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 программирование»**

Выполнил: студент учебной группы

ИСПк-203-52-00

Макаров Никита Андреевич

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+(0)\*x\*\*2+(3)\*x+(4)

**2. Ввод параметров интегрирования:**

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

**3. Вычисление ширины подынтервала:**

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

**4. Проверка четности:**

Необходимо проверить является ли число четным или нечетным посредством деления на 2

**5. Цикл по подынтервалам:**

Используйте цикл от 1 до n-1:

* Рассчитайте int — точку, соответствующую правым граням подынтервалов: int=f(a)+f(b) — это индекс текущего подынтервала.
* Вычисление суммы членом с четными/нечетными индексами
* Вычислите значение функции в этой точке: f(x).

**6. Вычисление площади:**

После завершения цикла необходимого подынтервала: area = h /3. Будет найдена 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: error= (b - a)^5 / (180 \* n^4) \* sdmax.

**8. Построение графика**

**Запросить у пользователи нижний, верхний предел, подынтервал, масштабирование**

**9. Проверка на выход:**

Позвольте пользователю повторить процесс или выйти из программы.

**Схема алгоритма:**

Изображение выглядит как диаграмма, текст, белый, зарисовка

Автоматически созданное описание

Рисунок 1 – схема алгоритма для функции f

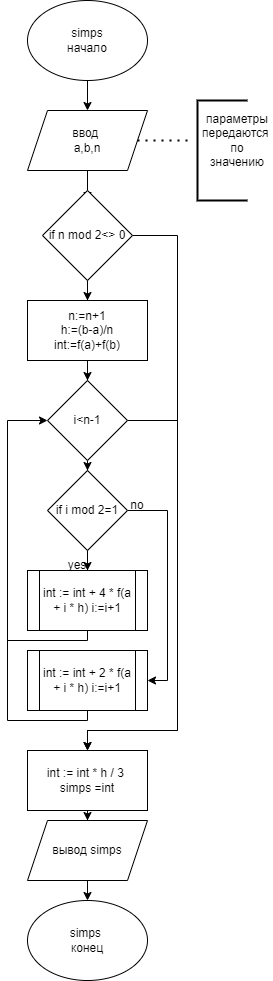


Рисунок 2 – схема алгоритма для функции Simps

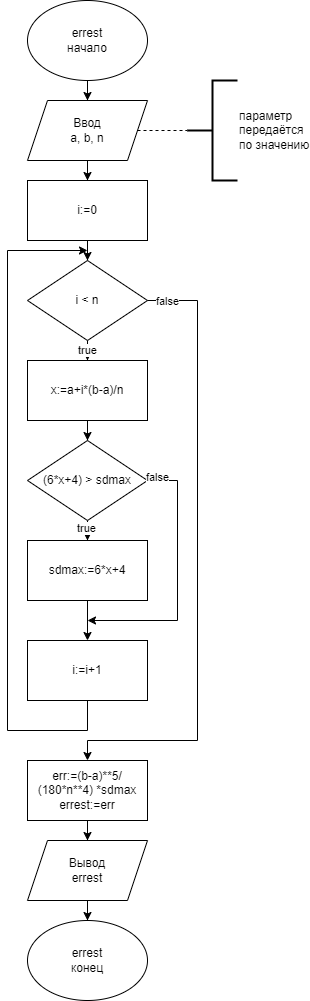


Рисунок 3 – схема алгоритма для функции errest

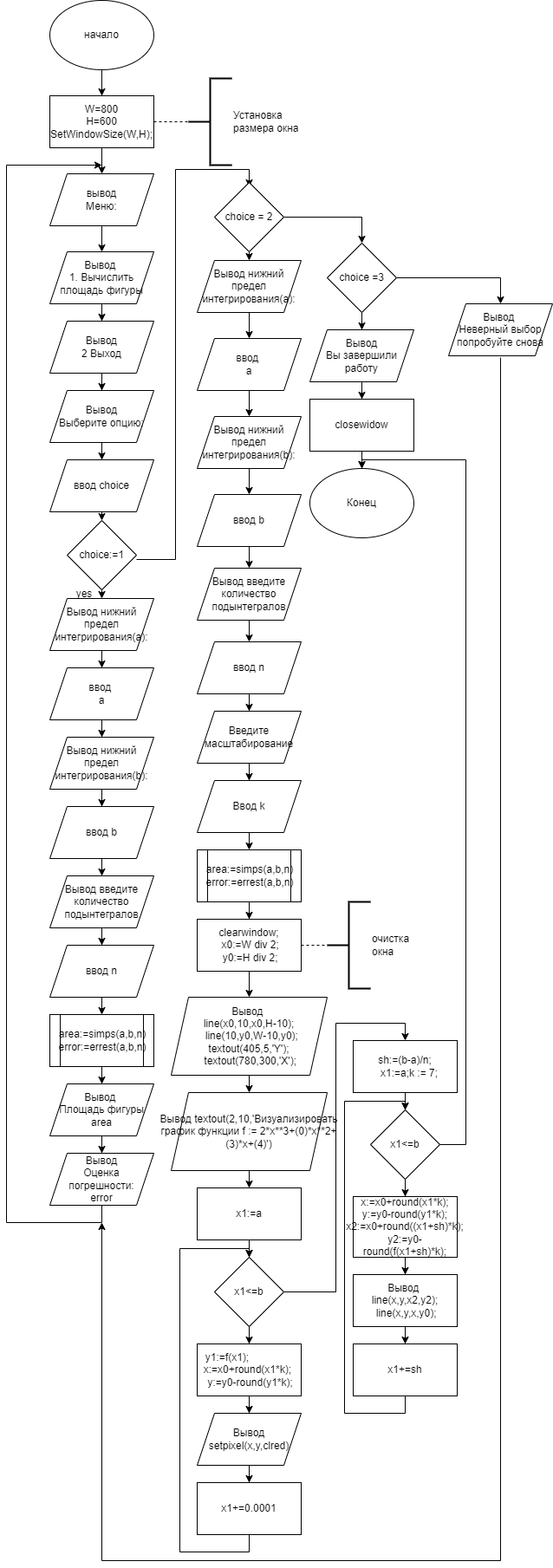


Рисунок 4 – схема алгоритма ввода и вывода результата

**Код программы:**

**uses GRAPHabc;**

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

**begin**

**f := 2\*x\*\*3+(0)\*x\*\*2+(3)\*x+(4);**

**end;**

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

**var**

**h, int: Real;**

**i: Integer;**

**begin**

**if n mod 2 <> 0 then**

**n := n + 1; // Если n нечетное, увеличиваем на 1**

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

**int := f(a) + f(b);**

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

**begin**

**if i mod 2 = 1 then**

**int := int + 4 \* f(a + i \* h) // сумма членов с нечетными индексами**

**else**

**int := int + 2 \* f(a + i \* h); // сумма членов с чётными индексами**

**end;**

**int := int \* h / 3;**

**simps := int;**

**end;**

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

**var**

**sdmax, x, error: 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;**

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

**errest := error;**

**end;**

**var**

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

**W,H,x,y,x0,y0,n,y2,x2: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 := simps(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 := simps(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(2,10,'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 := 7;**

**while x1 <= b do**

**begin**

**y1:=f(x1);**

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

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

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

**x2:=x0+round((x1+Sh)\*k);**

**y2:=y0-round(f(x1+Sh)\*k);**

**line(x,y,x2,y2);**

**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.**

Изображение выглядит как текст, снимок экрана, линия, График

Автоматически созданное описание

Рисунок 6 – результат выполнения программы

**Вывод:** в ходе выполнения задания мы научились создавать подпрограммы: функции и процедуры, изучили способы передачи данных в них, получили навыки работы с case-меню. Для решения задачи нужно было изучить метод Симпсона и реализовать его в коде. Был использован модуль graphAbc для визуализации функции на графике , также было визуализованно задание на экране и была построена трапеция на графике функции. Было реализовано масштабирование в реальном времени и вывод на экран подписей осей X и Y. Задание было решено в полном объеме и был получен опыт в построении графика с помощью функции.