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-202-52-00

Коршунов Даниэль Александрович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2022

1. **Цель работы:**

Освоить синтаксис построения процедур и функций, изучить способы передачи данных в подпрограммы, получить навыки организации минимального пользовательского интерфейса.

**2. Задание (Вариант 7):**

1.Реализовать программу вычисления площади фигуры, ограниченной кривой 2\*х^3+ (1) \*х^2 + (-4) \*х + (1) и осью ОХ (в положительной части по оси ОY).

2.Вычисление определенного интеграла должно выполняться численно, с применением метода средних прямоугольников.

3.Пределы интегрирования вводятся пользователем.

4.Взаимодействие с пользователем должно осуществляться посредством cas-меню.

5.Требуется реализовать возможность оценки погрешности полученного результата.

6.Необходимо использовать процедуры и функции там, где это целесообразно.

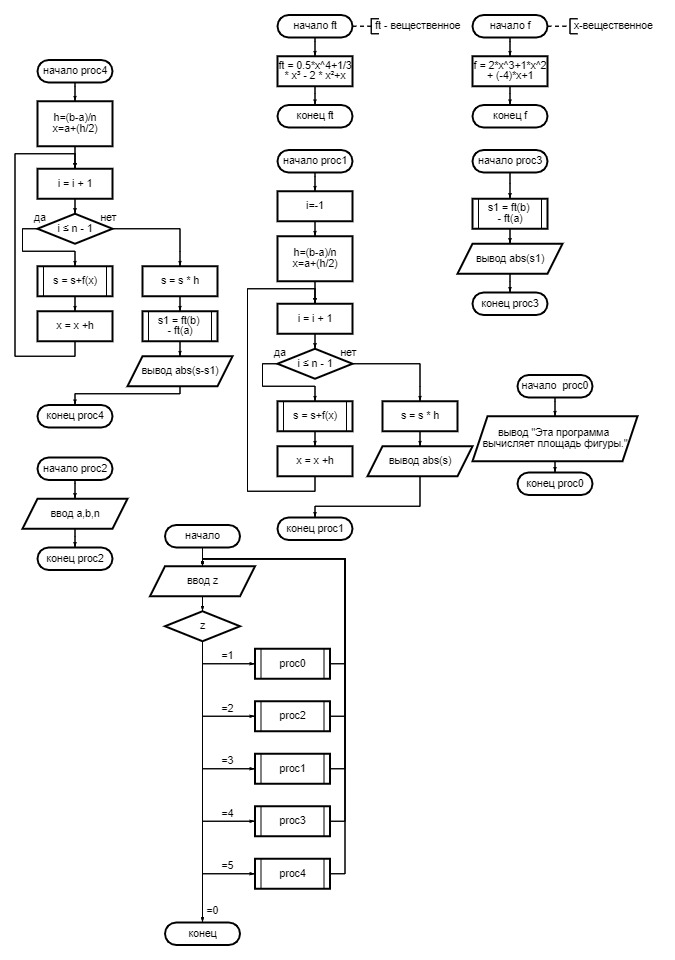
**3. Схема алгоритма**

Рисунок 1 - Схема алгоритма с комментариями для задания 1

**4. Код программы**

**Uses** Crt;

**var**

a, b, h, x, s, p, s1: real;

n, z: integer;

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

**begin**

f := 2 \* power(x, 3) + 1 \* power(x, 2) + (-4) \* x + 1;

**end**;

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

**begin**

ft := 0.5 \* power(x, 4) + 1 / 3 \* power(x, 3) - 2 \* power(x, 2) + x;

**end**;

**procedure** proc1;

**var**

i: integer;

**begin**

ClrScr;

h := (b - a) / n;

x := a + (h / 2);

**for** i := 0 **to** n - 1 **do**

**begin**

s += f(x);

x += h;

**end**;

s \*= h;

writeln('Приближённая площадь фигуры = ', abs(s):0:2);

Readkey;

**end**;

**procedure** proc3;

**var**

i: integer;

**begin**

ClrScr;

s1 := ft(b) - ft(a);

writeln('Точная площадь фигуры = ', abs(s1):0:2);

Readkey;

**end**;

**procedure** proc4;

**var**

i: integer;

**begin**

ClrScr;

h := (b - a) / n;

x := a + (h / 2);

**for** i := 0 **to** n - 1 **do**

**begin**

s += f(x);

x += h;

**end**;

s \*= h;

s1 := ft(b) - ft(a);

writeln('Погрешность = ', abs(s - s1):0:2);

Readkey;

**end**;

**procedure** proc2;

**begin**

ClrScr;

WriteLn('введите область итеграции(>1)');

ReadLn(a, b);

WriteLn('введите количество промежутков');

ReadLn(n);

**end**;

**procedure** proc0;

**begin**

ClrScr;

WriteLn('Эта программа вычисляет площадь фигуры.');

readkey;

**end**;

**begin**

**repeat**

ClrScr;

writeln('формулировка -1');

WriteLn('ввод a,b,n - 2');

WriteLn('приближённое вычисление - 3');

WriteLn('точное вычисление - 4');

WriteLn('погрешность - 5');

WriteLn('Exit - 0');

ReadLn(z);

**case** z **of**

1: proc0;

2: proc2;

3: proc1;

4: proc3;

5: proc4;

**end**;

**until** z = 0;

**end**.

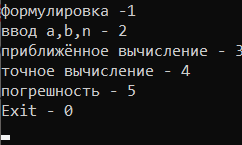
**5. Результат выполнения программы:**

Рисунок 2 – Главное меню программы

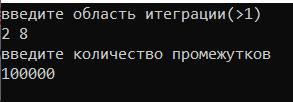
****

Рисунок 3 – Ввод данных

****

Рисунок 4 – Приближённая площадь

****

Рисунок 5 – Точная площадь

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок 3 – Погрешность

**6. Вывод:**

Программа работает так как нужно. Была не большая проблема при решении этого задания, но она быстро решилась. Было изучены Case меню и использованы в большом количестве подпрограммы, в частности функции и процедуры. Эта программа помогает на много быстрее выполнить вычисления, чем это делать самому. Это очень простое задание и программа для решения этого задания простая, но надо было вспоминать и писать формулы вычисления площади криволинейной трапеции. Было тяжело правильно написать алгоритм высчитывания не точного значения площади через средние прямоугольники. Нужно понимать, что этот алгоритм решения не точны и всегда будет погрешность. Можно было улучшить программу сделав в интерфейсе управление стрелочками.