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-204 – 52 - 00

Седов Павел Алексеевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2022

В отчете должны отображаться:

1. Цель контрольной работы
2. Формулировка задания (с вариантом)
3. Схема алгоритма с комментариями
4. Код программы
5. Результат выполнения программы
6. Вывод
7. Цель контрольной работы:

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

1. Формулировка задания (с вариантом):

Вариант 17:

**1**. Дополнить программу, реализованную в ходе предыдущей лабораторной работы, режимом визуализации.

**2.** Предусмотреть возможность вывода кривой, ограничивающей фигуру, на координатною 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 счета интеграла.

1. Схема алгоритма с комментариями:

Разрешили не делать

1. Код программы

Задание 1-4:

**uses** graphABC;

**var**

y: real;

mashtX, mashtY, x0, y0, i1: integer;

**var**

l: byte;

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

**begin**

writeln('Да - 1');

writeln('Нет - 0');

Write('Выберите вариант: ');

readln(v);

vibor := v;

**end**;

**var**

l1: byte;

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

**begin**

textout(10, 130, 'Увеличить x - 1');

textout(10, 150, 'Увеличить y - 2');

textout(10, 170, 'Уменьшить x - 3');

textout(10, 190, 'Уменьшить y - 4');

textout(10, 220, 'Закончить - 0');

textout(10, 250, 'Выберите вариант: ');

readln(v);

viborp := v;

**end**;

**var**

n, i: integer;

a, b, h, x, s, f, d: real;

**function** f0(**var** x: real): real;//основная

**begin**

**var** f1: real;

f1 := 1 \* power(x, 3) + (0) \* power(x, 2) + (5) \* x + (11);

f0 := f1;

**end**;

**function** f2(**var** x: real): real;//первообразная

**begin**

**var** f3: real;

f3 := 1 / 4 \* power(x, 4) + 5 / 2 \* power(x, 2) + 11 \* x;

f2 := f3;

**end**;

**var**

m: integer;

**function** graf(**var** p, n1: integer; **var** s1, s2: real): real; //График

**var**

i1, i, w, l, w1: integer;

p1, e: real;

**begin**

clearwindow;

MaximizeWindow;

x0 := windowwidth **div** 2;

y0 := windowheight **div** 2;

line(0, y0, windowwidth - 11, y0);

line(x0, 0, x0, windowheight - 11);

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

textout(windowwidth - 10, y0 - 20, 'x');

textout(x0 + 5, 0, 'y');

textout(10, 10, 'Найти площадь фигуры ограниченной кривой 1\*x^3+(0)\*x^2+(5)\*x+(11)');

**for** i := 1 **to** 100 **do**

**begin**

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

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

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

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

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

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

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

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

**end**;

setpencolor(clblue);

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

setpencolor(clgreen);

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

setpencolor(clblack);

l := round((x0 + round(s2 \* mashtX) - x0 - round(s1 \* mashtX)) / n1);

w := l;

w1 := 0;

p1 := (s2 - s1) / n1;

e := x;

x := a;

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

**begin**

setpixel(x0 + round(x \* mashtX), y0 - round(f0(x) \* mashtY), clred);

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

**begin**

setpencolor(clRed);

e := x;

**var** p2 := p1 / 2 + e;

Rectangle(x0 + round(s1 \* mashtX), y0 - round(f0(p2) \* mashtY), x0 + round(s1 \* mashtX + (w + l)), y0);

**end**;

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

**begin**

setpencolor(clRed);

e := x;

**var** p2 := p1 / 2 + e;

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

Rectangle(x0 + round(s1 \* mashtX + l), y0 - round(f0(p2) \* mashtY), x0 + round(s1 \* mashtX + (w + l)), y0);

l := l + w;

e := e + l;

inc(w1);

**if** (w1 + 1) = n1 **then**

**begin**

setpencolor(clBlack);

**break**;

**end**;

**end**;

x := x + 0.001;

**end**;

**for** i1 := -15000 **to** 15000 **do** //Линия

**begin**

x := 0.001 \* i1;

y := f0(x);

setpixel(x0 + round(mashtX \* x), y0 - round(mashtY \* y), clRed);

graf := p;

**end**;

**end**;

**function** masht: integer;//Маштабирование

**begin**

textout(10, 100, 'Увеличить?');

viborp(l1);

**case** l1 **of**

1:

**begin**

mashtX := mashtX + 10;

graf(m, n, a, b);

masht;

**end**;

2:

**begin**

mashtY := mashtY + 10;

graf(m, n, a, b);

masht;

**end**;

3:

**begin**

mashtY := mashtX - 10;

graf(m, n, a, b);

masht;

**end**;

4:

**begin**

mashtY := mashtY - 10;

graf(m, n, a, b);

masht;

**end**;

0:

**begin**

writeln('Ну ладно');

**exit**();

**end**;

**end**;

masht := 0

**end**;

**begin**

writeln('Найти площадь фигуры ограниченной кривой 1\*x^3+(0)\*x^2+(5)\*x+(11)');

writeln('Введите пределы интегрирования a и b при a<b');

readln(a, b);

**if** (a > b) **then begin** writeln('a не может быть больше b'); **exit**(); **end**;

writeln('Введите количество интервалов разбиения n');

readln(n);

h := (b - a) / n;

x := a + h;

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

**begin**

f := f0(x);

S := S + f;

x := x + h;

**end**;

S := (h \* (f0(a) + f0(b))) / 2 + S;

writeln('S= ', s);

writeln('Вывести погрешность?');

vibor(l);

mashtX := 20;

mashtY := 20;

**case** l **of**

1:

**begin**

d := abs((f2(b) - f2(a)) - S);

writeln('Погрешность= ', d);

**end**;

0:

**begin**

writeln('Ну и ладно');

**end**;

**end**;

writeln('Вывести график функций?');

vibor(l);

**case** l **of**

1:

**begin**

graf(m, n, a, b);

**end**;

0:

**begin**

writeln('Ну ладно');

**exit**();

**end**;

**end**;

masht;

**end**.

1. Результат выполнения программы

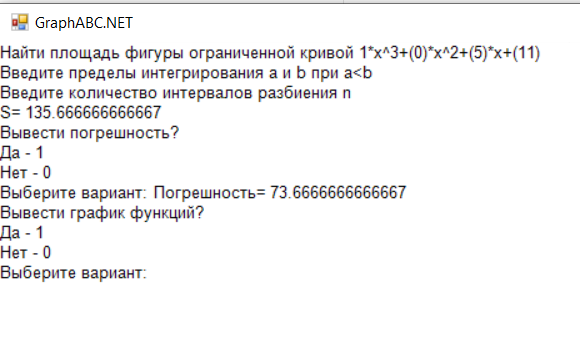


Рисунок 1 - Результат выполнения программы 1-4

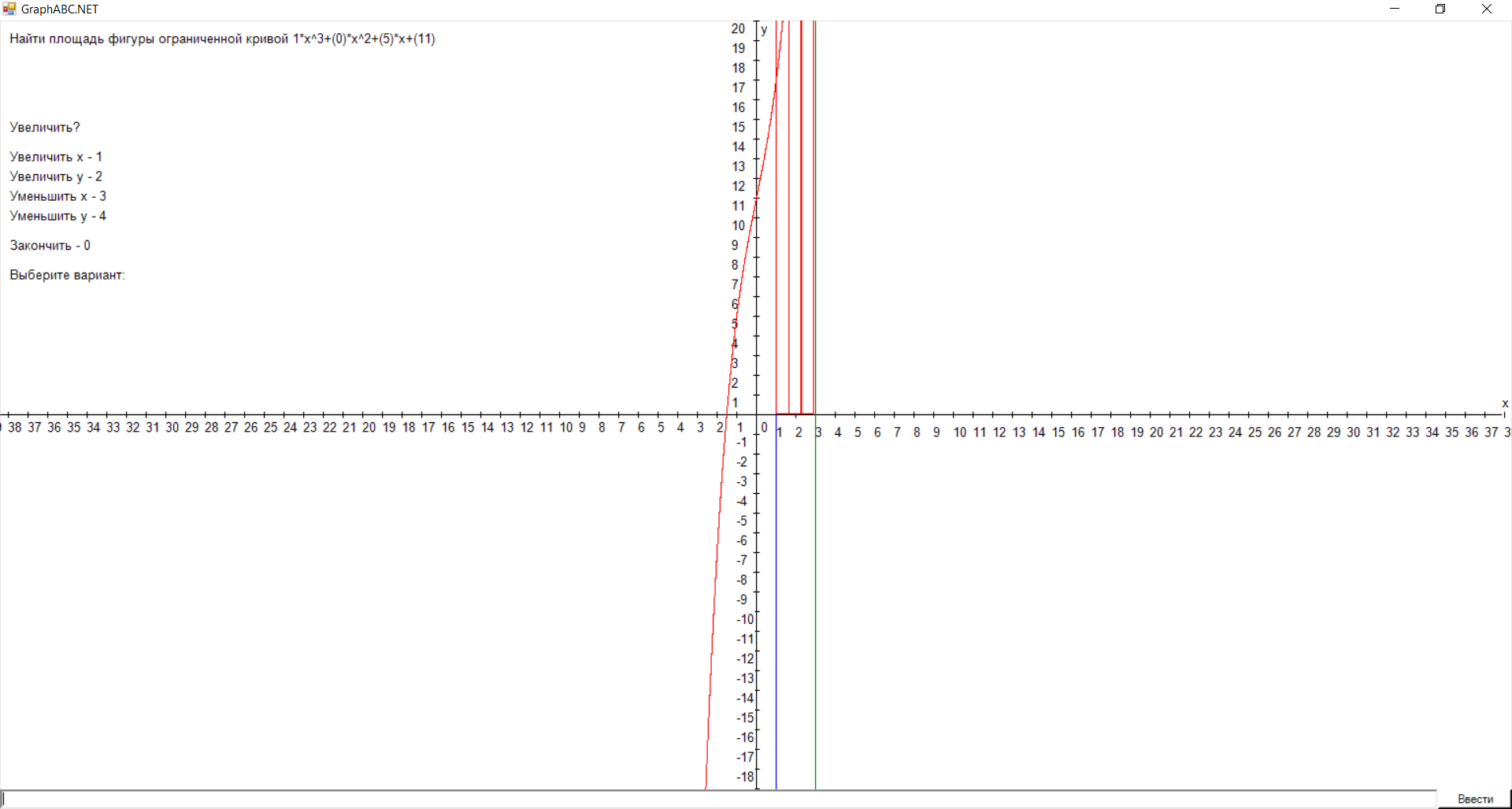


Рисунок 2 - Результат выполнения программы 1-4

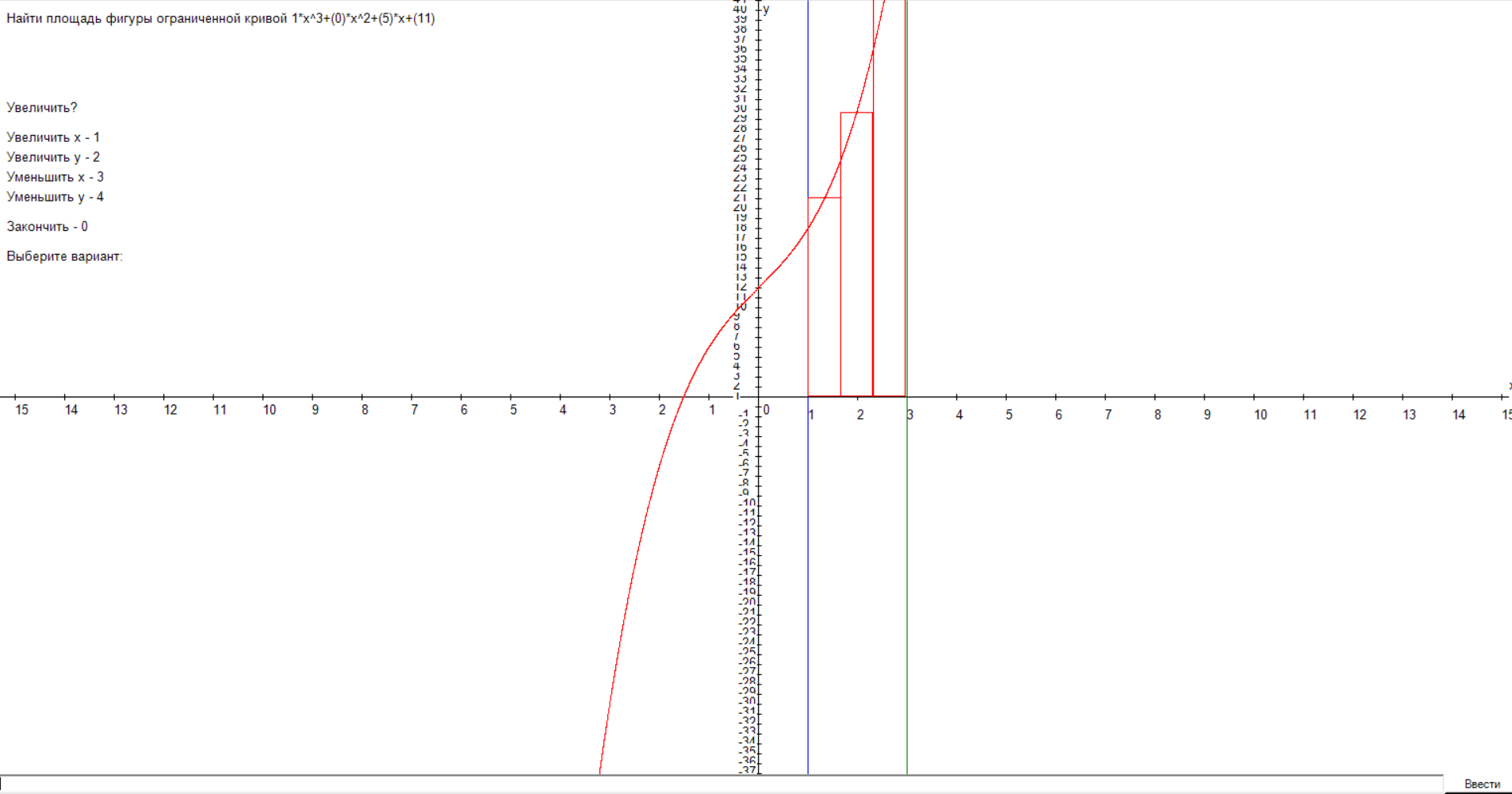


Рисунок 3 - Результат выполнения программы 1-4

1. Вывод

В этой работе было изучено построение графика функций в графическом режиме. С помощью программы нужно вычислить площадь фигуры, ограниченной кривой 1\*x^3+(0)\*x^2+(5)\*x+(11), использовав метод средних прямоугольников.

С самого начала для построения графика функций нужно сделать оси графика функций это делается через две переменные x0 и y0 они нужны для нахождения нулевой точки экрана. Из-за того, что экраны бывают разного разрешения поэтому эта программа, находит нулевую точку в середине экрана. Переменные mashtX и mashtY нужны для масштабирования графика функций по оси x и оси y, изначально по двум переменным стоят значения по 20, если пользователь захочет увеличить оси то ему будет дам выбор увеличить или уменьшить каждую ось на 10 нажав одну из представленных кнопок. Так же в есть переменные a и b которые нарисованы на графике и представляют собой ограничения по которым пользователь вычисляет площадь.

В этой программе возникли трудности:

- Визуализировать метод средних прямоугольников.

- Построение кривой линии т.к было не достаточно знаний как это сделать. По итогу было принято решение просто создать цикл где выводится очень много пикселей практически в упор друг к другу.