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**

**«Базовых принципов организации процедур и функций»**

**ПО ДИСЦИПЛИНЕ «Основы алгоритмизации и программирования»**

Выполнил: студент учебной группы

ИСПк-101-51-00

Клыков Денис Борисович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

В отчете должны отображаться:

1. Цель работы
2. Формулировка задания (с вариантом)
3. Описание алгоритма и ответы на вопросы
4. Схема алгоритма с комментариями
5. Код программы
6. Результат выполнения программы
7. Вывод

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

Задание

Написать программу, вычисляющую площадь фигуры ограниченной кривой (вариант 10)

f := 2 \* x \* x \* x + 2 \* x \* x - 2 \* x + 13; ось OX в положительной OY

**program** AreaUnderCurve;

**uses**

crt;

**type**

TFunction = **function**(x: Double): Double;

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

**begin**

f := 2 \* x \* x \* x + 2 \* x \* x - 2 \* x + 13;

**end**;

**function** LeftRectangles(a, b: Double; n: Integer; func: TFunction): Double;

**var**

h, x: Double;

i: Integer;

**begin**

h := (b - a) / n;

Result := 0;

x := a;

**for** i := 0 **to** n - 1 **do**

**begin**

Result := Result + func(x);

x := x + h;

**end**;

Result := Result \* h;

**end**;

**function** EstimateError(a, b: Double; n: Integer; func: TFunction): Double;

**var**

integral1, integral2: Double;

**begin**

integral1 := LeftRectangles(a, b, n, func);

integral2 := LeftRectangles(a, b, 2 \* n, func);

Result := Abs(integral2 - integral1);

**end**;

**procedure** CalculateArea;

**var**

a, b: Double;

n: Integer;

area, error: Double;

**begin**

Write('Введите нижний предел интегрирования (a): ');

ReadLn(a);

Write('Введите верхний предел интегрирования (b): ');

ReadLn(b);

Write('Введите количество разбиений (n): ');

ReadLn(n);

area := LeftRectangles(a, b, n, f);

error := EstimateError(a, b, n, f);

WriteLn('Площадь фигуры: ', area:0:6);

WriteLn('Оценка погрешности: ', error:0:6);

**end**;

**procedure** DisplayMenu;

**begin**

WriteLn('1. Вычислить площадь фигуры');

WriteLn('2. Выход');

Write('Выберите действие: ');

**end**;

**var**

choice: Integer;

**begin**

**repeat**

DisplayMenu;

ReadLn(choice);

**case** choice **of**

1: CalculateArea;

2: WriteLn('Выход из программы...');

**else**

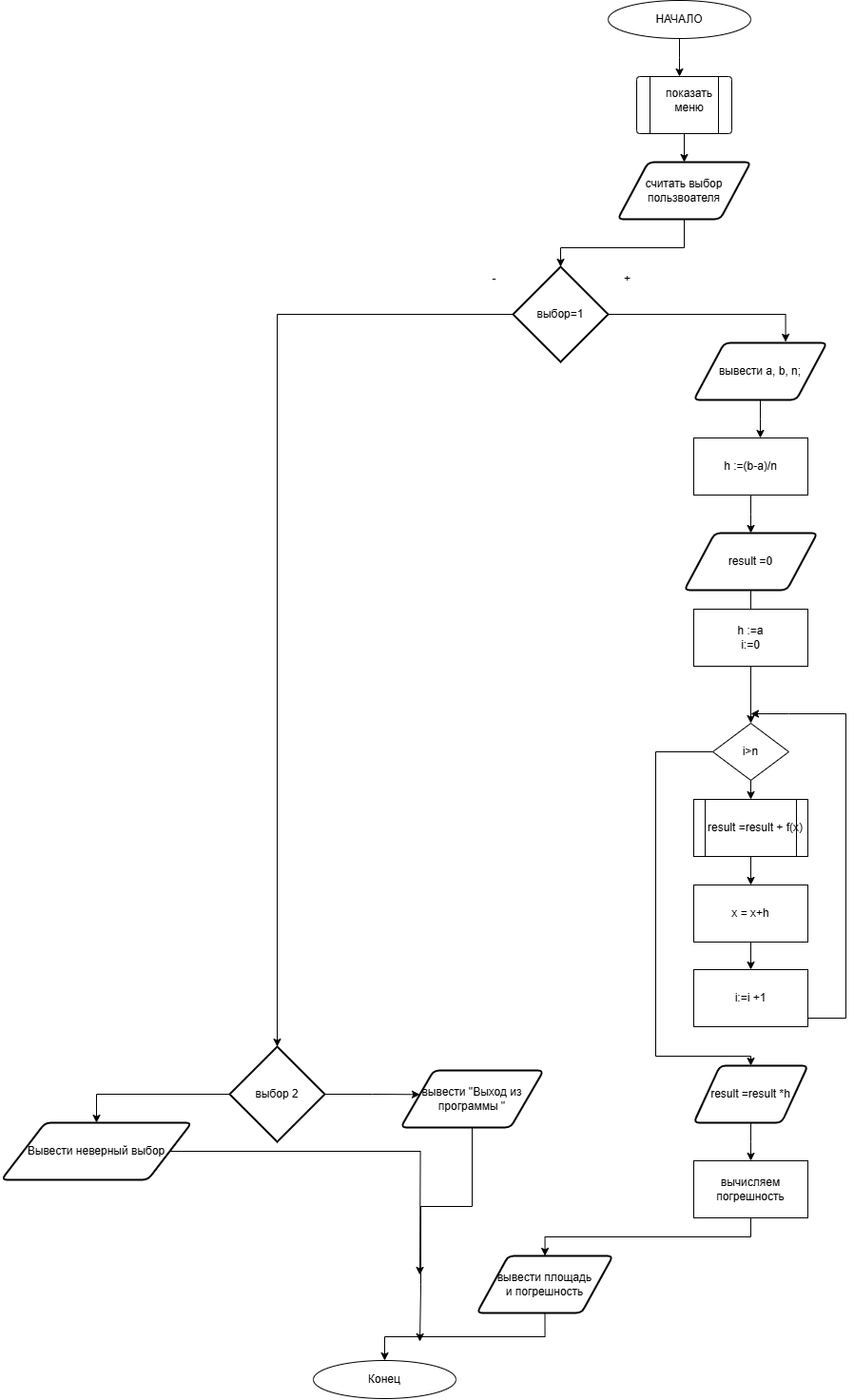
WriteLn('Неверный выбор. Попробуйте снова.');

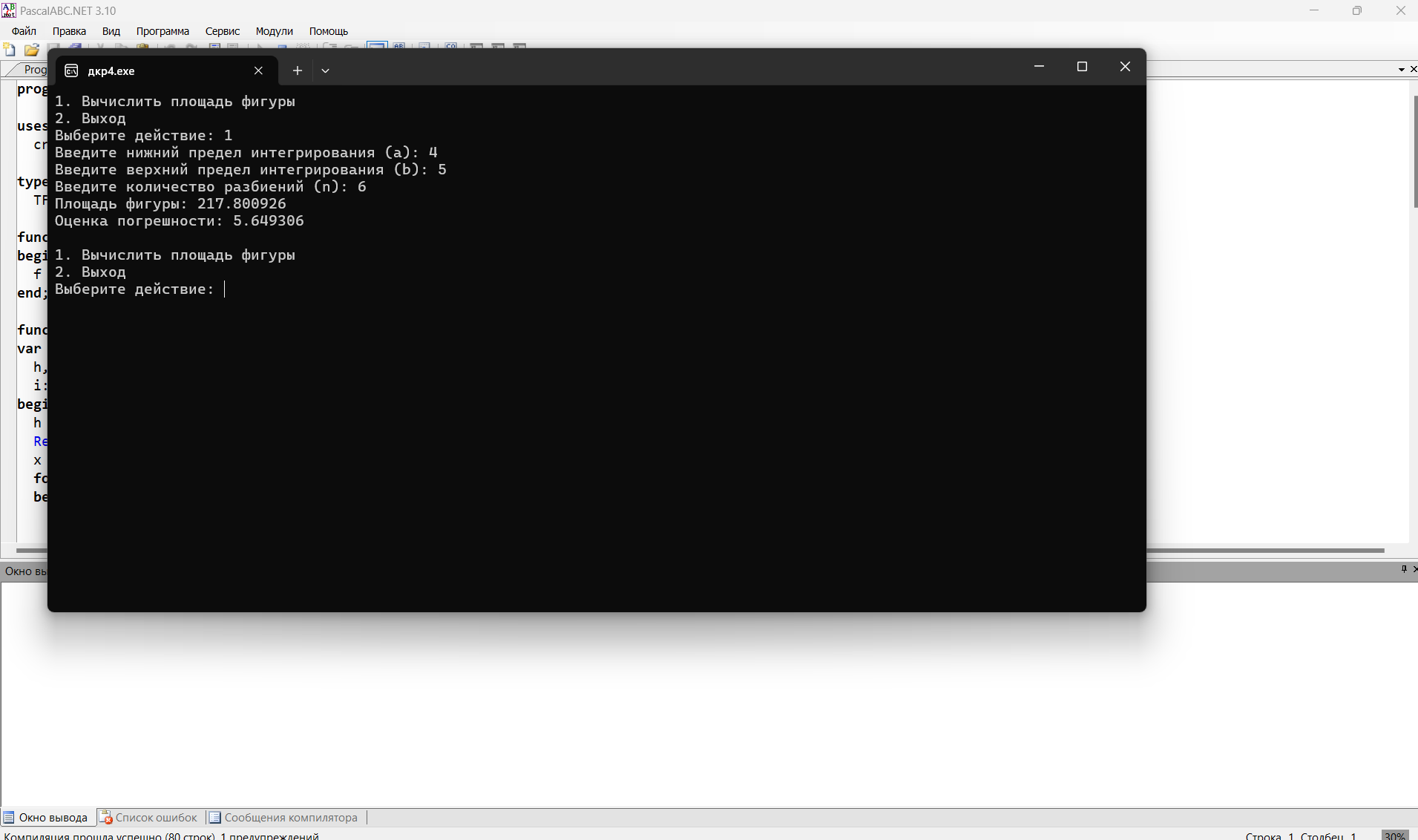
**end**;

WriteLn;

**until** choice = 2;

**end**.





**Вывод**

Была определена математическая функция, площадь под которой необходимо вычислить. Функция была выбрана в соответствии с заданием и реализована в виде отдельной процедуры.

**Метод численного интегрирования**: Для вычисления площади под кривой был использован метод левых прямоугольников. Этот метод позволяет оценить площадь, разбивая область под кривой на несколько прямоугольников и суммируя их площади. В программе была реализована функция, которая принимает пределы интегрирования и количество прямоугольников, а затем вычисляет площадь.

**Взаимодействие с пользователем**: Программа предоставляет пользователю меню для выбора действий. Пользователь может ввести пределы интегрирования и количество прямоугольников, что делает программу интерактивной и удобной в использовании.

**Оценка погрешности**: В программе была реализована возможность оценки погрешности полученного результата. Для этого использовалось увеличение количества прямоугольников, что позволяет сравнить результаты и оценить точность вычислений.

**Обработка ошибок**: Программа включает обработку ошибок, что позволяет избежать сбоев при вводе некорректных данных. Это повышает надежность и удобство использования программы