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

Цветкова Кристина Андреевна

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

1. **Цель работы**

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

1. **Формулировка задания**

Вариант: 22

1. Реализовать программу вычисления площади фигуры, ограниченной кривой.

2\*х^ 3 + (0)\*х^2 + (-5)\*× + (17) и осью ОХ (в положительной части по оси ОY).

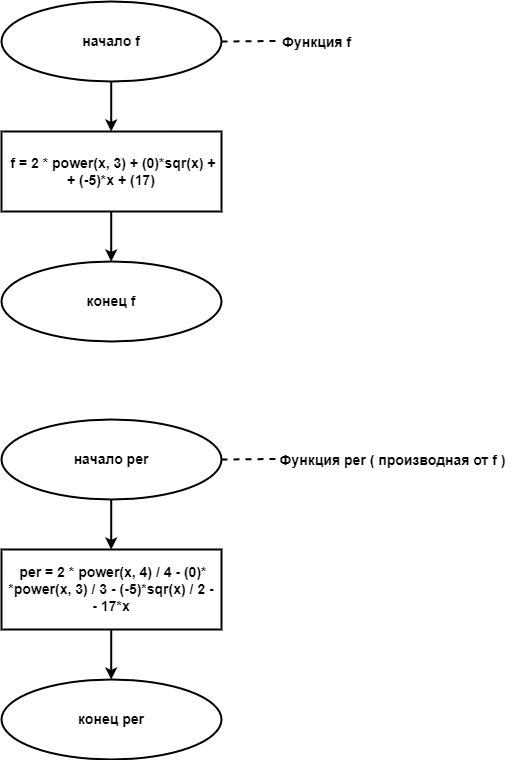
1. Вычисление определенного интеграла должно выполняться численно, с применением метода Симпсона.
2. Пределы интегрирования вводятся пользователем.
3. Взаимодействие с пользователем должно осуществляться посредством case-меню.
4. Требуется реализовать возможность оценки погрешности полученного результата.
5. Необходимо использовать процедуры и функций там, где это целесообразно.
6. **Описание алгоритма**
7. Начинаем программу.
8. Создаем функцию f(x), которая принимает вещественное число x и возвращает результат уравнения 2 \* x^3 - 5x + 17.
9. Создаем функцию per(x), которая принимает вещественное число x и возвращает результат уравнения 2 \* x^4 / 4 - 5x^2 / 2 - 17x.
10. Объявляем переменные a, b, s, y как вещественные числа, а также переменные n, g как целые числа.
11. Начинаем цикл повторения.
12. Выводим сообщение пользователю о выборе действия.
13. Считываем выбор пользователя в переменную g.
14. Если пользователь выбрал действие 1, то начинаем вложенный цикл повторения.
15. Выводим сообщение пользователю о вводе границ интегрирования.
16. Считываем значения a и b до тех пор, пока они не будут меньше или равны 10000.
17. Выводим сообщение пользователю о вводе числа промежутков.
18. Считываем значение n до тех пор, пока оно не будет меньше или равно 10000.
19. Вычисляем значение s как (f(a) + f(b)) / 2.
20. Запускаем цикл от 1 до n - 1 и на каждой итерации прибавляем к s значение f(a + i \* ((b - a) / n)).
21. Умножаем s на (b - a) / n.
22. Вычисляем значение y как per(b) - per(a).
23. Выводим результаты: площадь фигуры (модуль значения s с точностью до семи знаков после запятой) и погрешность (модуль разницы между y и s с точностью до семи знаков после запятой).
24. Возвращаемся к шагу 5, если пользователь не выбрал действие 2.
25. Завершаем программу, если пользователь выбрал действие 2.
26. **Схема алгоритма**

Рисунок 1 – Схемы алгоритмов функций

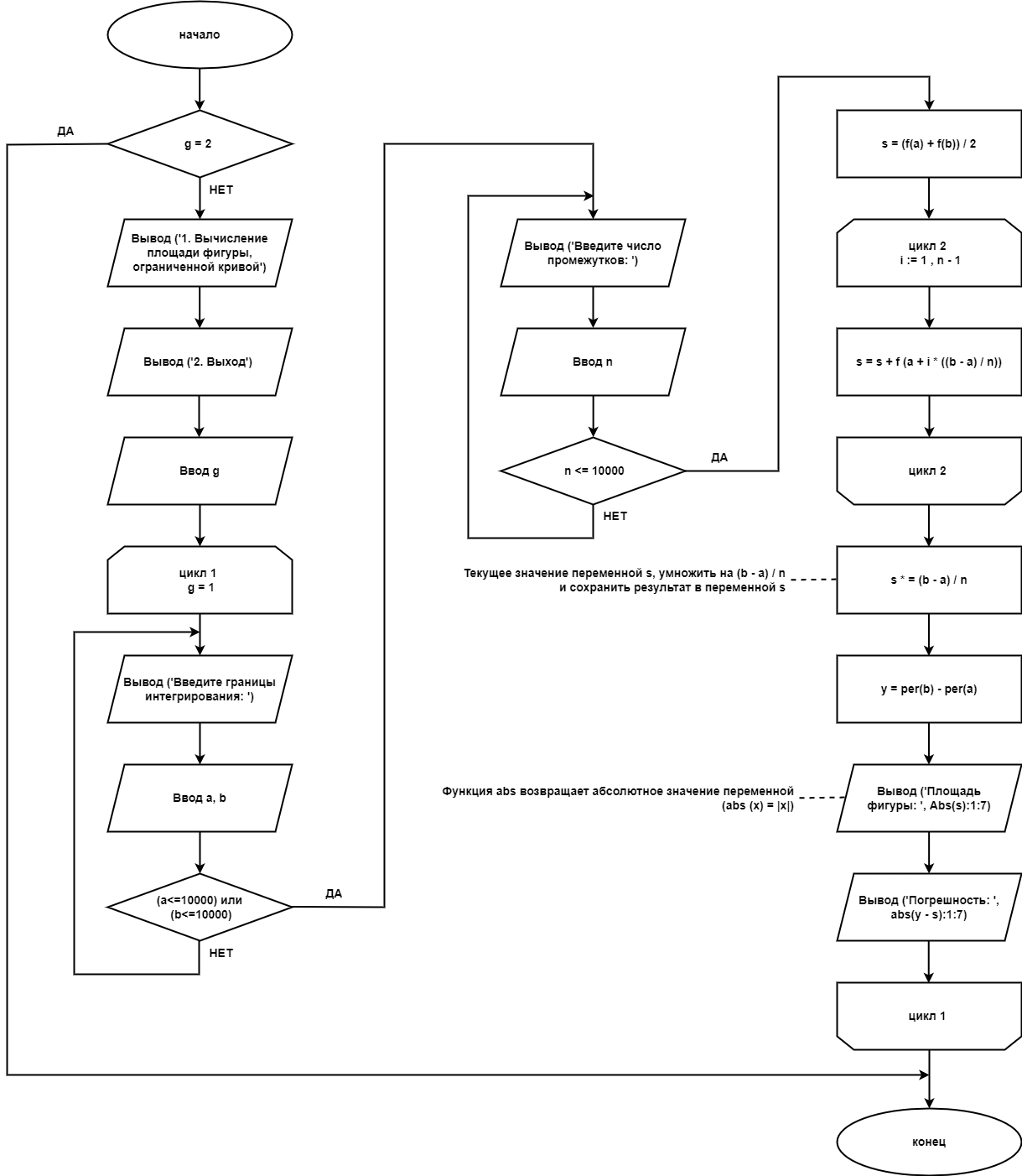


Рисунок 2 – Схема алгоритма

1. **Код программы**

**uses** Crt;

**procedure** q;

**begin**

writeln('1. Вычисление площади фигуры, ограниченной кривой.');

writeln('2. Выход.');

**end**;

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

**begin**

f := 2 \* power(x, 3) + (0)\*sqr(x) +(-5)\*x+(17);

**end**;

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

**begin**

per:=2 \* power(x, 4) / 4 - (0)\*power(x, 3) / 3 - (-5)\*sqr(x) / 2 - 17\*x;

**end**;

**var** a, b, h, s, y: real;

**var** n, g: integer;

**begin**

**repeat**

writeln('1. Вычисление площади фигуры, ограниченной кривой');

writeln('2. Выход');

write('Выберите действие: ');

readln(g);

**until** (g=1) **or** (g=2) ;

**repeat**

writeln('Введите границы интегрирования: ');

readln(a, b);

**until** (a<=1000)**or** (b<=1000);

**repeat**

write('Введите число промежутков: ');

readln(n);

**until** (n<=1000);

s := (f(a) + f(b)) / 2;

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

**begin**

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

**end**;

s \*= (b - a) / n;

y := per(b) - per(a);

**repeat**

q;

read(g);

**case** g **of**

1:

**begin**

writeln('Приблизительное значение площади = ', Abs(s):1:7);

writeln('Точное значение площади = ', abs(y):1:7);

writeln('Абсолютная погрешность = ', abs(y - s):1:7);

writeln('Относительная погрешность = ', abs(y - s) / y);

writeln;

**end**;

2: halt;

**end**;

**until** g=2;

**end**.

1. **Результат выполнения программы:**

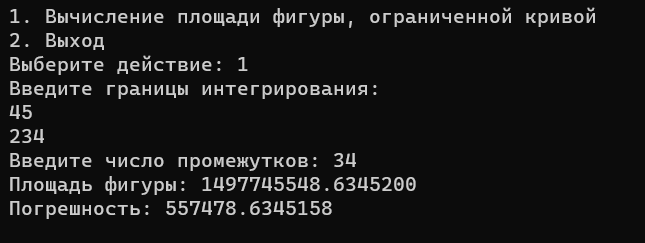
****

Рисунок 3 – Работа программы

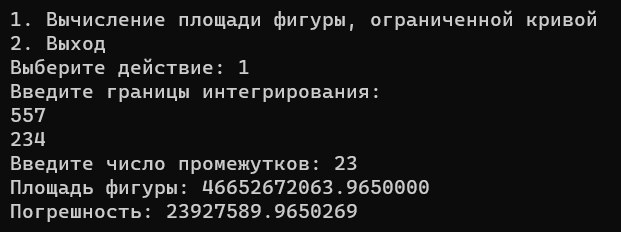


Рисунок 4 – Работа программы

1. **Вывод**

Выведенная программа реализует метод численного интегрирования методом Симпсона для вычисления площади фигуры, ограниченной кривой, определенной математической функцией.

При выполнении данной работы возникли трудности с реализацией case menu. Поэтому для решения задачи понадобилось использовать материалы по написанию программы с case menu, основным источником послужила предоставленная презентация.

В процессе выполнения работы, для создания блок схемы мы познакомились с программой Draw io. Draw.io — инструмент для создания диаграмм, схем алгоритмов, интеллект-карт, бизнес-макетов, отношений сущностей, программных блоков и другого. Draw.io обладает богатым набором функций для визуализации большинства задач пользователя.