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

Макаров Никита Андреевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

**Цель:** освоить синтаксис построения процедур и функций, изучить способы передачи данных в подпрограммы, получить навыки организации минимального пользовательского интерфейса.

**Формулировка задания (Вариант 9):**

1. Реализовать программу вычисления площади фигуры, ограниченной кривой 2 \* x^3 + (-1) \* x^2 + (-1) \* x + (16) и осью Ox (в положительной части по оси Oy).
2. Вычисление определенного интеграла должно выполняться численно, с применением метода Симпсона.
3. Пределы интегрирования вводятся пользователем.
4. Взаимодействие с пользователем должно осуществляться посредством case-меню.
5. Требуется реализовать возможность оценки погрешности полученного результата.
6. Необходимо использовать процедуры и функции там, где это целесообразно.

**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 разбран отрезок [a, b].

**3. Вычисление ширины подынтервала:**

Расчитайте 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. Вывод результатов:**

Отобразите вычисленную площадь и оценку погрешности пользователю.

**9. Проверка на выход:**

Позвольте пользователю повторить процесс или выйти из программы.

**Схема алгоритма:**

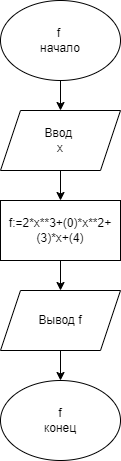


Рисунок 1 – схема алгоритма для функции f

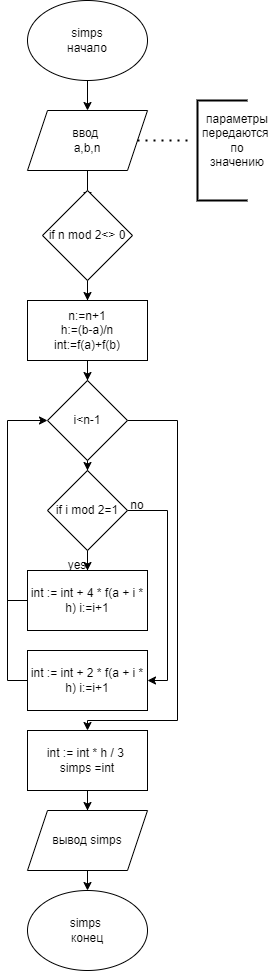


Рисунок 2 – схема алгоритма для функции simps

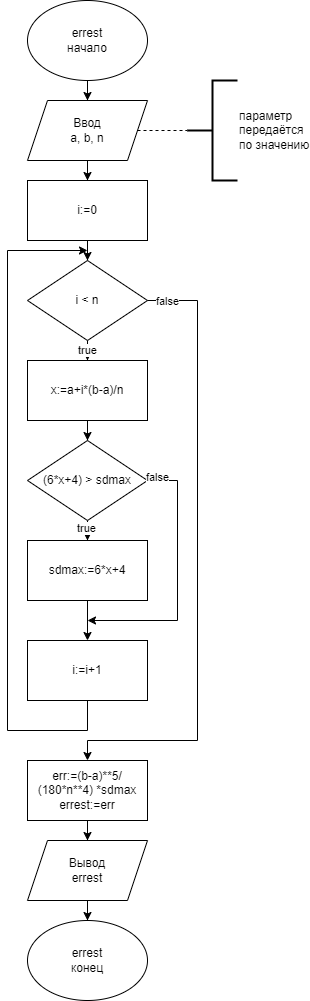


Рисунок 3 – схема алгоритма для функции errest

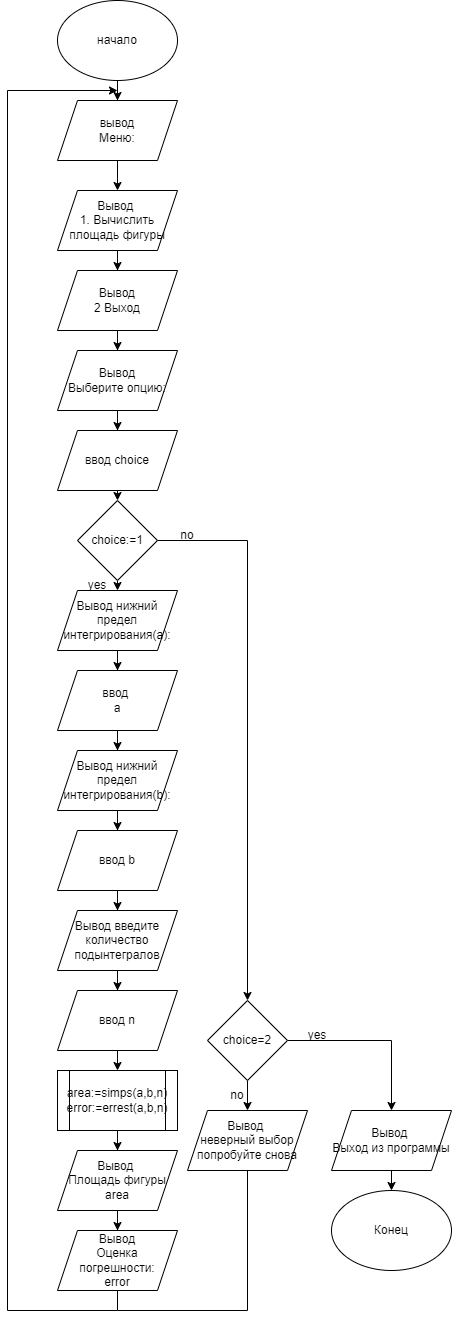


Рисунок 4 – схема алгоритма ввода и вывода результата

**Код программы:**

**uses crt;**

**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**

**a, b: Real;**

**n: Integer;**

**choice: Char;**

**area, error: Real;**

**begin**

**repeat**

**TextColor(15);**

**writeln('Меню:');**

**TextBackground(2);**

**writeln('1. Вычислить площадь фигуры');**

**TextBackground(4);**

**writeln('2. Выход');**

**TextBackground(0);**

**write('Выберите опцию: ');**

**readln(choice);**

**case choice of '1':**

**begin**

**ClrScr;**

**write('Введите нижний предел интегрирования (a): ');**

**readln(a);**

**write('Введите верхний предел интегрирования (b): ');**

**readln(b);**

**write('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':**

**writeln('Выход из программы.');**

**else**

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

**end;**

**until choice = '2';**

**end.**

Изображение выглядит как текст, снимок экрана, Шрифт

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. Также был получен опыт работы с модулем crt. Создали возможность постоянного взаимодействия пользователя с программой с помощью бесконечного цикла. В итоге, мы решили поставленную задачу.