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 ПРОГРАММИРИВАНИЯ»**

Выполнил: студент учебной группы

ИСПк-203-52-00

Цветкова Кристина Андреевна

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

Содержание

1. Цель лабораторной работы
2. Формулировка задания (с вариантом)
3. Описание алгоритма
4. Схема алгоритма с комментариями
5. Код программы
6. Результат выполнения программы
7. Вывод
8. **Цель работы:** освоить принципы работы в графическом режиме, получить базовые навыки взаимодействия с графическими примитивами.
9. **Формулировка задания (Вариант:22)**

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.

1. **Описание алгоритма**

Данный алгоритм реализует метод Симпсона (парабол) для приближенного вычисления определенного интеграла функции.

1. Создаем функцию f(x), которая представляет собой кубическое уравнение.

2. Создаем функцию simpson (a, b, n), которая использует метод Симпсона для вычисления определенного интеграла от функции f(x) на отрезке [a, b] с использованием n подотрезков.

3. Пользователь вводит нижний предел a, верхний предел b и количество подотрезков n.

4. Вызываем функцию simpson (a, b, n) для вычисления значения определенного интеграла.

5. Устанавливаем размер окна графика и рисуем оси координат.

6. Рисуем график функции f(x) на отрезке [-10, 10].

7. Рисуем область под кривой, используя метод Симпсона и параболы, проходящие через три точки на каждом подотрезке.

8. Добавляем метки к осям координат.

9. Выводим значение определенного интеграла по методу Симпсона.

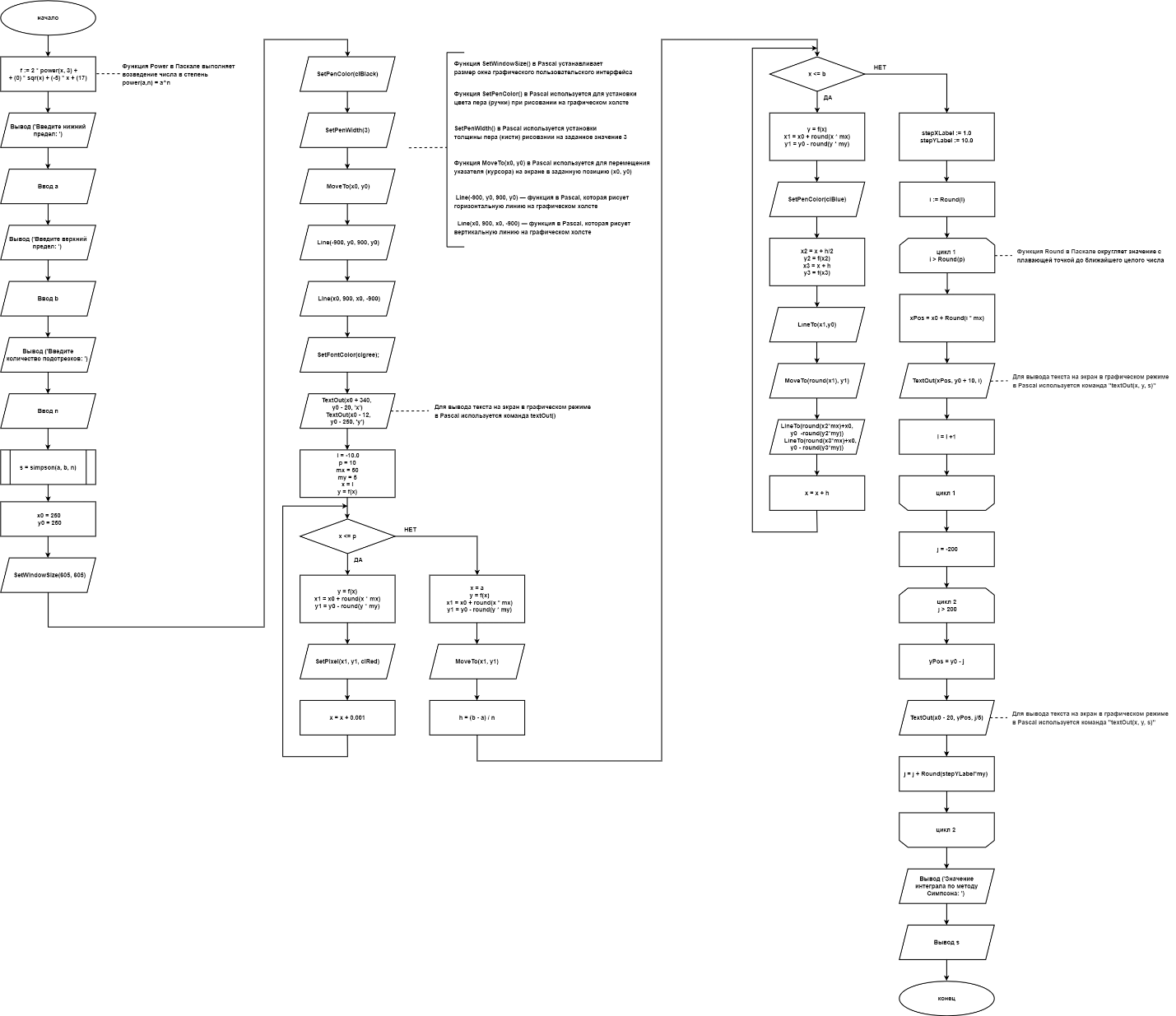
1. **Схема алгоритма с комментариями**

Рис 1. Схема алгоритма с комментариями

1. **Код программы**

**uses** graphABC;

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

**begin**

f := 2 \* power(x, 3) + (0) \* sqr(x) + (-5) \* x + (17);

**end**;

**function** simpson(a, b: real; n: integer): real;

**var**

h, x, s: real;

**begin**

h := (b - a) / n;

s := f(a) + f(b);

x := a + h;

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

**begin**

**if** i **mod** 2 = 0 **then**

s := s + 2 \* f(x)

**else**

s := s + 4 \* f(x);

x := x + h;

**end**;

simpson := s \* h / 3;

**end**;

**var**

x1 := 0;

y1 := 0;

**begin**

write('Введите нижний предел: ');

**var** a := ReadReal();

writeln(a);

write('Введите верхний предел: ');

**var** b := ReadReal();

writeln(b);

write('Введите количество подотрезков: ');

**var** n := ReadInteger();

writeln(n);

**var** s := simpson(a, b, n);

// Построение графика функции

**var** x0 := 250;

**var** y0 := 250;

SetWindowSize(605, 605);

// Нарисуйте оси X и Y

SetPenColor(clBlack);

SetPenWidth(3);

MoveTo(x0, y0);

Line(-900, y0, 900, y0);

Line(x0, 900, x0, -900);

// Нарисуйте метки осей

SetFontColor(clgreen);

TextOut(x0 + 340, y0 - 20, 'x');

TextOut(x0 - 12, y0 - 250, 'y');

//График функции рисования

**var** l := -10.0;

**var** p := 10;

**var** mx := 50;

**var** my := 5;

**var** x := l;

**var** y := f(x);

**while** x <= p **do**

**begin**

y := f(x);

x1 := x0 + round(x \* mx);

y1 := y0 - round(y \* my);

SetPixel(x1, y1, clRed);

x += 0.0001;

**end**;

// Нарисуйте область под кривой, используя параболы

x := a;

y := f(x);

x1 := x0 + round(x \* mx);

y1 := y0 - round(y \* my);

MoveTo(x1, y1);

**var** h := (b - a) / n;

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

**begin**

y := f(x);

x1 := x0 + round(x \* mx);

y1 := y0 - round(y \* my);

SetPenColor(clBlue);

// Нарисуйте параболу, используя три точки

**var** x2 := x + h / 2;

**var** y2 := f(x2);

**var** x3 := x + h;

**var** y3 := f(x3);

LineTo(x1, y0);

MoveTo(round(x1), y1);

LineTo(round(x2 \* mx) + x0, y0 - round(y2 \* my));

LineTo(round(x3 \* mx) + x0, y0 - round(y3 \* my));

// Перейти к следующему интервалу

x += h;

**end**;

// Нарисуйте метки осей

**var** stepXLabel := 1.0; // Step for X axis labels

**var** stepYLabel := 10.0; // Step for Y axis labels

// Метки по оси X

**for var** i := Round(l) **to** Round(p) **do**

**begin**

**var** xPos := x0 + Round(i \* mx);

TextOut(xPos, y0 + 10, i);

**end**;

// Метки по оси Y

**for var** j := -200 **to** 200 **step** Round(stepYLabel\*my) **do**

**begin**

**var** yPos := y0 - j;

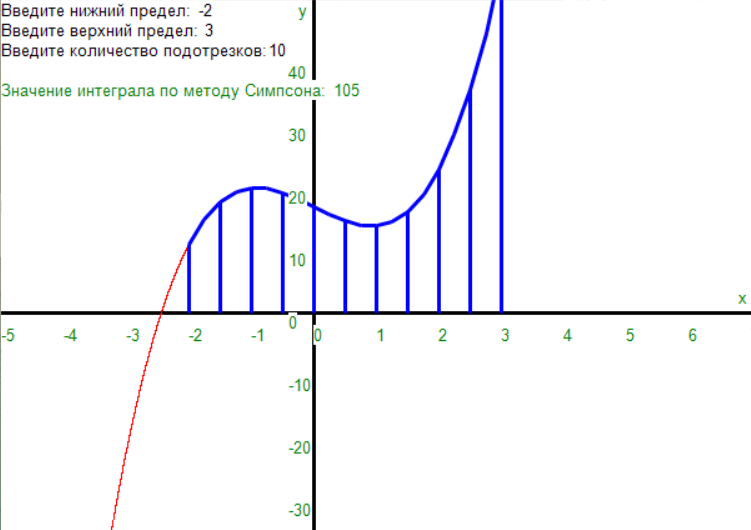
TextOut(x0 - 20, yPos, j / 5);

**end**;

writeln('');

writeln('Значение интеграла по методу Симпсона: ', s);

**end**.

1. **Результат выполнения программы**

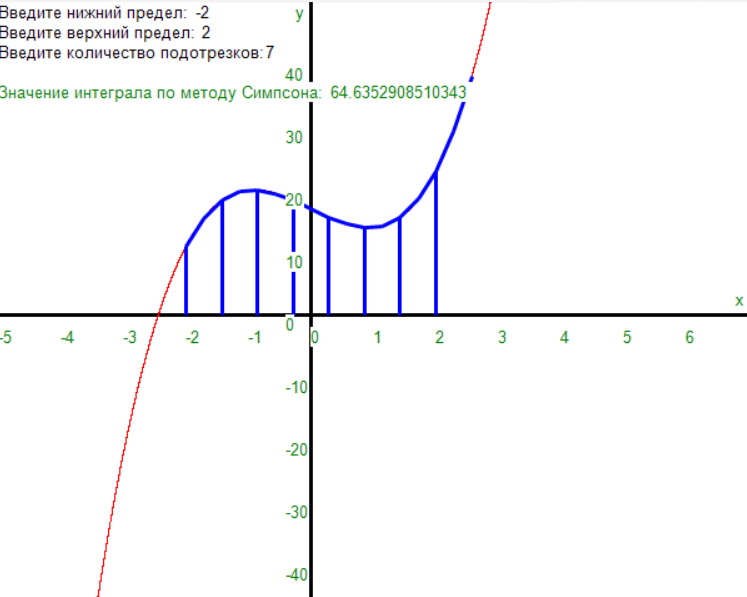
Рис.2. Результат выполнения программы

Рис.3. Результат выполнения программы

**Вывод**

В процессе выполнения домашней контрольной работы, мы узнали множество новых операций на языке Паскаль, как например, модуль GraphABC.

В языке программирования Паскаль (Pascal), GraphABC — это модуль программирования, который предоставляет пользователю простой интерфейс для создания и отображения графических объектов, таких как точки, линии, фигуры и т.д. Этот модуль позволяет пользователю создавать и изменять графики, используя программный код, без необходимости знания сложных математических формул или графических алгоритмов. Модуль GraphABC облегчает процесс создания визуальных элементов и позволяет пользователям визуализировать данные и создавать графические приложения. К тому же, мы узнали множество процедур и функций.

В процессе выполнения работы, для создания блок схемы мы познакомились с программой Draw io. Draw.io — инструмент для создания диаграмм, схем алгоритмов, интеллект-карт, бизнес-макетов, отношений сущностей, программных блоков и другого. Draw.io обладает богатым набором функций для визуализации большинства задач пользователя.