(clgreen);

line(x0 + round(s1 \* mx), 0, x0 + round(s1 \* mx), windowheight);

setpencolor(clgreen);

line(x0 + round(s2 \* mx), 0, x0 + round(s2 \* mx), windowheight);

**var** l, w, k, e: real;

l := round((x0 + round(s2 \* mx) - x0 - round(s1 \* mx)) / n);

k := l;

p := (s2 - s1) / ras;

w := 0;

x := a;

**while** x <= b **do**

**begin**

**if** (x0 + round(s1 \* mx)) = (x0 + round(x \* mx)) **then**

**begin**

**if** y0 - round(kos1(x) \* my) < y0 **then**

**begin**

setpencolor(clgreen);

u := p / 2 + x;

Rectangle(x0 + round(s1 \* mx), y0 - round(kos1(u) \* my), x0 + round(s1 \* mx + l), y0);

**end**;

**end**;

**if** (x0 + round(s1 \* mx + l)) = (x0 + round(x \* mx)) **then**

**begin**

**if** y0 - round(kos1(x) \* my) < y0 **then**

**begin**

setpencolor(clgreen);

**for var** v := x0 + round(s1 \* mx - l) **to** x0 + round(s2 \* mx) **do**

**begin**

Rectangle(x0 + round(s1 \* mx + l), y0 - round(kos1(x) \* my), x0 + round(s1 \* mx + (k + l)), y0);

**end**;

**end**;

l := l + k;

w := w + 1;

**if** (w + 1) = n **then**

**break**;

**end**;

x := x + 0.001;

**end**;

x := a;

**while** x <= b **do**

**begin**

setpixel(x0 + round(x \* mx), y0 - round(kos1(x) \* my), clred);

x := x + 0.001;

**end**;

**end**;

**function** graphdop(**var** s1, s2, h: real; ras: integer): integer;

**var**

con: integer; s: string;

**begin**

con := 0;

**repeat**

s := 'Масштаб по x: ' + m1;

textout(0, 10, s);

s := 'Масштаб по у: ' + m2;

textout(0, 30, s);

textout(0, 50, 'Изменить маштаб по x - 1');

textout(0, 70, 'Изменить маштаб по y - 2');

textout(0, 90, 'Продолжить - 0');

read(input);

**case** input **of**

1:

**begin**

textout(0, 130, 'Введите маштаб от 10 до 50');

readln(m1);

graph(s1, s2, h, ras);

**end**;

2:

**begin**

textout(0, 130, 'Введите маштаб от 10 до 50');

readln(m2);

graph(s1, s2, h, ras);

**end**;

0: con := 1;

**end**;

**until** con = 1;

graph(s1, s2, h, ras);

**end**;

**function** main: integer;

**var**

a, b, h, f, x, s, pog: real; ss: string;

**begin**

clearwindow;

textout(0, 0, 'Вычисление площади фигуры, ограниченной кривой 1\*x^3+(-2)\*x^2+(-5)\*x+(17) и осью Ох (в положительной части по оси Оу)');

textout(0, 15, 'Введите пределы интегрирования a и b и количество разбиений n:');

readln(a, b, n);

ss := a + ' ' + b + ' ' + n;

textout(415, 15, ss);

h := (b - a) / n;

x := a;

m1 := 20;

m2 := 20;

**for var** i := 0 **to** n **do**

**begin**

f := kos1(x);

s := s + f;

x := x + h;

**end**;

m1 := 20;

m2 := 20;

s := s \* h;

s := Round(s \* 1000) / 1000;

ss := 'Площадь =' + s;

textout(0, 50, ss);

textout(0, 70, 'Вывести погрешность?');

textout(0, 110, 'Да - 1');

textout(0, 130, 'Нет - 0');

read(input);

**case** input **of**

1:

**begin**

pog :=

abs((kos2(b) - kos2(a)) - s);

pog := Round(pog \* 1000) / 1000;

ss := 'Погрешность =' + pog;

textout(0, 180, ss);

textout(0, 200, 'Нажмите enter чтоб продолжить');

readln();

readln;

**end**;

0:

**begin**

textout(0, 150, 'Нажмите enter чтоб продолжить');

readln();

readln;

**end**;

**end**;

graph(a, b, h, n);

graphdop(a, b, h, n);

textout(0, 10, 'Запустить программу заново?');

textout(0, 30, 'Да - 1');

textout(0, 50, 'Нет - 0');

read(input);

**case** input **of**

1: main;

0: **exit**();

**end**;

main := 0;

**end**;

**begin**

MaximizeWindow;

Writeln('Вычисление площади фигуры, ограниченной кривой 1\*x^3+(-2)\*x^2+(-5)\*x+(17) и осью Ох (в положительной части по оси Оу)');

Writeln('Для ввода пределов интегрирования введите 1, для выхода введите 0');

input := kos(input);

**case** input **of**

1: main;

**end**;

**end**.

1. Результат выполнения программы:

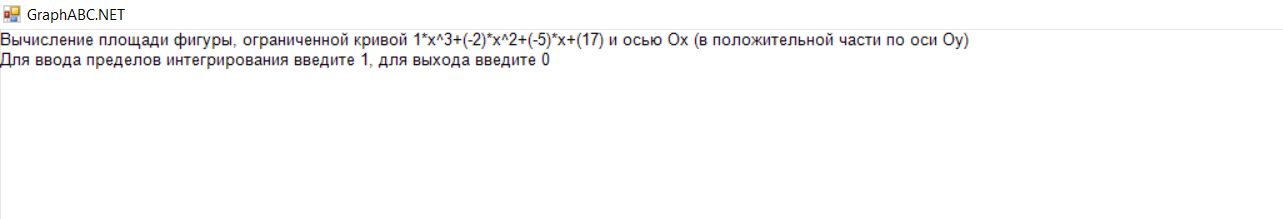


Рисунок 1 – Результат выполнения программы

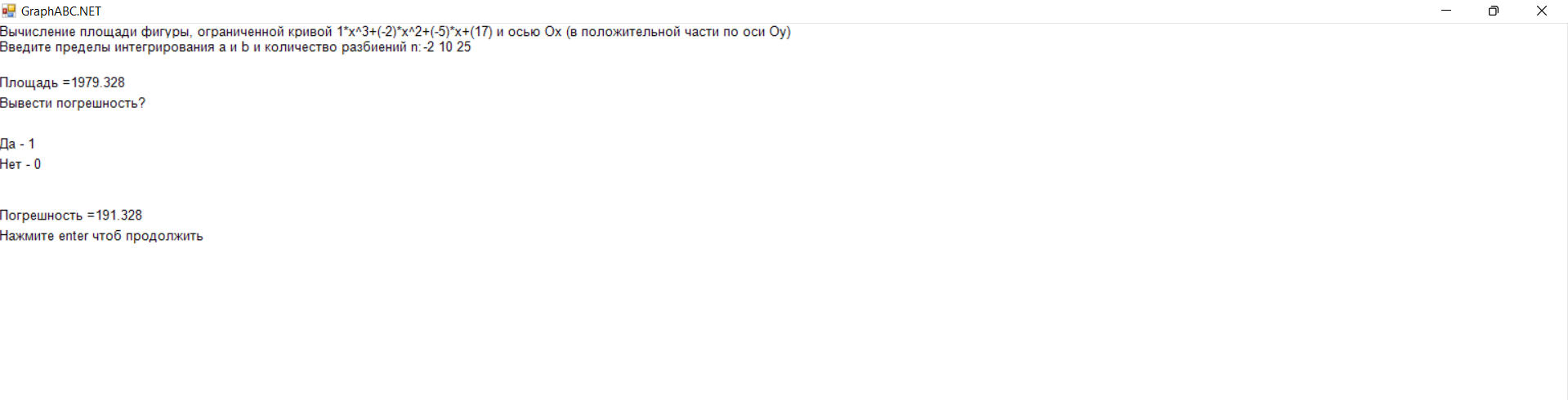


Рисунок 2 – Результат выполнения программы

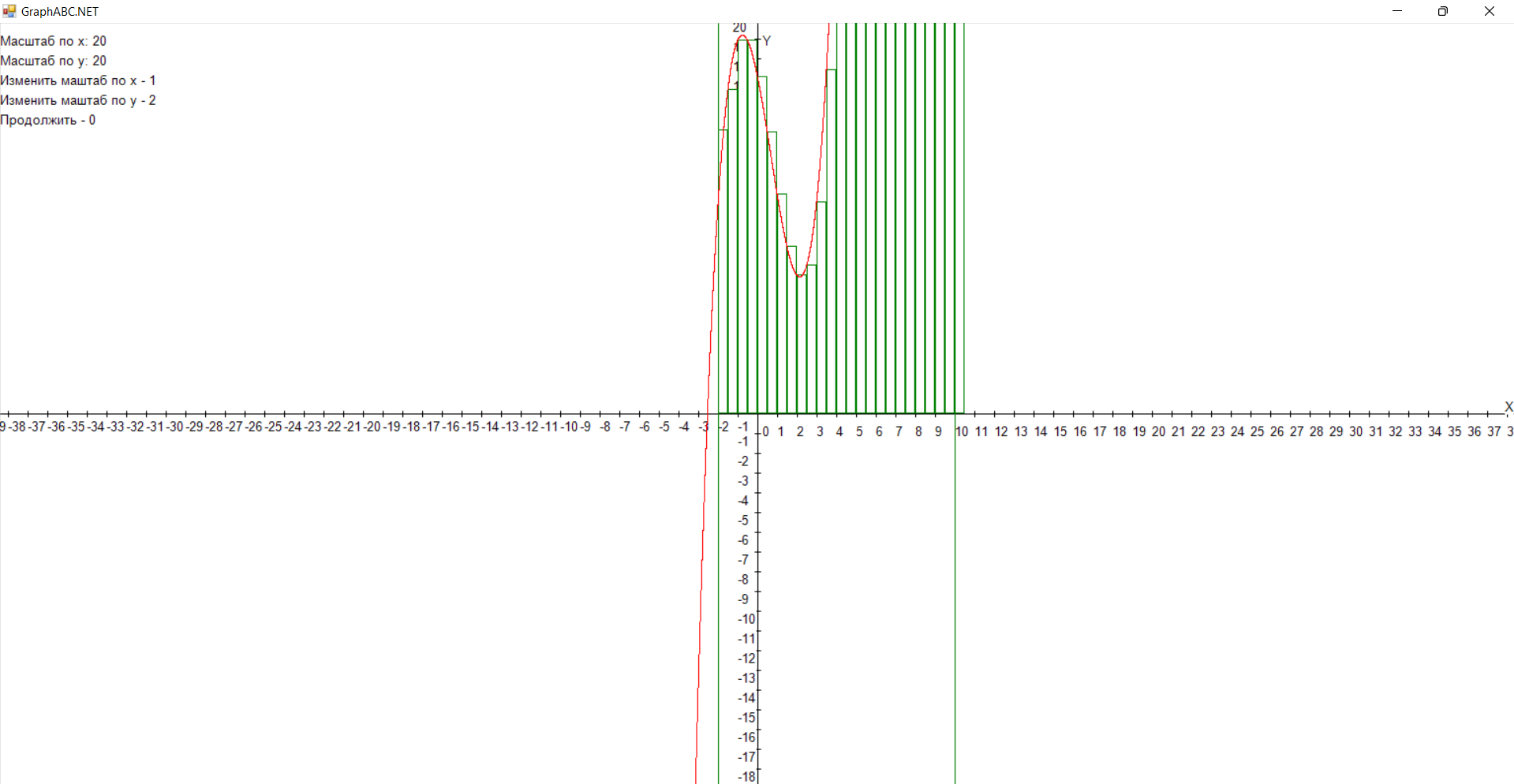


Рисунок 3 – Результат выполнения программы

5. Вывод

Для решения домашней контрольной работы №4 была изучена библиотека GraphABC в языке программирования PascalABС. В ходе работы мы освоили принципы работы в графическом режиме; получили базовые навыки взаимодействия с графическими примитивами