Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №3**

**«ВЫЧИСЛЕНИЕ ЗНАЧЕНИЯ ФУНКЦИИ»**

**ПО ДИСЦИПЛИНЕ «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»**

Выполнил: студент учебной группы

ИСПк 205-52-00 Касьянова Славяна

Александровна

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

1.Цель работы: Освоить синтаксис построения процедур и функций, изучить способы передачи данных программы, получить навыки организации минимального пользовательского интерфейса.

Формулировка задания (с вариантом)

Вариант 8

1. 2\*×^ 3 + (1) \*x^2 + (1) \*x+ (19) и осью ОХ (в положительной части по оси OY). Вычисление определенного интеграла должно выполняться численно, с применением метода Симпсона. Пределы интегрирования вводятся пользователем. Взаимодействие с пользователем должно осуществляться посредством case-меню. Требуется реализовать возможность оценки погрешности полученного результата. Необходимо использовать процедуры и функции там, где это целесообразно.
2. Описание алгоритма

Задание 1:

В этой программе мы используем метод прямоугольников для вычисления площади фигуры, ограниченной кривой. Мы задаем начальную и конечную точку на оси OX (a и b), затем вычисляем значение функции в каждой точке от a до b с шагом h. Затем суммируем площади прямоугольников, образованных этими значениями функции, чтобы получить общую площадь фигуры.

В этой программе мы используем метод Симпсона для вычисления определенного интеграла. Пользователь вводит нижний и верхний пределы интегрирования, а также количество разбиений. Затем мы используем функцию SimpsonMethod для вычисления значения определенного интеграла. Мы также можем вызвать процедуру CalculateError для оценки погрешности результата.

Программа также вычисляет погрешность метода, сравнивая результат с точным значением интеграла.

Основные функции программы:

1. Функция f(a:real):real; - вычисляет значение функции 2\*x\*x\*x + x\*x + x + 19.

2. Функция metod(a, b: Real; n: Integer): Real; - реализует метод прямоугольников для вычисления приближенного значения определенного интеграла функции на заданном интервале.

3. Функция metod2(a,b:real): Real; - вычисляет точное значение определенного интеграла функции на заданном интервале.

4. Процедура pogresh; - вычисляет погрешность метода прямоугольников.

Основной блок программы представляет собой меню для выбора программы:

- Пункт "1" запускает программу для вычисления интеграла методом прямоугольников и выводит результаты на экран.

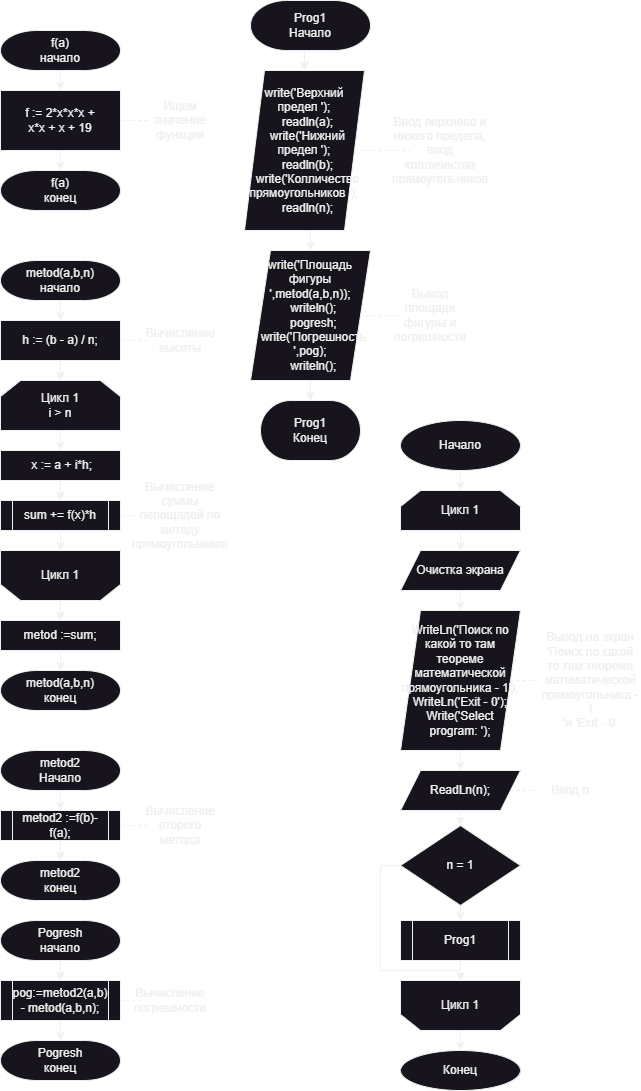
- Пункт "0" завершает выполнение программы.

При выборе пункта "1" пользователю предлагается ввести верхний и нижний пределы интегрирования, а также количество прямоугольников для аппроксимации интеграла. Затем программа вычисляет приближенное значение определенного интеграла и погрешность метода прямоугольников, выводит результаты на экран.

Программа продолжает работу до тех пор, пока пользователь не выберет пункт "0" для завершения выполнения программы.

1. Схема алгоритма с комментариями

Задача 1



1. Код программы

Задача 1

**uses** crt;

**var**

a,b,pog:real;

n,j,x:integer;

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

**begin**

f := 2\*x\*x\*x + x\*x + x + 19;

**end**;

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

**var**

h, x, sum: Real;

i: Integer;

**begin**

h := (b - a) / n;

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

**begin**

x := a + i\*h;

sum += f(x)\*h

**end**;

metod :=sum;

**end**;

**function** metod2(a,b:real): Real;

**begin**

metod2 :=f(b)-f(a);

**end**;

**procedure** pogresh;

**begin**

pog:=metod2(a,b) - metod(a,b,n);

**end**;

**Procedure** prog1;

**begin**

write('Верхний предел ');

readln(a);

write('Нижний предел ');

readln(b);

write('Колличество прямоугольников ');

readln(n);

write('Площадь фигуры ',metod(a,b,n));

writeln();

pogresh;

write('Погрешность ',pog);

writeln();

**end**;

**begin**

**repeat**

ClrScr;

WriteLn('Поиск по теореме симпсона - 1');

WriteLn('Exit - 0');

Write('Select program: ');

ReadLn(n);

**Case** n **of**

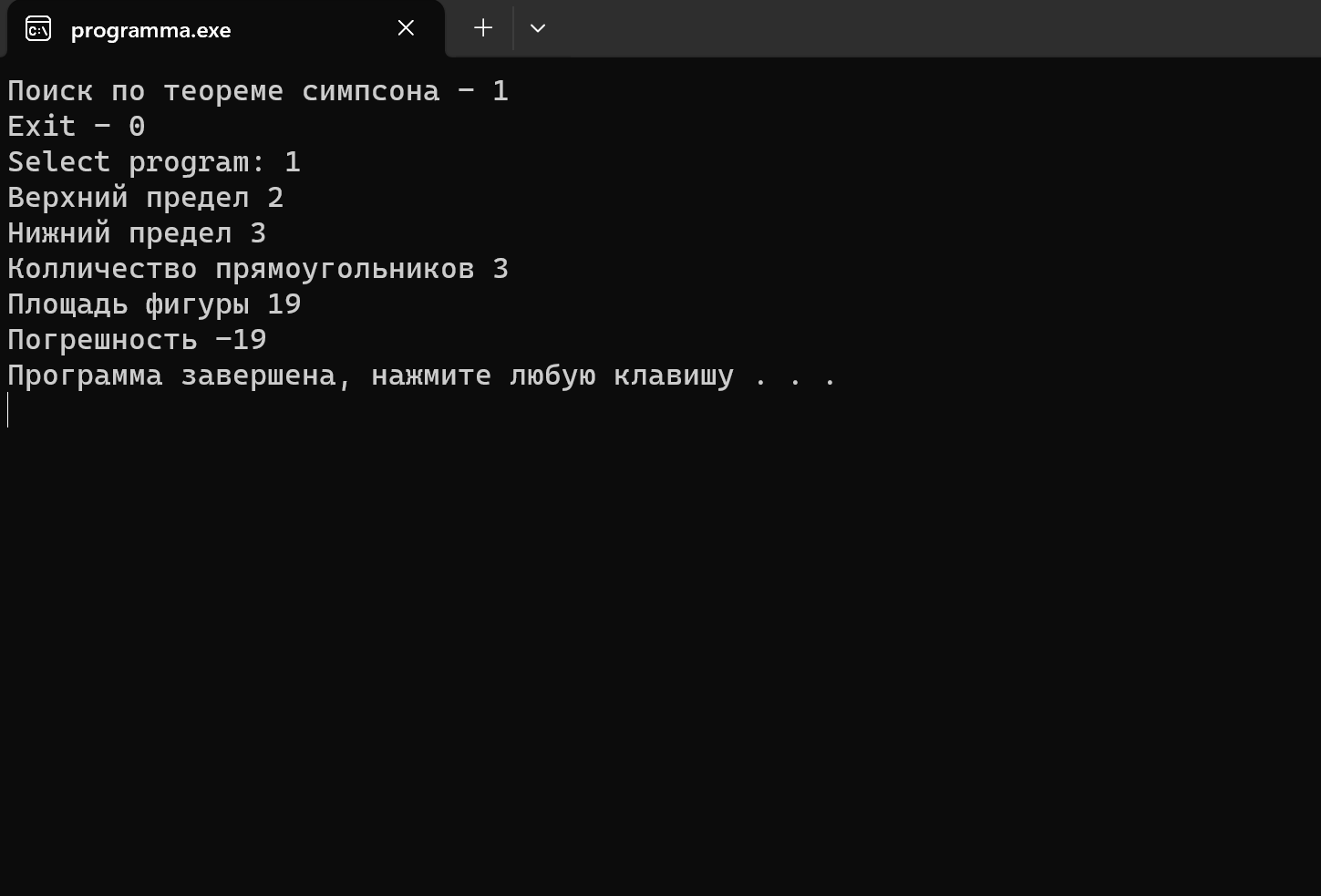
1: prog1;

**end**;

**until** j = 0;

**end**.

1. Результат при выводе



1. По итогу завершения работы, мы изучили базовую структуру организации программы и основы конструкции языка программирования

Pascal. В ходе выполнения контрольной работы, мы узнали новую переменную function и разобрались с её значением. Так же столкнулись с некоторыми проблемами. Во-первых мы не знали как работает функция в Pascal.ABC в следствии чего разобрались с данной проблемой и узнали новый для себя метод Симпсона.