Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №3**

**«Изучение базовых принципов организации процедур и функций»**

**ПО ДИСЦИПЛИНЕ «Основы алгоритмизации и программирование»**

Выполнил: студент учебной группы

ИСПк-203-52-00

Четвериков Д. А.

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

1. **Цель работы**

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

**2. Формулировка задания:** Вариант 0.

Задание:

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.

**Описание алгоритма:**

**1. Определение функции:**

Задайте функцию, площадь под графиком которой нужно вычислить. Например, в вашем коде это f(x) = 2\*x^3-x^2-x+16.

**2. Ввод параметров интегрирования:**

* Получите от пользователя нижний предел (a) и верхний предел (b) интегрирования.
* Определите количество подынтервалов (n), на которые будет разбран отрезок [a, b].

**3. Вычисление ширины подынтервала:**

Расчитайте ширину каждого подынтервала (h) по формуле: h=(b-a)/n.

**4. Инициализация суммы:**

Установите переменную `sum` равной 0, которая будет хранить сумму значений функции на выбранных точках.

**5. Цикл по подынтервалам:**

Используйте цикл от 1 до n:

* Рассчитайте x — точку, соответствующую правым граням подынтервалов: x = a + i \* h, где i — это индекс текущего подынтервала.
* Вычислите значение функции в этой точке: f(x).
* Добавьте полученное значение к сумме.

**6. Вычисление площади:**

После завершения цикла перемножьте сумму значений функции на ширину подынтервала: area = h \* sum. Это и будет итоговая площадь под кривой.

**7. Оценка погрешности:**

Вычислите оценку погрешности. Для этого определите максимальное значение второй производной функции на интервале [a, b] и используйте формулу для расчета погрешности: err = (b - a)^5 / (180 \* n^4) \* sdmax.

**8. Вывод результатов:**

Отобразите вычисленную площадь и оценку погрешности пользователю. Если пользователь выбирает в case меню 2, то происходит визуализация интегрирования по методу правых прямоугольников на экране.

**9. Проверка на выход:**

Позвольте пользователю повторить процесс или выйти из программы.

**Схема алгоритма:**

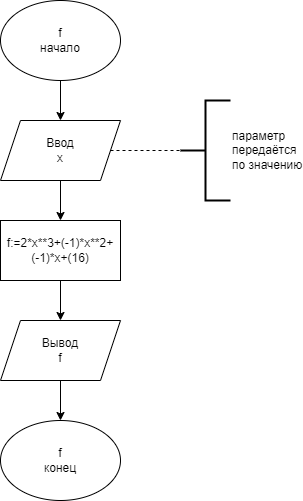


Рисунок 1 – схема алгоритма для функции f

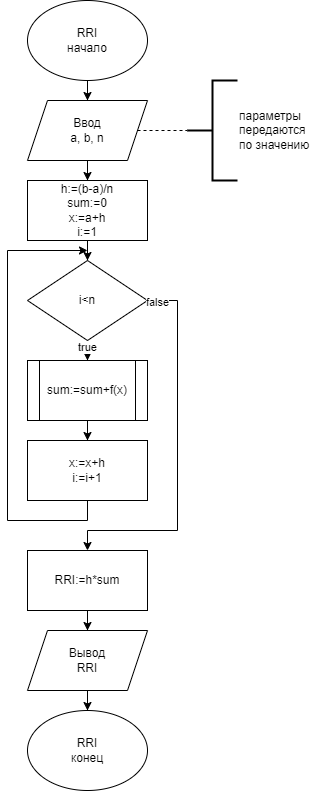


Рисунок 2 – схема алгоритма для функции RRI

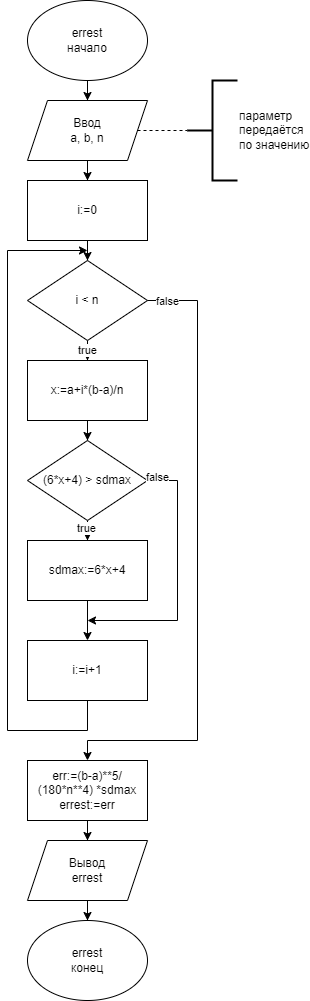


Рисунок 3 – схема алгоритма для функции errest

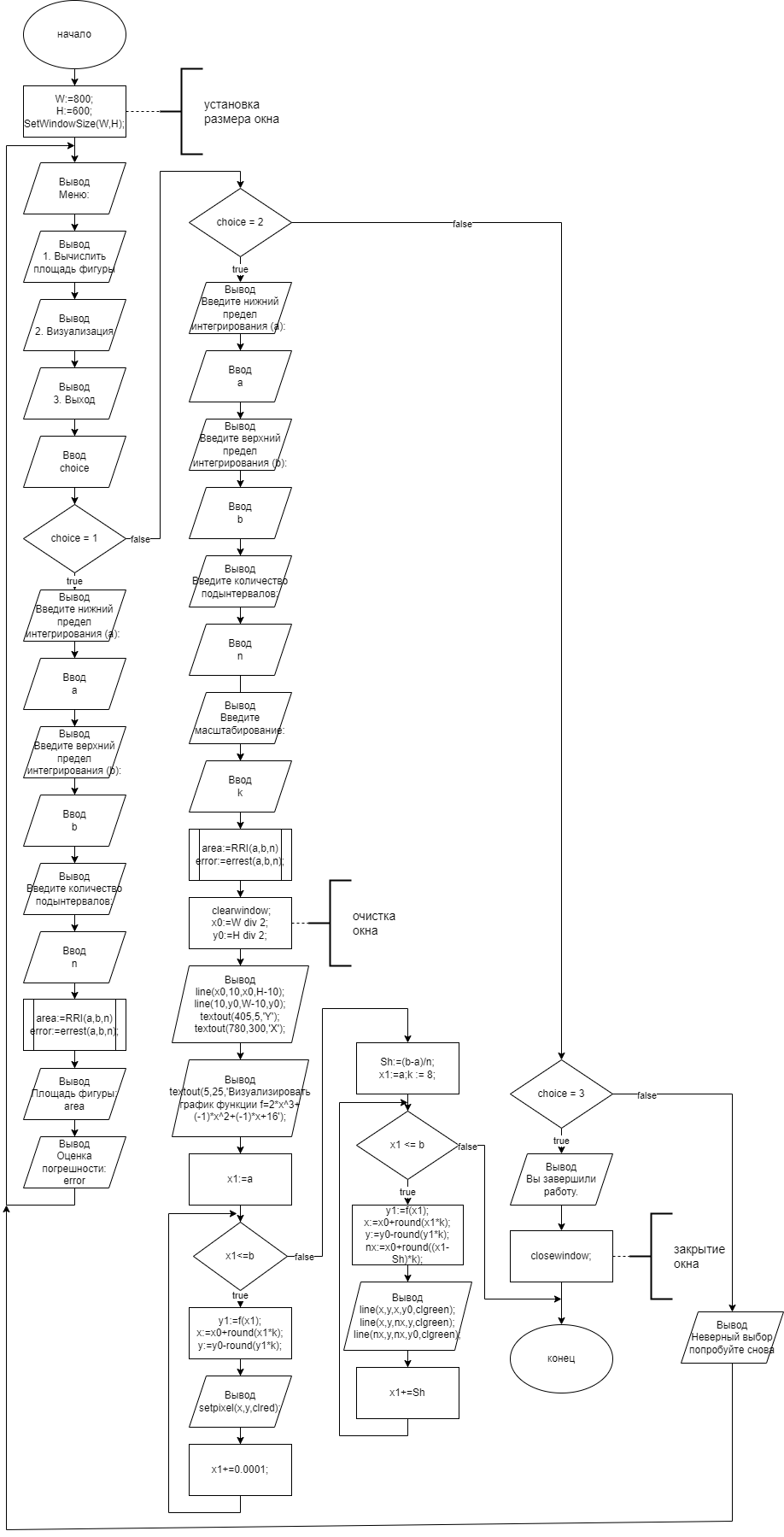


Рисунок 4 – схема алгоритма ввода и вывода результата

**Код программы:**

**uses** graphABC;

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

**begin**

f := 2 \* x \*\* 3 + (-1) \* x \*\* 2 + (-1) \* x + (16);

**end**;

**function** RRI(a, b: Real; n: Integer): Real;

**var**

h, x, sum: real;

i: integer;

**begin**

h := (b - a) / n;

sum := 0;

x := a + h;

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

**begin**

sum := sum + f(x);

x := x + h;

**end**;

RRI := h \* sum;

**end**;

**function** errest(a, b: Real; n: Integer): Real;

**var**

sdmax, x, err: Real;

i: Integer;

**begin**

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

**begin**

x := a + i \* (b - a) / n;

**if** (6 \* x + 4) > sdmax **then** // первообразная 2-ого порядка от начальной функции

sdmax := 6 \* x + 4;

**end**;

err := (b - a) \*\* 5 / (180 \* n \*\* 4) \* sdmax; // Формула для вычисления погрешности

errest := err;

**end**;

**var**

x1,y1,a,b,k,Sh:real;

W,H,x,y,x0,y0,n,nx:integer;

choice: Char;

area, error: Real;

**begin**

W:=800;

H:=600;

SetWindowSize(W,H);

**repeat**

writeln('Меню:');

writeln('1. Вычислить площадь фигуры');

writeln('2. Визуализация');

writeln('3. Выход');

readln(choice);

**case** choice **of** '1':

**begin**

writeln('Введите нижний предел интегрирования (a): ');

readln(a);

writeln('Введите верхний предел интегрирования (b): ');

readln(b);

writeln('Введите количество подынтервалов: ');

readln(n);

area := RRI(a, b, n);

error := errest(a, b, n);

writeln(' Площадь фигуры: ', area:0:6);

writeln(' Оценка погрешности: ', error:0:6);

**end**;

'2':

**begin**

writeln('Введите нижний предел интегрирования (a).');

readln(a);

writeln('Введите верхний предел интегрирования (b).');

readln(b);

writeln('Введите количество подынтервалов.');

readln(n);

writeln('Введите масштабирование ');

readln(k);

area := RRI(a, b, n);

error := errest(a, b, n);

clearwindow;

x0:=W **div** 2;

y0:=H **div** 2;

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

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

textout(405,5,'Y');

textout(780,300,'X');

textout(5,25,'Визуализировать график функции f=2\*x^3+(-1)\*x^2+(-1)\*x+16');

x1:=a;

**while** x1 <= b **do**

**begin**

y1:=f(x1);

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

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

setpixel(x,y,clred);

x1+=0.0001;

**end**;

Sh:=(b-a)/n;

x1:=a;k := 8;

**while** x1 <= b **do**

**begin**

y1:=f(x1);

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

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

nx:=x0+round((x1-Sh)\*k);

line(x,y,x,y0,clgreen);

line(x,y,nx,y,clgreen);

line(nx,y,nx,y0,clgreen);

x1+=Sh;

**end**;

**end**;

'3':

**begin**

writeln('Вы завершили работу.');

sleep(1000);

closewindow;

**end**

**else**

writeln('Неверный выбор. Попробуйте снова.');

**end**;

**until** (choice = '3') **or** (choice = '2');

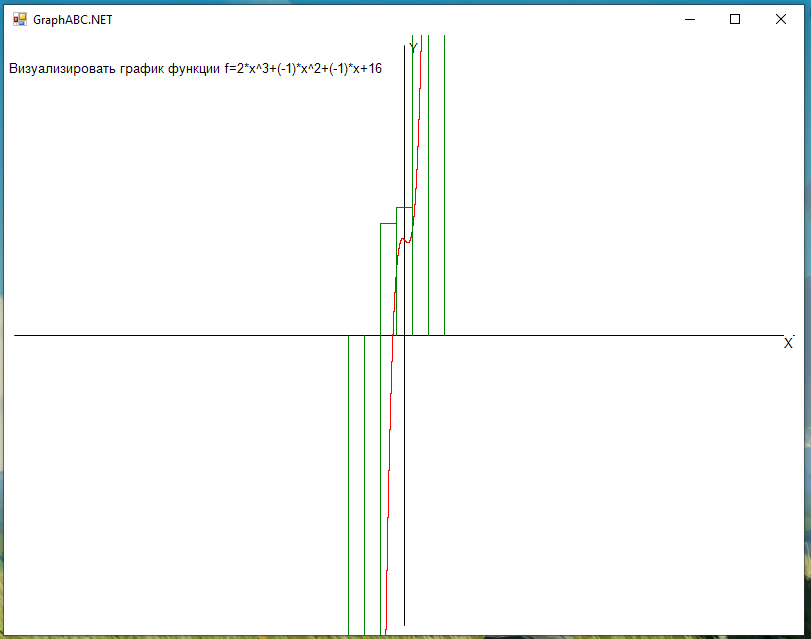
**end**. 

Рисунок 6 – результат выполнения программы

**Вывод:** в ходе работы на языке Pascal, направленной на освоение принципов работы в графическом режиме и получение базовых навыков взаимодействия с графическими примитивами, были достигнуты важные результаты.

Во-первых, удалось освоить основные функции и процедуры работы с графикой в Pascal, что позволило понять, как организована графическая среда, как устанавливать графический режим и управлять им.

Во-вторых, работа с графическими примитивами, такими как линии, круги, прямоугольники и текст, позволила закрепить навыки рисования и визуализации информации, что является основополагающим для создания простых графических приложений.

В-третьих, изучение взаимодействия с графическими элементами и их свойствами, такими как цвет, координаты и размеры, дало возможность развить навыки алгоритмического мышления и улучшить понимание логики программирования.

Таким образом, выполненная работа положила начало для дальнейшего углубленного изучения графики в Pascal и подготовки к более сложным проектам, связанным с разработкой графических интерфейсов и игр. Полученные знания и навыки будут полезны для дальнейшего освоения более продвинутых технологий программирования и дизайна.