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-204-52-00

Зараменских Илья Алексеевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

1. **Цель работы:** освоить синтаксис построения процедур и функций, изучить способы передачи данных в подпрограммы, получить навыки организации минимального пользовательского интерфейса
2. **Формулировка задания:** Вариант 6
   1. Реализовать программму вычисления площади фигуры, ограниченной кривой
   2. 2 \* x ^ 3 + (2) \* x ^ 2 + (2) \* x + (19) и осью OX (в положительной части по оси OY).
   3. Вычисление определенного интеграла должно выполняться численно, с применением метода правых прямоугольников.
   4. Пределы интегрирования вводятся пользователем.
   5. Взаимодействие с пользователем должно осуществляться посредством case-меню.
   6. Требуется реализовать возможность оценки погрешности полученного результата.
   7. Необходимо использовать процедуры и функции там, где это целесообразно.
3. **Описание алгоритма:** Программа представляет собой реализацию численного метода вычисления определенного интеграла, а именно метода правых прямоугольников. В начале работы происходит инициализация переменных и вывод интерактивного меню, которое предоставляет пользователю возможность выбрать одно из четырех действий: ввод пределов интегрирования и количества разбиений, вычисление площади фигуры, оценка погрешности результата, либо завершение работы программы. Пользователь делает выбор действия через ввод соответствующего номера.

Для вычислений используется функция f(x), которая задает математическое выражение. Эта функция возвращает значение выражения для заданного x. Основной расчет площади выполняется методом правых прямоугольников, где область под графиком разбивается на n равных частей. Шаг разбиения вычисляется, после чего значения функции суммируются в правых концах каждого из подотрезков. Произведение этой суммы на шаг дает приближенное значение площади под графиком.

Для оценки точности вычислений реализована функция, которая сравнивает результаты метода правых прямоугольников с количеством разбиений n и 2n. Разница между ними в абсолютном выражении служит оценкой погрешности результата. Пользователь может поочередно вводить пределы интегрирования, выбирать количество разбиений, вычислять площадь фигуры и анализировать точность расчета. Работа программы продолжается до тех пор, пока пользователь не выберет опцию завершения.

1. **Схема алгоритма:**

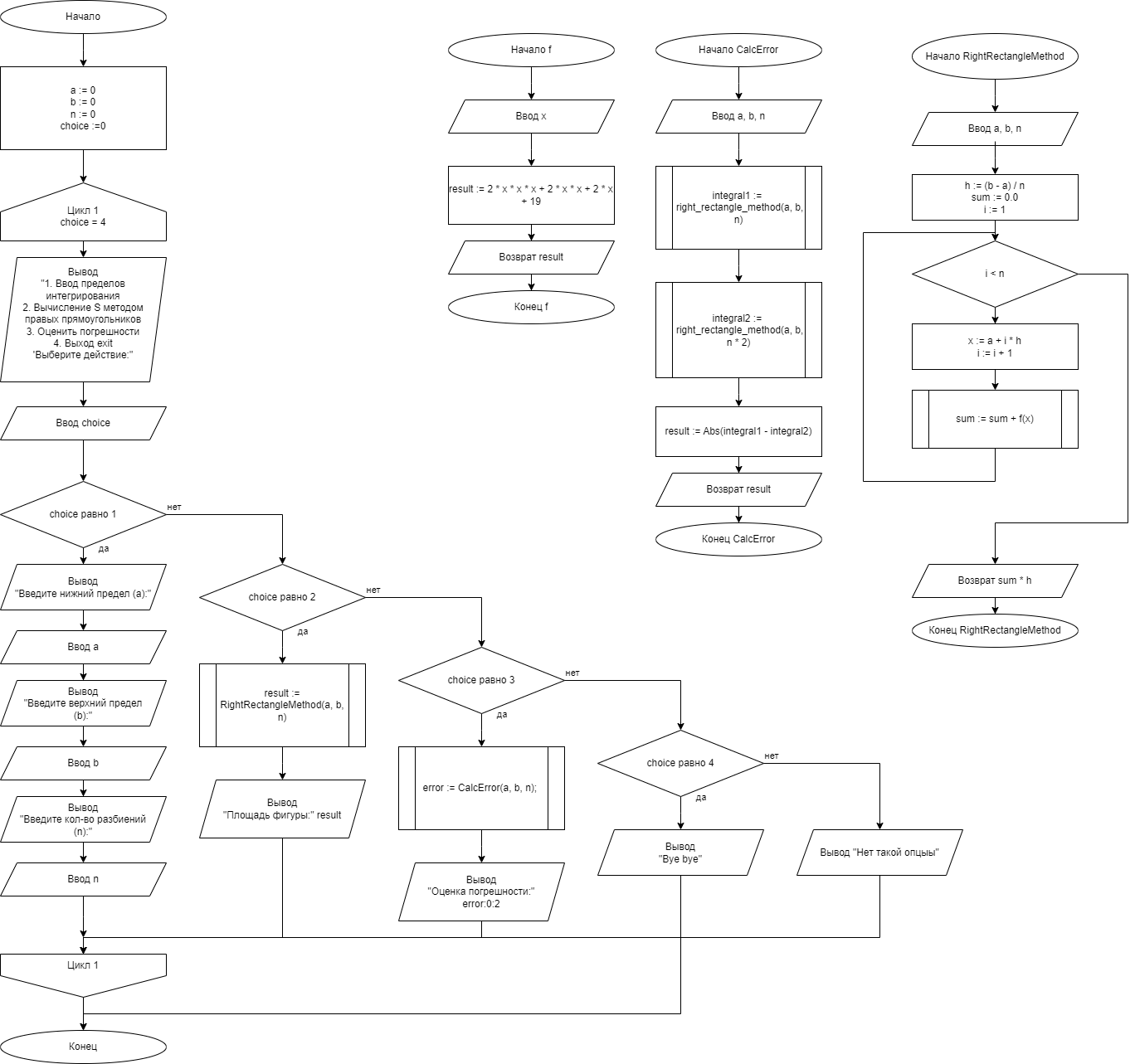
****

Рисунок 1 – Схема алгоритма

1. **Код программы:**

uses crt;

function f(x: Real): Real;

begin

f := 2 \* x \* x \* x + 2 \* x \* x + 2 \* x + 19; // Вычисляем значение функции

end;

// Метод правых прямоугольников

function RightRectangleMethod(a, b: Real; n: Integer): Real;

var

i: Integer;

x, h, sum: Real;

begin

h := (b - a) / n;

sum := 0.0;

// Цикл для сложнеших калькуляций

for i := 1 to n do

begin

x := a + i \* h; // Правая граница

sum := sum + f(x);

end;

RightRectangleMethod := sum \* h; // сумма \* шаг

end;

// Оценка погрешности

function CalcError(a, b: Real; n: Integer): Real;

var

integral1, integral2: Real;

begin

integral1 := RightRectangleMethod(a, b, n); // Интеграл с n частями

integral2 := RightRectangleMethod(a, b, n \* 2); // Интеграл с 2n частями

CalcError := Abs(integral1 - integral2); // Разница

end;

// Основная программа

var

a, b: Real;

n, choice: Integer;

result, error: Real;

begin

ClrScr; // Протереть экран

repeat

// Меню через case

writeln('1. Ввод пределов интегрирования');

writeln('2. Вычисление S методом правых прямоугольников');

writeln('3. Оценить погрешности');

writeln('4. Выход exit');

write('Выберите действие: ');

readln(choice);

case choice of

1:

begin

// Менюшка

writeln('Введите нижний предел (a): ');

read(a);

writeln('Введите верхний предел (b): ');

read(b);

writeln('Введите кол-во разбиений (n): ');

read(n);

end;

2:

begin

// Посчитать площадь

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

writeln(' ');

writeln('S фигуры: ', result:0:2);

writeln(' ');

end;

3:

begin

// Оценка погрешности

error := CalcError(a, b, n);

writeln(' ');

writeln('Оценка погрешности: ', error:0:2);

writeln(' ');

end;

4:

writeln('Bye bye')

else

writeln('Нет такой опцыы');

end;

until choice = 4; // Цикл работает до выхода

end.**Результат выполнения программы**

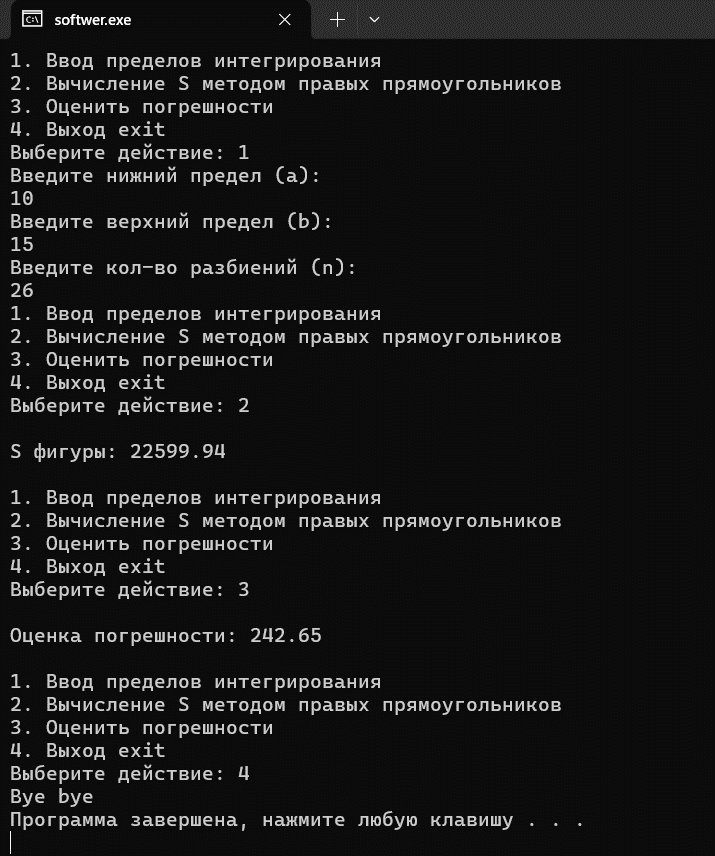


Рисунок 2 – результат выполнения кода

1. **Вывод:** В ходе работы были изучены базовые принципы организации процедур и функций, что является важным аспектом структурного программирования. Были применены ключевые конструкции языка Pascal, такие как циклы, условные операторы и ветвления, которые позволяют реализовать гибкую и понятную логику программы.

Одним из важных моментов стало использование процедур и функций для декомпозиции задачи на небольшие, легко управляемые части. Это повышает читаемость кода, облегчает его отладку и повторное использование. Функции позволяют возвращать результат вычислений, а процедуры используются для выполнения определенных действий без необходимости возврата значения.

Для управления потоками выполнения программы применялись циклы и конструкция `case`, которые обеспечивают возможность выбора пользователем различных действий. Это продемонстрировало, как можно организовать взаимодействие программы с пользователем в структурированной и понятной форме.

Также в ходе работы рассматривалось форматирование вывода результатов, что позволяет повысить удобство их восприятия. Использование числовых форматов помогает привести выводимые данные к стандартному виду с заданной точностью.

Работа помогла освоить принципы структурного подхода к программированию, научиться организовывать код с использованием процедур и функций, а также эффективнее использовать базовые конструкции языка Pascal для создания функциональных и понятных программ.