яяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяМинистерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образованияя

«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

Подпорин Антон Александрович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

**Цель работы**: освоить синтаксис построения процедур и функций, изучить способы передачи данных в подпрограммы, получить навыки организации минимального пользовательского интерфейса.

1. **Формулировка задания 1. Вариант 17**

**Постановка задачи**

Реализовать программу вычисления площади фигуры, ограниченной кривой

1 \* x^3 + (- 2) \* x^2 + (- 2) \* x + (12) и осью ОХ (в положительной части по оси OY).

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

﻿﻿﻿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.

**Описание алгоритма**

1. Вводим число, которое удовлетворяет условию (от 1 до 3).
2. Если человек выбрал 1, то вводим границы интегрирования и количество прямоугольников.
3. Если выбрал 2, то программа производит вычисления по введенным данным.
4. Если выбрали 3, то выводятся полученные результаты.
5. Выбор чисел продолжается до тех пор, пока человек не выбрал 3.
6. **Схема алгоритма с комментариями**

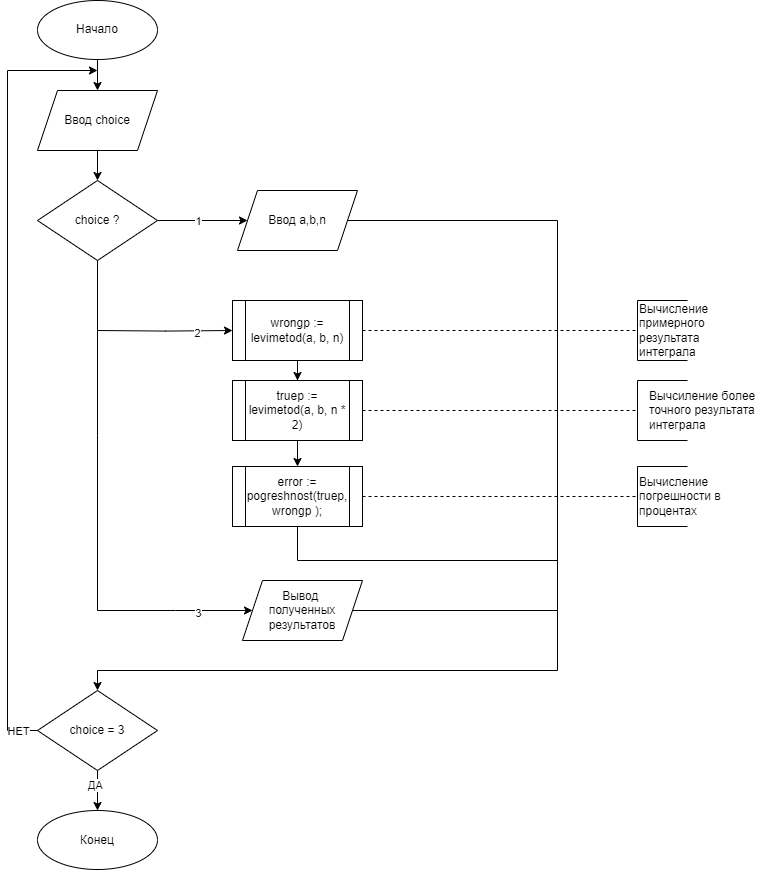
****

Рисунок 1 – Алгоритм решения задачи №1

Изображение выглядит как текст, снимок экрана, Шрифт, Графика

Автоматически созданное описание

Рисунок 2 – Алгоритм выполнения функции func

Изображение выглядит как текст, снимок экрана, Шрифт, графический дизайн

Автоматически созданное описание

Рисунок 3 – Алгоритм выполнения функции levimetod

**Изображение выглядит как текст, снимок экрана, Шрифт, графический дизайн

Автоматически созданное описание**

Рисунок 4 – Алгоритм выполнения функции pogreshnost

**Код программы**

**program** dkr3;

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

**begin**

func := 1 \* x \* x \* x + (- 2) \* x \* x + (- 2) \* x + (12);

**end**;

**function** levimetod(a, b: real; n: integer): real;

**var**

dx, y, x: real;

i: integer;

**begin**

dx := (b - a) / n;

y := 0;

x := a;

**while** x < b **do**

**begin**

y := y + func (x);

x := x + dx;

**end**;

y:= y \* dx;

levimetod := y;

**end**;

**function** pogreshnost(truerazmer, approxrazmer: real): real;

**begin**

pogreshnost := abs((truerazmer - approxrazmer) / truerazmer) \* 100

**end**;

**var**

a, b: real;

n: integer;

wrongp, truep, error: real;

choice: integer;

**begin**

**repeat**

writeln('Выберите действие для вычисления площади фигуры: ');

writeln('1. Ввести значения интегрирования фигуры');

writeln('2. Произвести вычисления по заданным значениям + вычисление погрешности');

writeln('3. Вывод полученных результатов');

readln(choice);

**case** choice **of**

1 : **begin**

writeln('Введите левую границу интегрирования (a): ');

readln(a);

writeln('Введите правую границу интегрирования (b): ');

readln(b);

writeln('Введите количество прямоугольников (n): ');

readln(n);

**end**;

2 : **begin**

wrongp := levimetod(a, b, n);

truep := levimetod(a, b, n \* 2);

error := pogreshnost(truep, wrongp );

**end**;

3 : **begin**

writeln('Приближенная площадь фигуры: ', wrongp:0:4);

writeln('Более правдивая площадь фигуры: ', truep:0:4);

writeln('Примерная погрешность: ', error:0:2, '%');

**end**;

**end**;

**until** choice = 3;

**end**.

**Результат выполнения программы**

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 5 – Результаты тестирования задачи №1

**Вывод**

В процессе работы над домашней контрольной работой мы освоили синтаксис создания процедур и функций, а также изучили методы передачи данных в подпрограммы и приобрели навыки формирования простого пользовательского интерфейса.

Мы также получили знания о таких понятиях, как подпрограмма, процедура и функция. Для разработки алгоритмических схем мы изучили, как правильно составлять письменные алгоритмы для достижения верного решения задач, и сами схемы в программе draw.io — бесплатном онлайн-сервисе для создания и совместного редактирования диаграмм и других визуальных представлений данных.

Тем не менее, в процессе еще возникли некоторые сложности. Например, вместо ожидаемого результата программа выводила значение 0. Это происходило из-за того, что в ней не выполнялся возврат переменной y, которая представляет собой сумму вычислений.

В конечном итоге, благодаря полученным знаниям и исправлению ошибок, нам удалось достичь своей цели — решить задачу методом интегрирования левых прямоугольников и получить новые знания о программировании на языке Pascal.