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

Генинг Богдана Андреевна

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

1. **Цель работы:** освоить синтаксис построения процедур и функций, изучить способы передачи данных подпрограммы, получить навыки организации минимального пользовательского интерфейса.
2. **Формулировка задания (Вариант:7)**
3. Реализовать программу вычисления площади фигуры, ограниченной кривой 1\*x^3+(-2)\*x^2+(3)\*x+(0) и осью 0x (в положительной части по оси 0y)
4. Вычисление определённого интеграла должно выполняться численно, с применением метода Симпсона
5. Пределы интегрирования вводятся пользователем
6. Взаимодействие с пользователем должно осуществляться посредством case-меню
7. Требуется реализовать возможность оценки погрешности полученного результата
8. Необходимо использовать процедуры функции там, где это целесообразно
9. **Описание алгоритма**

Данный алгоритм реализует метод Симпсона для вычисления приближенной площади под графиком функции F(x). Он содержит несколько функций и процедур:

1. Функция F(x: Real): Real; - вычисляет значение функции F(x) = 1\*x\*x\*x + (-2)\*x\*x + 3\*x + 0.

2. Функция Simpson(a, b: Real; n: Integer): Real; - выполняет вычисление методом Симпсона. Принимает на вход нижний (a) и верхний (b) пределы интегрирования, а также количество разбиений (n).

3. Процедура Menu() - предоставляет пользователю меню для выбора действий. Включает в себя запрос на ввод нижнего и верхнего пределов интегрирования, а также количества разбиений. Рассчитывает значение площади фигуры методом Симпсона и вычисляет погрешность.

4. Основная программа начинается с вызова процедуры Menu(), которая обеспечивает взаимодействие с пользователем.

Данный алгоритм позволяет вычислять площадь фигуры под графиком функции используя метод Симпсона и оценивать погрешность данного вычисления.

1. **Схема алгоритма**

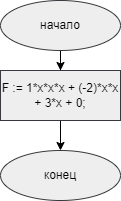
****

Рис 1. Схема алгоритма

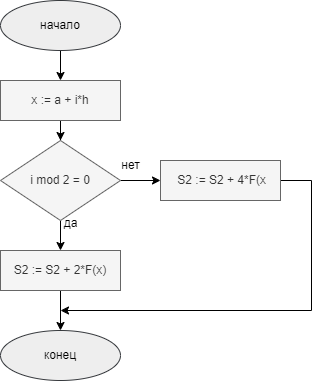


Рис 2. Схема алгоритма

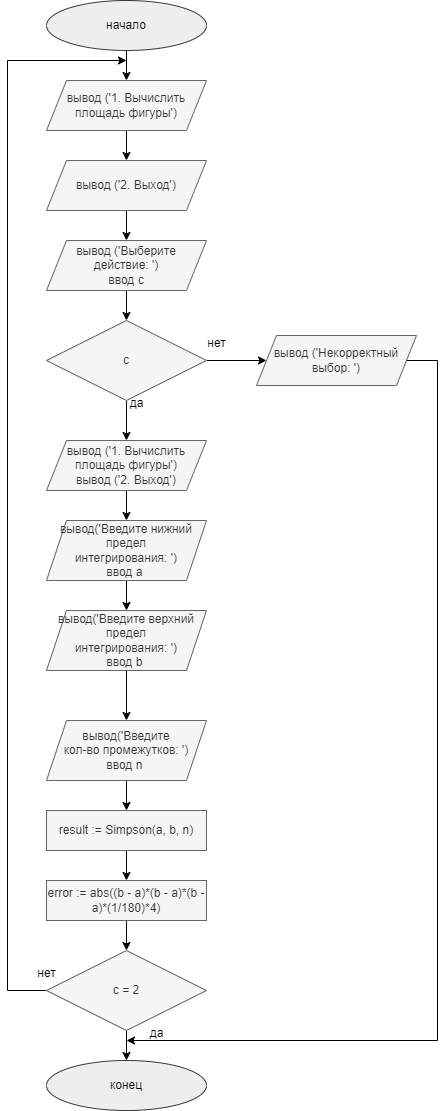
****

Рис 3. Схема алгоритма

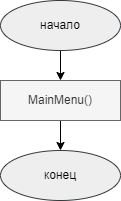
****

Рис 4. Схема алгоритма

1. **Код программы:**

**function** F(x: Real): Real;

**begin**

F := 1\*x\*x\*x + (-2)\*x\*x + 3\*x + 0;

**end**;

**function** Simpson(a, b: Real; n: Integer): Real;

**var**

h, x, S1, S2: Real;

i: Integer;

**begin**

h := (b - a) / n;

S1 := F(a) + F(b);

S2 := 0;

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

**begin**

x := a + i\*h;

**if** i **mod** 2 = 0 **then**

S2 := S2 + 2\*F(x)

**else**

S2 := S2 + 4\*F(x);

**end**;

Simpson:= (h/3)\*(S1 + S2);

**end**;

**procedure** Menu();

**var**

a, b: Real;

n: Integer;

result, error: Real;

c: Integer;

**begin**

**repeat**

writeln('1. Вычислить площадь фигуры');

writeln('2. Выход');

write('Выберите действие: ');

readln(c);

**case** c **of**

1:

**begin**

write('Введите нижний предел интегрирования: ');

readln(a);

write('Введите верхний предел интегрирования: ');

readln(b);

write('Введите количество промежутков: ');

readln(n);

result := Simpson(a, b, n);

writeln('Площадь фигуры: ', result:0:4);

// вычисление погрешности

error := abs((b - a)\*(b - a)\*(b - a)\*(1/180)\*4); // формула погрешности для метода Симпсона

writeln('Погрешность: ', error:0:8);

**end**;

2: writeln('Выход');

**else**

writeln('Некорректный выбор');

**end**;

**until** choice = 2;

**end**;

**begin**

Menu();

**end**.

1. **Результат выполнения программы**

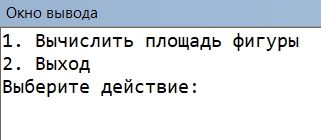


Рис 5. Результат выполнения программы

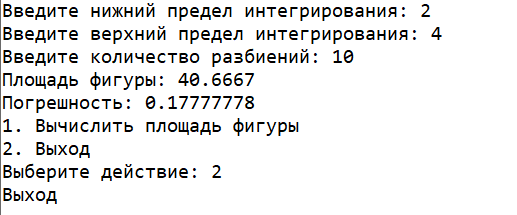


Рис 6. Результат выполнения программы

1. **Вывод**

В ходе данной домашней контрольной работы мы освоили синтаксис построения процедур и функций, изучили способы передачи данных подпрограммы, получили навыки организации минимального пользовательского интерфейса.

Мы успешно справились с выполнением задания и были заинтересованы и увлечены процессом изучения языка программирования Pascal. Понимание использования циклов и условий было закреплено, и мы достигли всех поставленных целей. Это дало нам уверенность в наших навыках программирования и мотивацию для дальнейшего развития.