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

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

**Формулировка задания (Вариант 3):**

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 правых прямоугольников.
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-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.

**9. Проверка на выход:**

Позвольте пользователю повторить процесс или выйти из программы.

**Схема алгоритма:**

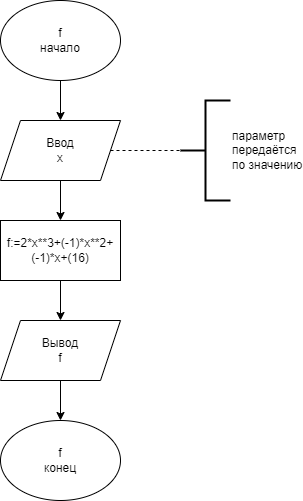


Рисунок 1 – схема алгоритма для функции f

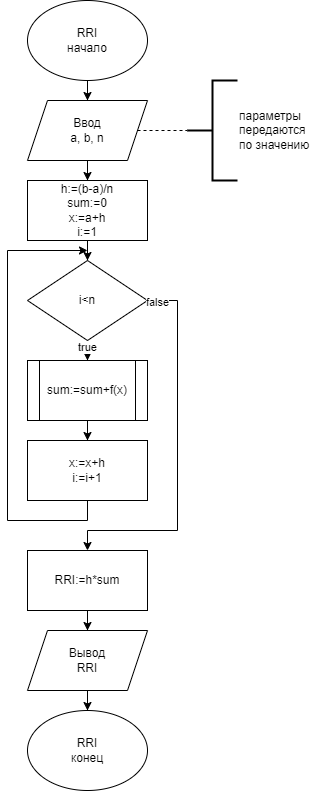


Рисунок 2 – схема алгоритма для функции RRI

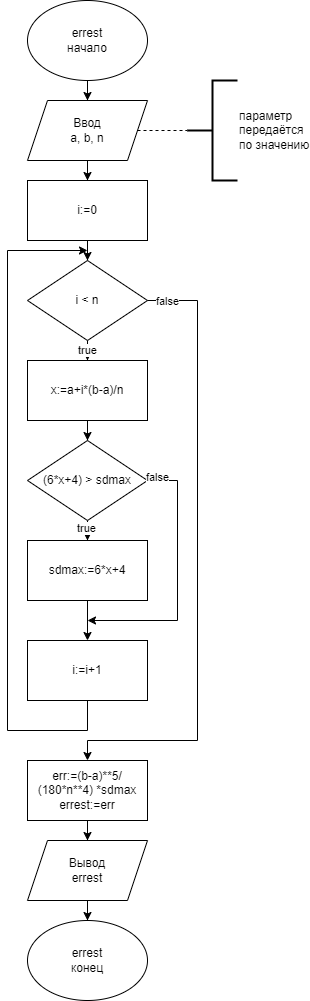


Рисунок 3 – схема алгоритма для функции errest

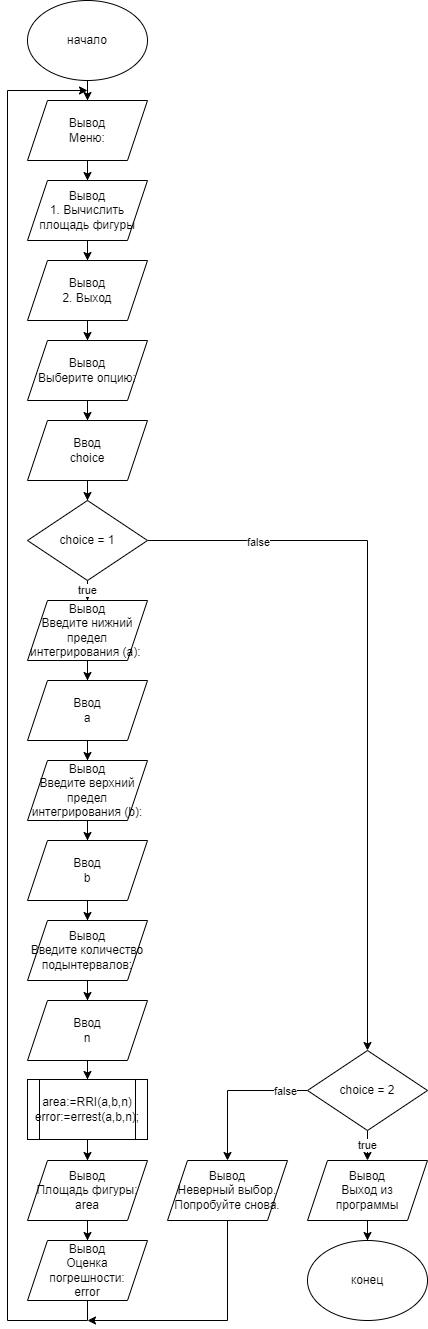


Рисунок 4 – схема алгоритма ввода и вывода результата

**Код программы:**

**uses** crt;

**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**

a, b: Real;

n: Integer;

choice: Char;

area, error: Real;

**begin**

**repeat**

writeln('Меню:');

writeln('1. Вычислить площадь фигуры');

writeln('2. Выход');

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 := RRI(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**.

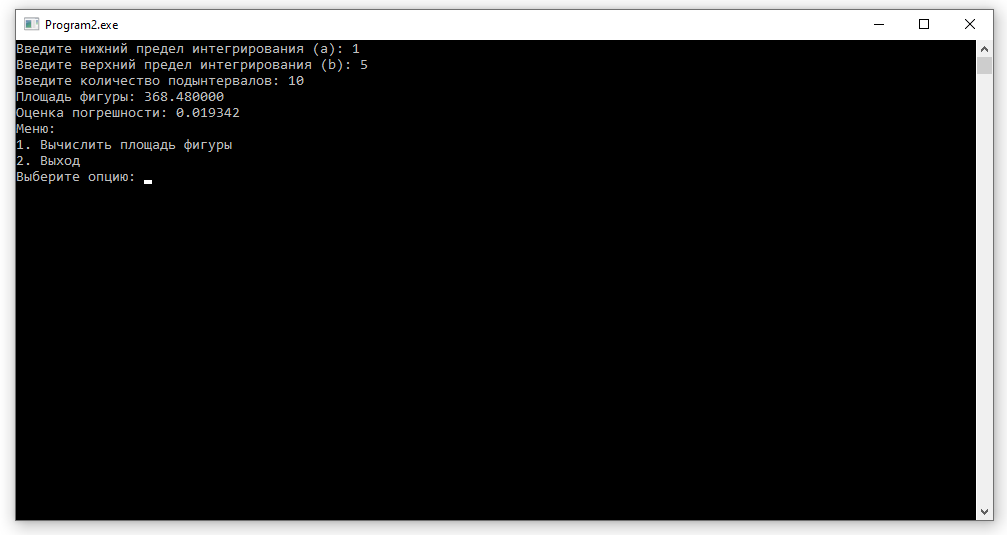


Рисунок 6 – результат выполнения программы

**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 впервые воспользовались модулем Crt. Создали возможность постоянного взаимодействия пользователя с программой с помощью бесконечного цикла. В итоге, мы решили поставленную задачу.