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 программирования»**

Выполнил: студент учебной группы

ИСПк-203-52-00

Яковлев Александр Станиславович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров, 2024

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

**Формулировка задания:**

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

**Описание алгоритма:**

Для решения поставленного здания использовалось три подпрограммы: функция f, необходимая для вычисления начальной функции – 1\*x^3+(2)\*x^2+(3)\*x+(3); функция simpsons\_rule, которая необходима для счёта площади на всех подынтервалов; функция error\_estimate, служащая для счёта погрешности.

В основной программе реализована возможность коммуникации пользователем в case-меню, а также запись основных данных для счёта площади фигуры и возможность завершения работы программы самим пользователем.

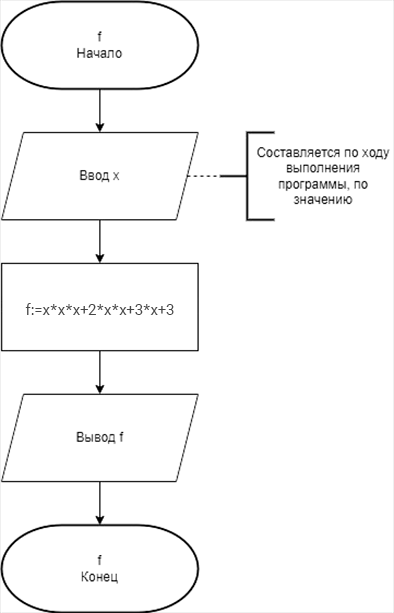


Рисунок 1 – схема алгоритма для функции f

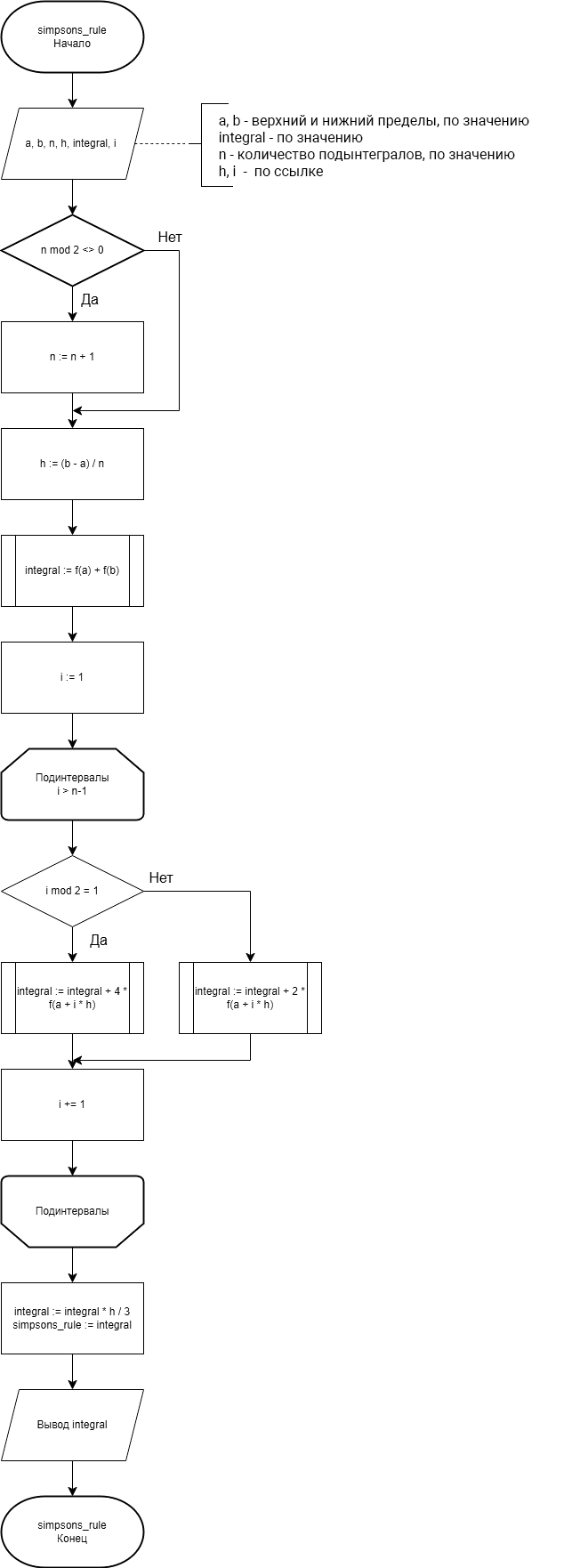


Рисунок 2 – схема алгоритма для функции simpsons\_rule

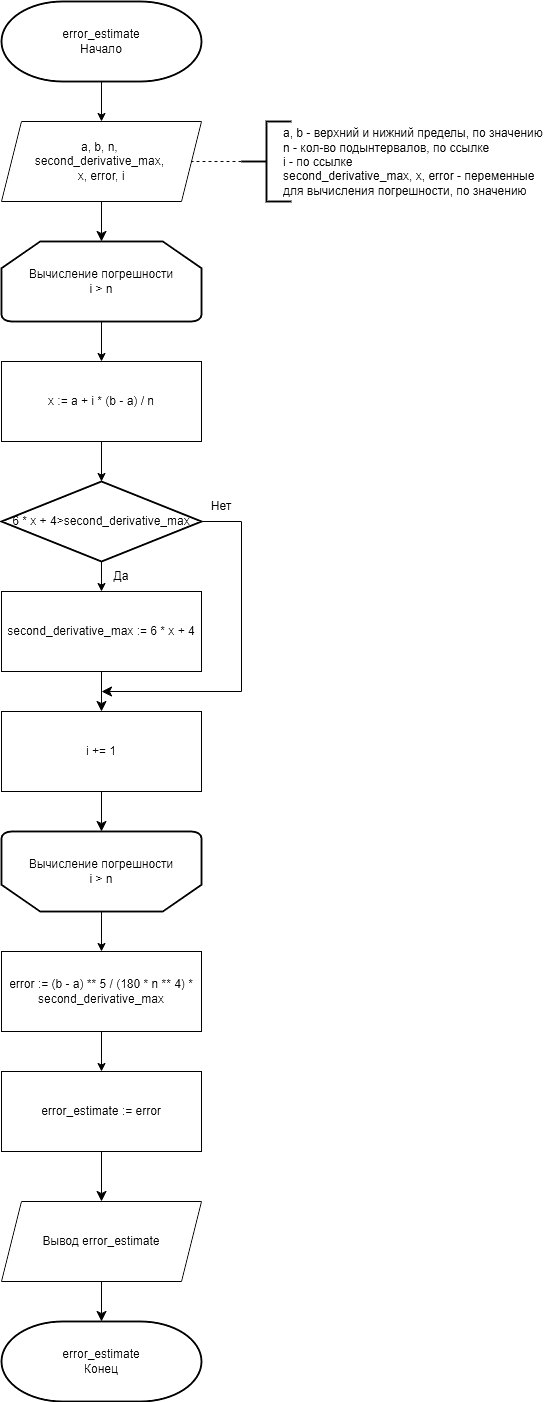


Рисунок 3 – схема алгоритма для функции error\_estimate

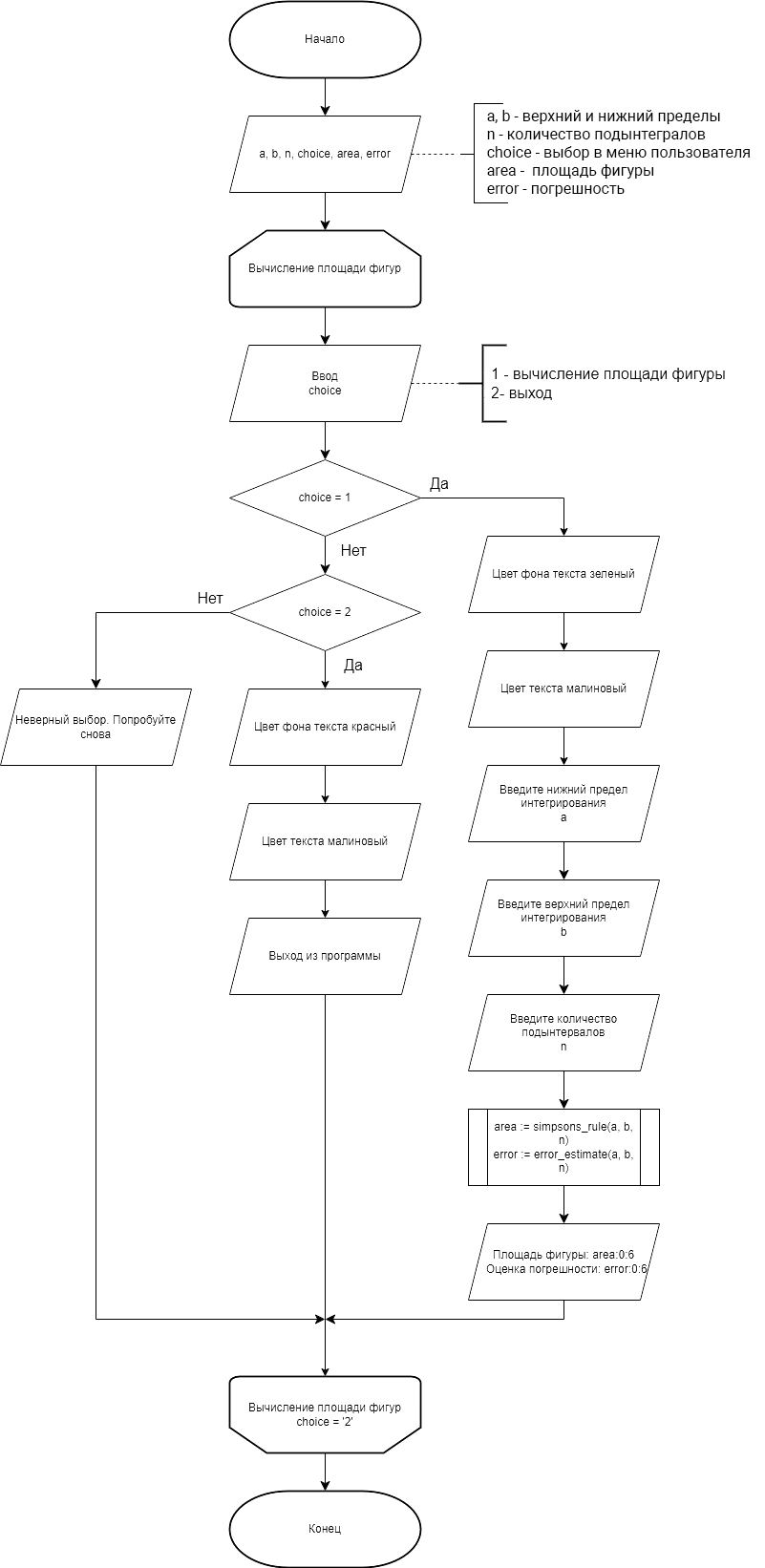


Рисунок 4 – схема алгоритма основной программы

**Код программы:**

**uses** crt;

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

**begin**

f := x \* x \* x + 2 \* x \* x + 3 \* x + 3;

**end**;

**function** simpsons\_rule(a, b: real; n: integer): real;

**var**

h, integral: real;

i: integer;

**begin**

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

n := n + 1;

h := (b - a) / n;

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

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

**begin**

**if** i **mod** 2 = 1 **then**

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

**else**

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

**end**;

integral := integral \* h / 3;

simpsons\_rule := integral;

**end**;

**function** error\_estimate(a, b: real; n: integer): real;

**var**

second\_derivative\_max, x, error: real;

i: integer;

**begin**

second\_derivative\_max := 0;

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

**begin**

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

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

second\_derivative\_max := 6 \* x + 4;

**end**;

error := (b - a) \* power(n, -4) \* second\_derivative\_max / 180; // формула для вычисления погрешности

error\_estimate := error;

**end**;

**var**

a, b: real;

n: integer;

choice: char;

area, error: real;

**begin**

**repeat**

writeln('меню:');

writeln('1. вычислить площадь фигуры');

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 := simpsons\_rule(a, b, n);

error := error\_estimate(a, b, n);

writeln('площадь фигуры: ', area:0:8);

writeln('оценка погрешности: ', error:0:8);

**end**;

'2':

writeln('выход из программы.');

**else**

writeln('неверный выбор. попробуйте снова.');

**end**;

**until** choice = '2';

**end**.

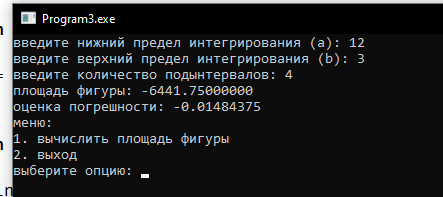


Рисунок 5 – результат выполнения программы

**Вывод:** эта работа предоставила мне возможность глубже разобраться в синтаксисе создания процедур и функций, а также изучить различные методы передачи данных в подпрограммах. Я смог развить навыки по организации простого, но функционального пользовательского интерфейса. Особенно мне понравилось, что данное задание оказалось более сложным по сравнению с предыдущими. Чтобы успешно его выполнить, мне пришлось изучить метод Симпсона и реализовать его в коде, что стало для меня настоящим вызовом.

Кроме того, я ознакомился с работой стандартной библиотеки CRT, что позволило мне расширить свои знания о возможностях языка программирования. Я также попытался улучшить внешний вид пользовательского меню, чтобы сделать его более удобным и интуитивно понятным для пользователей. В целом, хотя работа не была чрезмерно сложной, она оказалась весьма интересной и увлекательной, открыв для меня новые возможности.