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-202-52-00

Коршунов Даниэль Александрович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2022

1. **Цель работы:**

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

**2. Задание (Вариант 7):**

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**

**Uses** GraphABC;

**var**

a, b, h, x, s, p, s1, y: real;

n, z: integer;

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

**begin**

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

**end**;

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

**begin**

ft := 0.5 \* power(x, 4) + 1 / 3 \* power(x, 3) - 2 \* power(x, 2) + x;

**end**;

**procedure** proc1;

**var**

i: integer;

**begin**

ClearWindow;

h := (b - a) / n;

x := a + (h / 2);

**for** i := 0 **to** n - 1 **do**

**begin**

s += f(x);

x += h;

**end**;

s \*= h;

TextOut(10, 10, 'Приближённая площадь фигуры = ');

TextOut(10, 30, s);

Readln;

**end**;

**procedure** proc3;

**var**

i: integer;

**begin**

ClearWindow;

s1 := ft(b) - ft(a);

TextOut(10, 10, 'Точная площадь фигуры = ');

TextOut(10, 30, abs(s1));

Readln;

**end**;

**procedure** proc4;

**var**

i: integer;

**begin**

ClearWindow;

h := (b - a) / n;

x := a + (h / 2);

**for** i := 0 **to** n - 1 **do**

**begin**

s += f(x);

x += h;

**end**;

s \*= h;

s1 := ft(b) - ft(a);

TextOut(10, 10, 'Погрешность = ');

TextOut(10, 30, abs(s - s1));

Readln;

**end**;

**procedure** proc2;

**begin**

ClearWindow;

TextOut(10, 10, 'введите область итеграции(>1)');

ReadLn(a, b);

TextOut(10, 30, 'введите количество промежутков');

ReadLn(n);

Readln;

**end**;

**procedure** proc0;

**begin**

ClearWindow;

TextOut(10, 10, 'Эта программа вычисляет площадь фигуры.');

Readln;

**end**;

**procedure** g;

**var**

gx, gy, x0, y0, k, x1, x2, y1, i: integer;

x, y: real;

**begin**

ClearWindow;

TextOut(10, 10, 'Введите маштаб:');

readln(k);//масштаб

ClearWindow;

gx := 500;

gy := 500;

setwindowsize(gx, gy);

x0 := gx **div** 2; // центр х

y0 := gy **div** 2; //центр у

//ось х

line(0, y0, gx, y0);

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

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

textout(gx - 10, y0 + 10, 'x');

//ось у

line(x0, 0, x0, gy);

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

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

textout(x0 + 10, 10, 'y');

//цикл для черточек х

i := -20;

**while** i < 21 **do**

**begin**

line(x0 + i \* k, y0 + 5, x0 + i \* k, y0 - 5);

textout(x0 + i \* k, y0 + 15, inttostr(i));

i += 1

**end**;

//черточки у

i := -20;

**while** i < 21 **do**

**begin**

**if** i <> 0 **then begin**

line(x0 + 5, y0 + i \* k, x0 - 5, y0 + i \* k);

textout(x0 + 10, y0 - i \* k, inttostr(i));

**end**;

i += 1;

**end**;

x := -5;

**while** x < 5 **do**

**begin**

y := f(x);

x1 := X0 + round(x \* k);//координата х отн центра

y1 := y0 - round(y \* k);//координата у отн центра

setpencolor(clred);

SetPixel(x1, y1, clred);

X += 0.0001;

**end**;

setpencolor(clBlue);

line(X0 + round(a \* k), y0 - round(f(a) \* k), X0 + round(a \* k), 250);

line(X0 + round(b \* k), y0 - round(f(b) \* k), X0 + round(b \* k), 250);

h := (b - a) / n;

x := a + (h / 2);

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

**begin**

y := f(x);

x1 := X0 + round((x - h / 2) \* k);//координата х отн центра

x2 := X0 + round((x + h / 2) \* k);

y1 := y0 - round(y \* k);//координата у отн центра

setpencolor(clBlack);

Rectangle(x1, y1, x2, 250);

X += h;

**end**;

Readln;

**end**;

**begin**

**repeat**

ClearWindow;

TextOut(10, 10, 'формулировка -1');

TextOut(10, 30, 'ввод a,b,n - 2');

TextOut(10, 50, 'приближённое вычисление - 3');

TextOut(10, 70, 'точное вычисление - 4');

TextOut(10, 90, 'погрешность - 5');

TextOut(10, 110, 'график - 6');

TextOut(10, 130, 'Exit - 0');

ReadLn(z);

**case** z **of**

1: proc0;

2: proc2;

3: proc1;

4: proc3;

5: proc4;

6: g;

**end**;

**until** z = 0;

**end**.

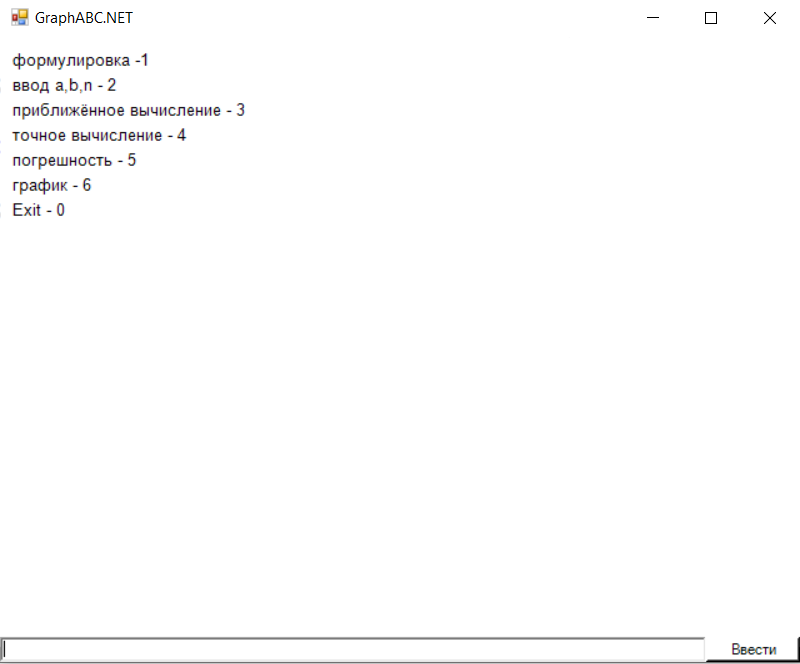
**4. Результат выполнения программы:**

Рисунок 2 – Главное меню программы

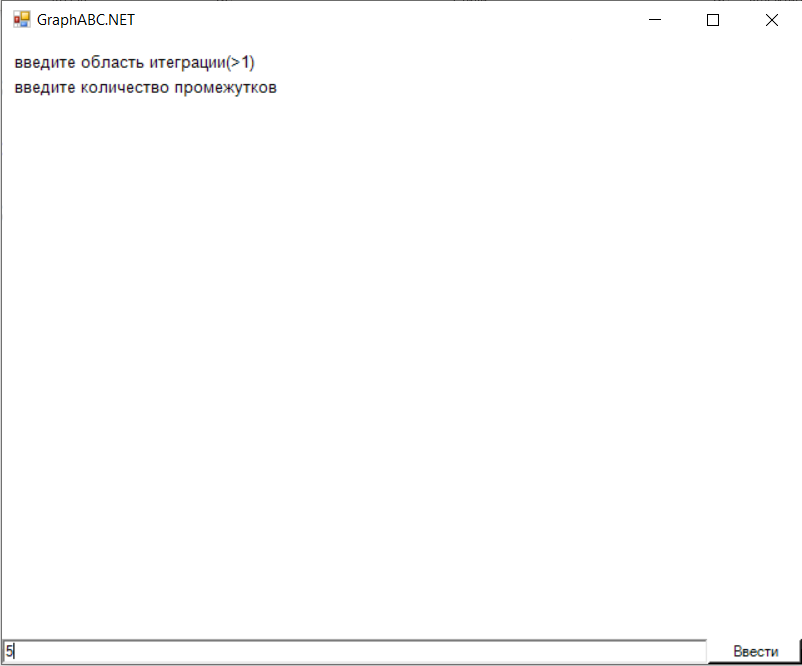
****

Рисунок 3 – Ввод данных

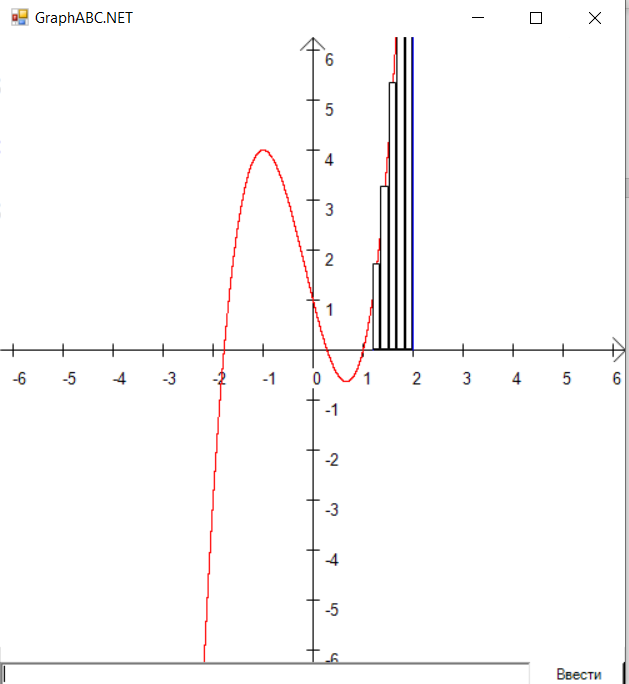
****

Рисунок 4 – График функции

**5. Вывод:**

Это задание показалось трудным. Несмотря на то, что надо было дополнить существующий код, было тяжело понять структуру кода и то, как его правильно дополнить. Было достаточно много трудностей. Основная трудность заключалась в том, что надо было понять, как правильно сделать график функции и правильно его оформить. В основном были ошибки в исполнении масштабирования графика и выделение вычисляемой области. Но все ошибки были исправлены и задние выполнено. Для задания было сделано масштабирование и визуализация расчета вычисляемого интеграла. Вся работа была выполнена в Pascal. В процессе выполнения данного задания было получено много опыта работы с графиками и работы в графическом режиме pascal.