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

Криницын Олег Сергеевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2022

1. **Цель домашней контрольной работы:**

Освоить принципы работы в графическом режиме; получить базовые навыки взаимодействия с графическими примитивами.

1. **Формулировка задания:**

Вариант 11.

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.
5. **Код программы**

**uses** graphabc;

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

**begin**

func := 1 \* x \* x \* x + (-1) \* x \* x + (-4) \* x + 17;

**end**;

**var** i,n: integer;

x1, y1, x0, y0,downx,upx,step,o,y1h,x1h: real;

**var**

y3, y4: real;

sq: real;

Efn: real;

Ef: real;

N1: integer;

x3: real;

**begin**

Writeln('Введите нижнюю границу интегрирования:');

Readln(downx);

Writeln('Введите верхнюю границу интегрирования:');

Readln(upx);

Writeln('Введите шаг интегрирования:');

Readln(step);

writeln('Введите масштаб');

readln(n);

ClearWindow;

N1 := 0;

sq := 0;

x3 := downx;

Ef := step \* step \* step;

**repeat**

inc(N1);

y3 := func(x3);

x3 := x3 + step;

y4 := func(x3);

sq := sq + ((y3 + y4) / 2) \* step - Ef;

**until** x3 > upx;

SetFontColor(clred);

Writeln('Площадь вычисляемой фигуры равна: ', sq:0:4);

Efn := N1 \* step \* step \* step;

Writeln('Погрешность: ', Efn:0:4);

setwindowsize (800,550);

SetFontColor(clred);

SetFontSize(10);

textout (30,30,'1 \* x \* x \* x + (-1) \* x \* x + (-4) \* x + 17');

setpencolor (clblack);

setpenwidth (1);

line (80,400,720,400);

line (400,30,400,520);

SetFontSize(8);

**for** i:=-15 **to** 15 **do**

**begin**

line (400+i\*n,395,400+i\*n,405);

textout (395+i\*n,410, floattostr (i));

**end**;

textout (720,420, 'X');

**for** i:=-18 **to** 5 **do**

**begin**

line (395,400+i\*n,405,400+i\*n);

textout (370,400+i\*n, '');

textout (375,395+i\*n, floattostr (-i\*(10)));

**end**;

textout (420,20, 'Y');

downx:=400+(downx\*n);

upx:=400+(upx\*n);

step:=n\*step;

o:=downx;

**for** i:=-300 **to** 300 **do**

**begin**

x1:=i/n;

y1:= 1 \* x1 \* x1 \* x1 + (-1) \* x1 \* x1 + (-4) \* x1 + 17;

line(trunc(x0), trunc(y0), trunc(400+x1\*n), trunc(400-y1\*(n/10)), clRed);

**if** trunc(400+x1\*n)=downx **then** line(trunc(400+x1\*n), trunc(400-y1\*(n/10)), trunc(downx), 400, clred);

**if** (trunc(400+x1\*n)=o) **and** (o<upx) **then**

**begin**

line(trunc(400+x1\*n), trunc(400-y1\*(n/10)), trunc(o), 400, clred);

x1h:=(i+step)/n;

y1h:=1 \* x1h \* x1h \* x1h + (-1) \* x1h \* x1h + (-4) \* x1h + 17;

line(trunc(400+x1h\*n),trunc(400-y1h\*(n/10)), trunc(400+x1\*n), trunc(400-y1\*(n/10)),clred);

o:=o+step;

**end**;

**if** trunc(400+x1\*n)=upx **then** line(trunc(400+x1\*n), trunc(400-y1\*(n/10)), trunc(upx), 400, clred);

x0:=trunc(400+x1\*n);

y0:=trunc(400-y1\*(n/10));

**end**;

**end**.

1. **Результат выполнения программы**

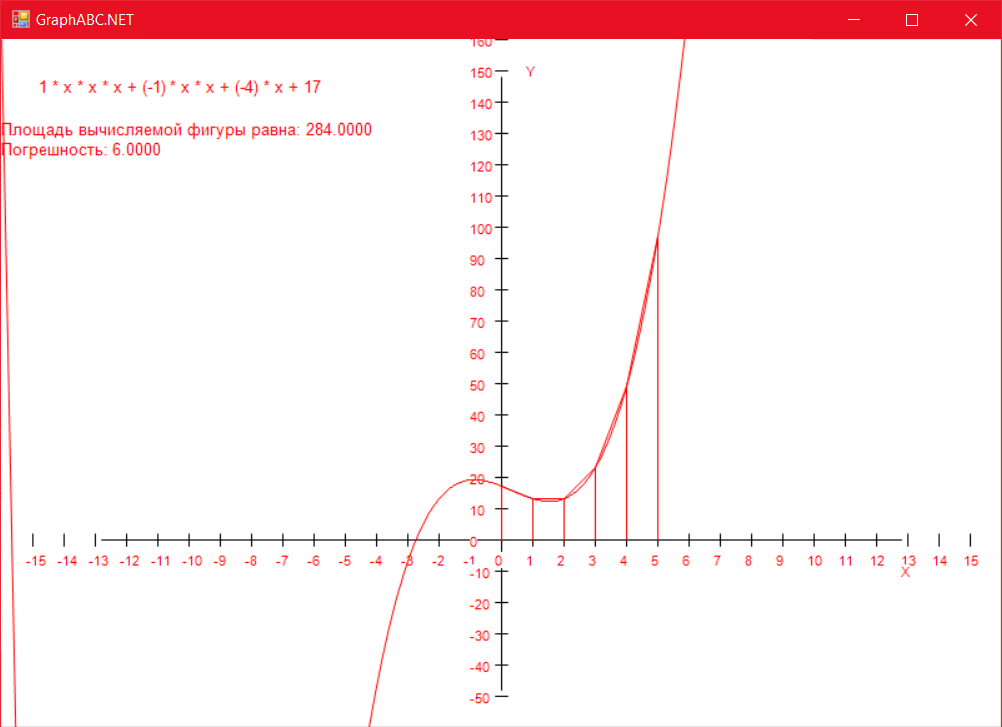
****

Рисунок 1 - Результат выполнения программы

1. **Вывод**

Нами была поставлена цель работы: Освоить принципы работы в графическом режиме; получить базовые навыки взаимодействия с графическими примитивами. Работа была разделена на несколько этапов: цель, формулировка задания, написание кода программы, результат выполнения программы, вывод. В коде программы используются модуль graphabc и функция func(x1). GraphABC - модуль для рисования в PascalABC.Net. Процедуры и функции для рисования примитивов. В GraphABC существует множество процедур и функций для рисования линий, прямоугольников, ломаных и текста. func(x1) - в этой функции мы присваиваем func (1 \* x^3 + (-1) \* x^2 + (-4) \* x + 17). В коде программы, в первую очередь, мы вводим верхнюю, нижнюю границу и шаг интегрирования, а также масштаб. Далее мы считаем площадь фигуры и погрешность. В конце, рисуем график и штриховку. Код программы был написан на PascalABC. Итак, делая вывод, хочу сказать, что в ходе работы все цели были достигнуты.