Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО КОНТРОЛЬНОЙ РАБОТЕ №3**

**«ИЗУЧЕНИЕ БАЗОВЫХ ПРИНЦИПОВ ОРГАНИЗАЦИИ ПРОЦЕДУР И ФУНКЦИЙ»**

**ПО ДИСЦИПЛИНЕ «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»**

Выполнил: студент учебной группы

ИСПк-204-52-00

Зяблицев Артем Павлович

Преподаватель:

Сергеева Елизавет Григорьевна

Киров

2022

В отчете должны отображаться:

* Цель лабораторной работы
* Формулировка задания (с вариантом)
* Код программы
* Результат выполнения программы
* Вывод

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

Задание: вариант 7

1. Дополнить программу, реализованную в ходу предыдущей лабораторной работы, режимом визуализации.
2. Предусмотреть возможность вывода кривой, ограничивающей фигуры, 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, визуализации численного расчета интеграла.

Код программы

**uses** crt;

**uses** graphABC;

**var**

mx, my, e,p: real;

a, b, x0, y0, n, w, l, k: integer;

casem, vibo: byte;

**function** F1(x: real): real;

**begin**

F1 := 1\*power(x,3)+(-1)\*power(x,2)+(-1)\*x+(14);

**end**;

**function** vibor(**var** s: byte): byte;

**begin**

Textcolor(2);

writeln('Увеличить по x - 1');

Textcolor(4);

writeln('Увеличить по y - 2');

Textcolor(15);

writeln('Увеличить по x и y - 3');

Writeln('Выберите вариант: ');

readln(s);

vibor := s;

**end**;

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

**var**

i: integer; x: real;

**begin**

ClearWindow;

a := -5;

b := 50;

x0 := windowwidth **div** 2;

y0 := windowheight **div** 2;

vibo := vibor(vibo);

**case** vibo **of**

1:

**begin**

writeln('Введите размер промежутка по x');

readln(mx);

my := 10;

**end**;

2:

**begin**

writeln('Введите размер промежутка по y');

mx := 10;

readln(my);

**end**;

3:

**begin**

writeln('Введите размер промежутка по y');

readln(my);

writeln('Введите размер промежутка по x');

readln(mx);

**end**;

**end**;

line(0, y0, windowwidth, y0);

line(x0, 0, x0, windowheight);

**for** i := 1 **to** windowwidth **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);

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

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

**end**;

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

**begin**

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 - 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');

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

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

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

k := l;

p :=(s2-s1)/ras;

w := 0;

x := a;

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

**begin**

setpixel(x0 + round(x \* mx), y0 - round(F1(x) \* my), clred);

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

**begin**

setpencolor(clnavy);

e := x;

Rectangle(x0 + round(s1 \* mx), round(y0 - round(F1(p/2+e)\*my)), x0 + round(s1 \* mx + (k + l)), y0);

**end**;

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

**begin**

setpencolor(clnavy);

e := x;

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

Rectangle(x0 + round(s1 \* mx + l), round(y0 - round(F1(p/2+e) \* my) ), x0 + round(s1 \* mx + (k + l)), y0);

l := l + k;

e := e + l;

inc(w);

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

**begin**

setpencolor(clblack);

**break**;

**end**;

**end**;

x := x + 0.001;

**end**;

textout(10, 10, 'Найти площади фигуры, ограниченной кривой 1\*power(x,3)+(-1)\*power(x,2)+(-1)\*x+(14) и осью Ох (в положительной части по оси Оу)');

textout(10, 900, 'Нажмите на "Enter" что бы продолжить');

readln;

ClearWindow;

**end**;

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

**var**

i: integer; x: real;

**begin**

ClearWindow;

a := -5;

b := 50;

x0 := windowwidth **div** 2;

y0 := windowheight **div** 2;

mx := 10;

my := 10;

line(0, y0, windowwidth, y0);

line(x0, 0, x0, windowheight);

**for** i := 1 **to** windowwidth **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);

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

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

**end**;

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

**begin**

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 - 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');

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

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

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

k := l;

p:=(s2-s1)/ras;

w := 0;

x := a;

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

**begin**

setpixel(x0 + round(x \* mx), y0 - round(F1(x) \* my), clred);

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

**begin**

setpencolor(clnavy);

e := x;

Rectangle(x0 + round(s1 \* mx), round(y0 - round(F1(p/2+e) \* my) \* 1.4), x0 + round(s1 \* mx + (k + l)), y0);

**end**;

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

**begin**

setpencolor(clnavy);

e := x;

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

Rectangle(x0 + round(s1 \* mx + l), round(y0 - round(F1(p/2+e) \* my) \* 1.4), x0 + round(s1 \* mx + (k + l)), y0);

l := l + k;

e := e + l;

inc(w);

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

**begin**

setpencolor(clblack);

**break**;

**end**;

**end**;

x := x + 0.001;

**end**;

textout(10, 10, 'Найти площади фигуры, ограниченной кривой 1\*power(x,3)+(-1)\*power(x,2)+(-1)\*x+(14) и осью Ох (в положительной части по оси Оу)');

textout(10, 600, 'Нажмите на "Enter" что бы продолжить');

readln;

ClearWindow;

**end**;

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

**begin**

Textcolor(2);

writeln('Да - 1');

Textcolor(4);

writeln('Нет - 0');

Textcolor(15);

Writeln('Выберите вариант: ');

readln(v);

casemf := v;

**end**;

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

**begin**

**var** f1: real;

f1 := 1\*power(x,3)+(-1)\*power(x,2)+(-1)\*x+(14);

func := f1;

**end**;

**function** func1(**var** x: real): real;

**begin**

**var** f2: real;

f2 := 1/4 \* power(x,4)- 1/3 \* power(x,3)- 1/2 \* power(x,2)+14\*x;

func1 := f2;

**end**;

**function** predel: integer;

**begin**

ClrScr;

**var** a, b, h, f, x, S: real;

**var** n: integer;

Textcolor(5);

Writeln('Вычисление площади фигуры, ограниченной кривой 1\*power(x,3)+(-1)\*power(x,2)+(-1)\*x+(14) и осью Ох (в положительной части по оси Оу)');

Textcolor(3);

print('Введите пределы интегрирования "a"<"b":');

read(a, b);

writeln(a, ' ', b);

print('Количество интервалов разбиения:');

read(n);

writeln(n);

h:= (b-a)/n;

x :=a+h/2;

**for var** i := 1 **to** n - 1 **do**

**begin**

f := func(x);

S := S + f;

x := x + h;

**end**;

S := S \* h;

Textcolor(10);

writeln('Ответ ', S);

writeln;

Textcolor(15);

writeln('Вывести погрешность полученного результата?');

casem := casemf(casem);

**case** casem **of**

1:

**begin**

Textcolor(6);

writeln('Погрешность = ', abs((func(b) - func1(a)) - S):10:3);

**end**;

0:

**else**

textcolor(black);

Textbackground(red);

writeln('Такого варианта нет');

textbackground(black);

textcolor(white);

**end**;

writeln;

writeln('Рисуем график');

casem := casemf(casem);

**case** casem **of**

1: grap(a, b, h, n);

0:

**else**

textcolor(black);

Textbackground(red);

writeln('Такого варианта нет');

textbackground(black);

textcolor(white);

**end**;

writeln('Увеличить размер графика');

casem := casemf(casem);

**case** casem **of**

1: grapx(a, b, h, n);

0:

**else**

textcolor(black);

Textbackground(red);

writeln('Такого варианта нет');

textbackground(black);

textcolor(white);

**end**;

writeln;

Textcolor(15);

writeln('Начать заново?');

casem := casemf(casem);

**case** casem **of**

1: predel;

0:

**begin**

textcolor(red);

writeln('Досвидание');

textcolor(white);

**exit**()

**end**

**else**

textcolor(black);

Textbackground(red);

writeln('Такого выбора нет');

writeln('Перезапустите программу');

textbackground(black);

textcolor(white);

**end**;

predel := 0;

**end**;

**begin**

ClrScr;

MaximizeWindow;

Textcolor(5);

Writeln('Вычисление площади фигуры, ограниченной кривой 1\*power(x,3)+(-1)\*power(x,2)+(-1)\*x+(14)) и осью Ох (в положительной части по оси Оу)');

Textcolor(3);

Writeln('Ввести пределы интегрирования в ручную?');

casem := casemf(casem);

**case** casem **of**

1: predel;

0:

**begin**

textcolor(red);

writeln('Досвидание');

textcolor(white);

**exit**()

**end**

**else**

textcolor(black);

Textbackground(red);

writeln('Такого варианта нет');

writeln('Перезапустите программу');

textbackground(black);

textcolor(white);

**end**;

**end**.

Результат

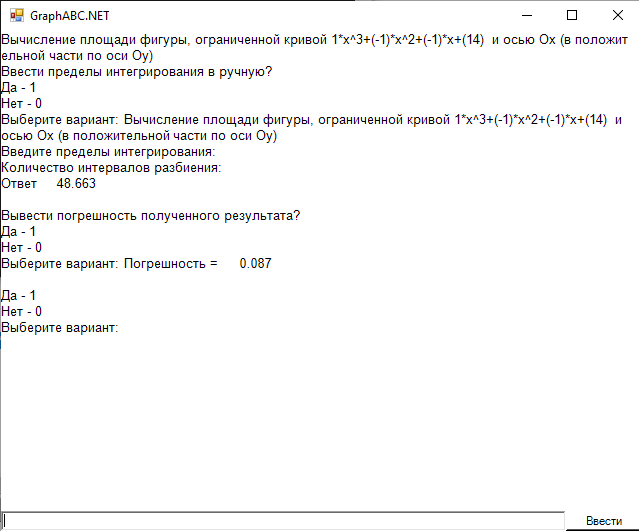


Рисунок 1 – Результат программы

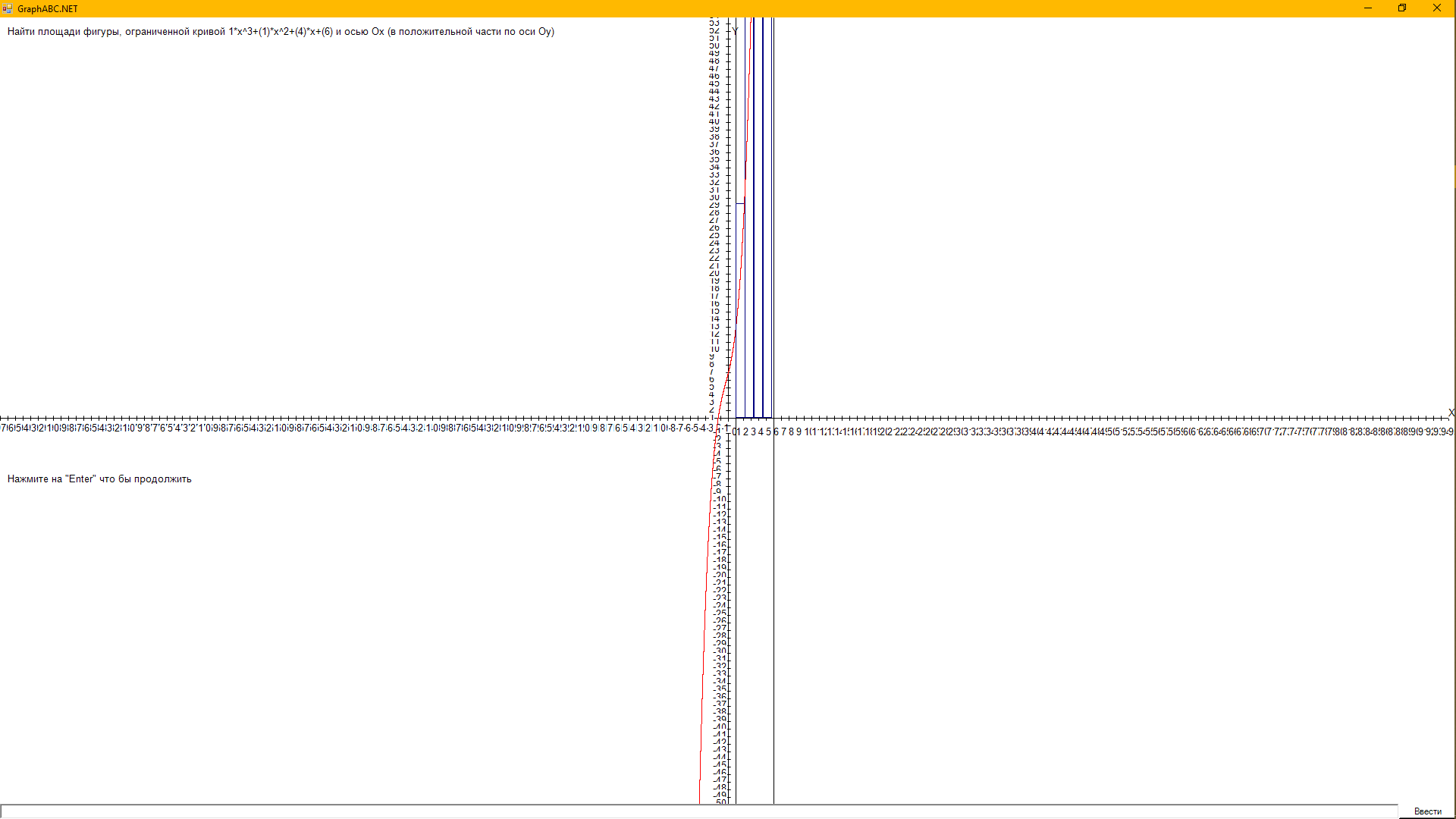


Рисунок 2 – результат программы

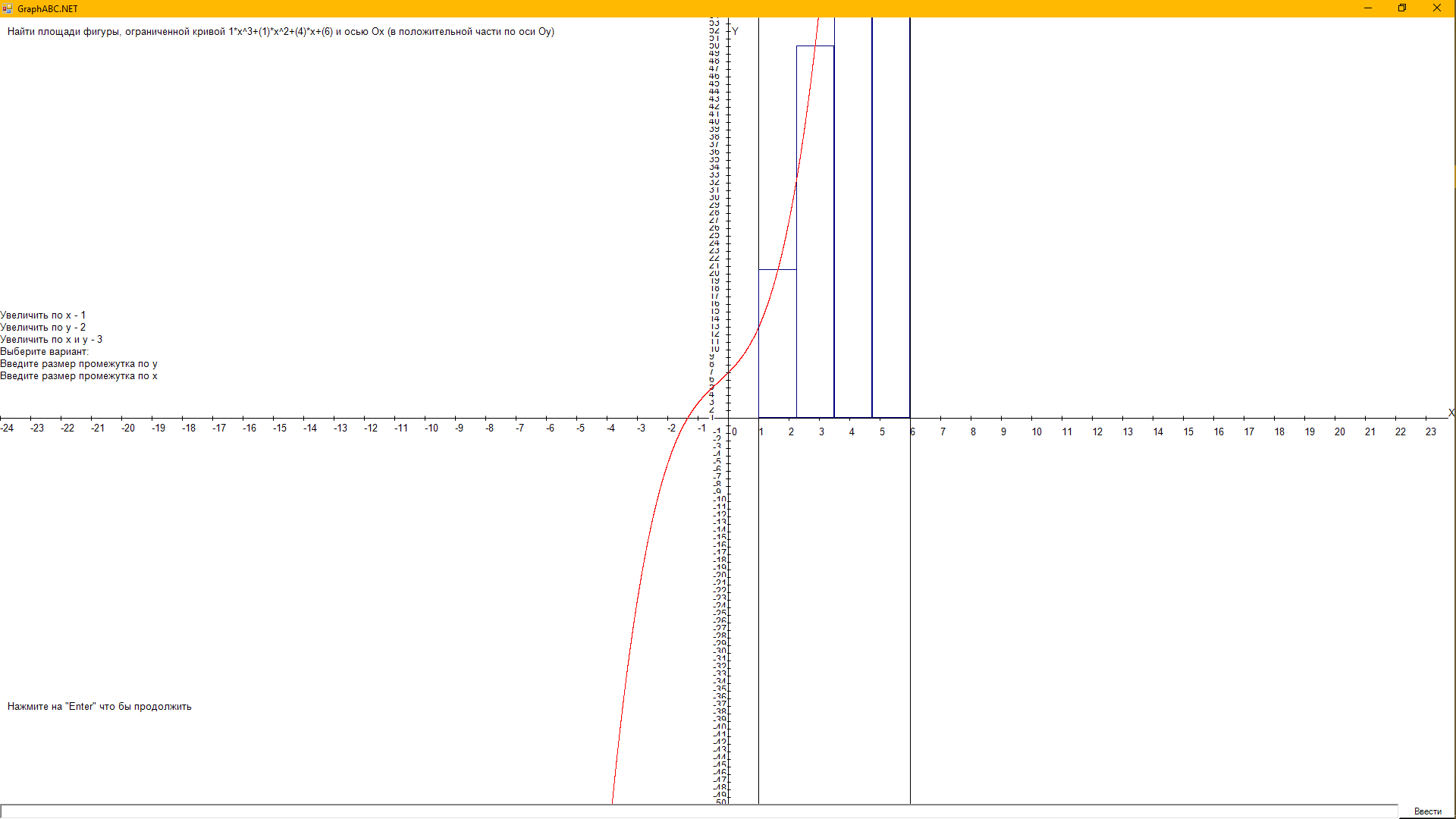


Рисунок 3 – результат программы

Вывод

В работе научился работать в графическом режиме и строить графики.

В программе график строится через пять переменных. Первые две переменные называются x0 и y0 они нужны для нахождение нулевой точки экрана. Из-за того, что экраны бывают разного разрешения поэтому моя программа, находит нулевую точку экрана. Следующие две переменные называются mx и my они нужны для масштаба графика, то есть для нахождения расстояние в пикселях от нулевой точки до единичной. При изменении эти переменных меняется весь график, то есть масштаб графика. Последняя переменная называется b она нужна для того чтоб определить сколько подписей на осях будет. Обычна эта переменная не трогалась так как при изменении её график выглядел либо сжато, либо программа вообще не работала.

В этой программе возникли трудности:

Были трудности с построения графика так как вообще не представлял, как можно было построить график. Так же возникли ошибки с выводом прямоугольников.

Были трудности с построением кривой линии так как опять же не представлял, как можно было построить кривую линию.