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»**

Выполнил: студент учебной группы

ИСПк-204-52-00

Исупов Максим Олегович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

**Цель работы**

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

**Формулировка задания**

Вариант: 8

Лабораторная работа №3

Изучение базовых принципов организации процедур и функций

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

1. Реализовать программу вычисления площади фигуры, ограниченной кривой 2 \* x ^ 3 + (-1) \* x ^ 2 + (0) \* x + (2) и осью OX (В положительной части по оси OY).
2. Вычисление определённого интеграла должно выполняться численно, с применением метода средних прямоугольников.
3. Пределы интегрирования вводятся пользователем.
4. Взаимодействие с пользователем должно осуществляться посредством case-меню.
5. Требуется реализовать возможность оценки погрешности полученного результата.
6. Необходимо использовать процедуры и функции там, где это целесообразно.

**Описание алгоритма первой программы**

Необходимо посчитать площадь фигуры с помощью метода средних прямоугольников, погрешность, определённый интеграл.

1. Создаём функцию f(x), которая вычисляет функцию введённого x;
2. Создаём функции integral, которая вычисляет интеграл введённого значения;
3. Создаём функцию square, которая вычисляет площадь разностью пределов интегрирования;
4. Создаём функцию sr\_pryam, с помощью неё вычисляем определённый интеграл с применением метода средних прямоугольников;
5. Создаём две процедуры: pogreshnost (выводит значение погрешности) и new\_predel (вводим пределы интегрирования и количество прямоугольников).
6. Создаём case-меню, с помощью которых пользователь будет запускать функции и процедуры.

**Схема алгоритма первой программы**

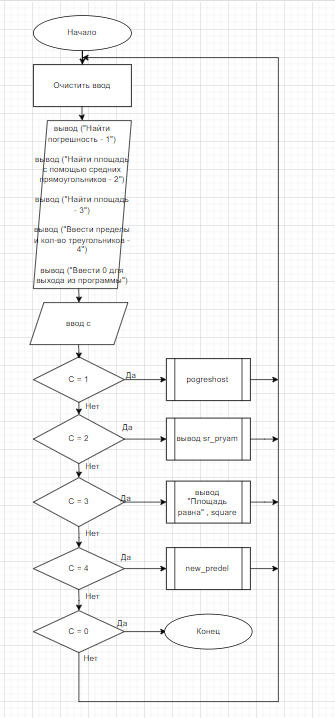


Рисунок 1 – Схема алгоритм задачи

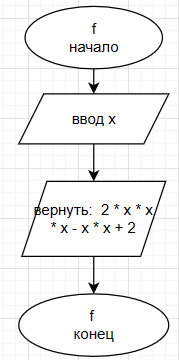
****

Рисунок 2 – схема алгоритма функции f

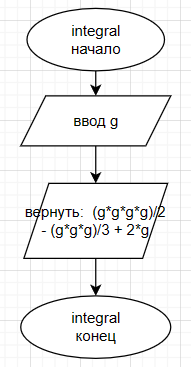


Рисунок 3 – схема алгоритма функции integral

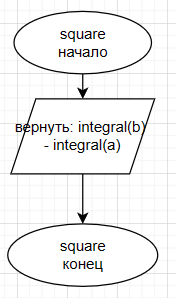
****

Рисунок 4 – схема алгоритма функции square

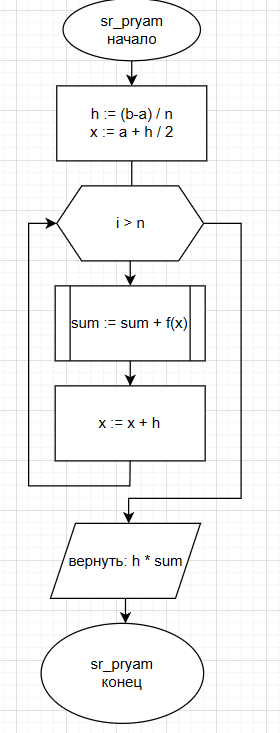


Рисунок 5 – схема алгоритма функции sr\_pryam

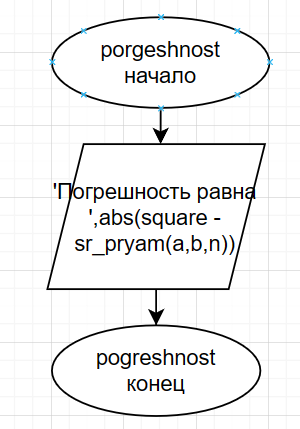
****

Рисунок 6 – схема алгоритма процедуры pogreshnost

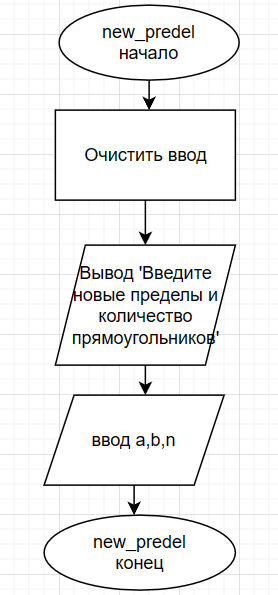


Рисунок 7 – схема алгоритма процедуры new\_predel

**Код программы**

**uses** CRT;

**var** i,c,a,b,n : integer; x,y, integralsum : real;

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

**begin**

Result := 2 \* x \* x \* x - x \* x + 2;

**end**;

**function** integral(g : real) : real;

**begin**

Result := (g\*g\*g\*g)/2 - (g\*g\*g)/3 + 2\*g

**end**;

**function** square : real;

**begin**

Result := integral(b) - integral(a);

**end**;

**function** sr\_pryam(a, b: Real; n: Integer): Real;

**var**

h, x, sum: Real;

i: Integer;

**begin**

h := (b - a) / n;

x := a + h / 2;

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

**begin**

sum := sum + f(x);

x := x + h;

**end**;

Result := h \* sum;

**end**;

**procedure** pogreshnost;

**begin**

Writeln('Погрешность равна ',abs(square - sr\_pryam(a,b,n)));

readln();

**end**;

**procedure** new\_predel;

**begin**

ClrScr;

writeln('Введите новые пределы и количество прямоугольников: ');

readln(a,b,n);

**end**;

**begin**

**repeat**

ClrScr;

writeln('Найти погрешность - 1');

writeln('Найти площадь с помощью средних прямоугольников - 2');

writeln('Найти площадь - 3');

writeln('Ввести пределы и количество прямоугольников- 4');

writeln('Ввести 0 для выхода из программы');

readln(c);

**case** c **of**

1 : pogreshnost;

2 : **begin** print(sr\_pryam(a,b,n)); **var** ch : char; **repeat** readln(ch) **until** ch = #13; **end**;

3 : **begin** write('Площадь равна '); writeln(square); readln(); **end**;

4 : new\_predel;

**end**;

**until** c = 0;

**end**.

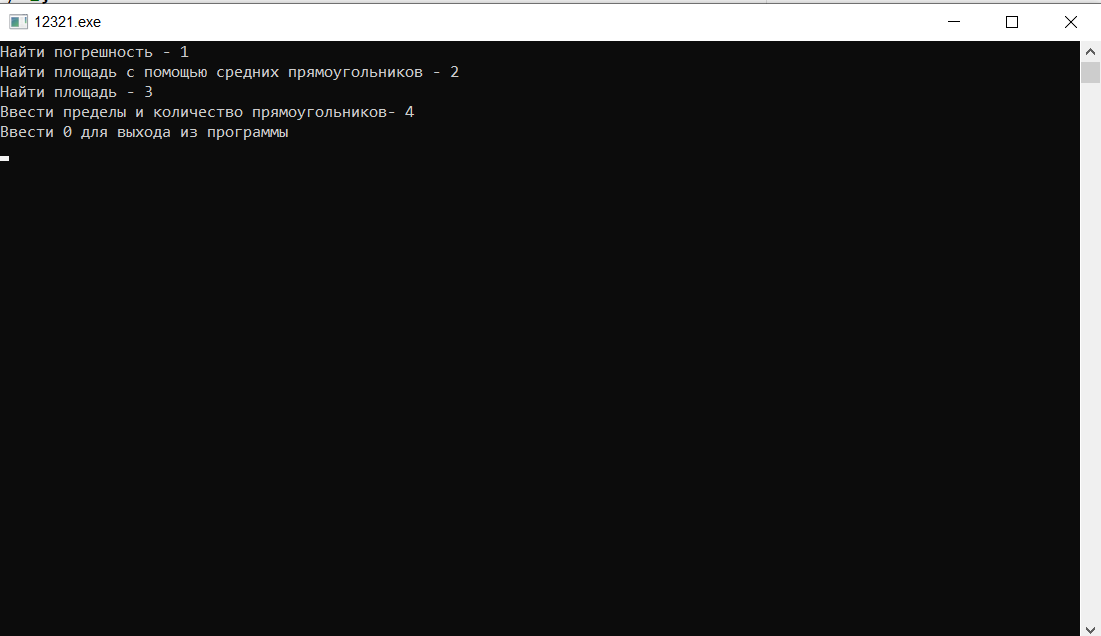
****

Рисунок 8 – Результат выполнения программы 1

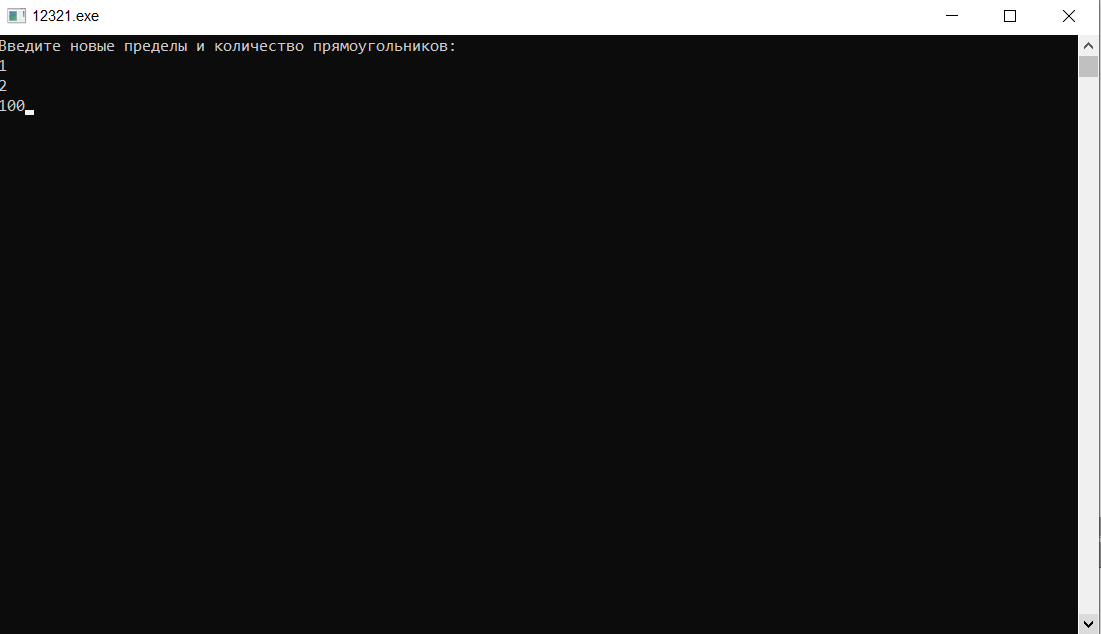


Рисунок 9 – Результат выполнения программы 2

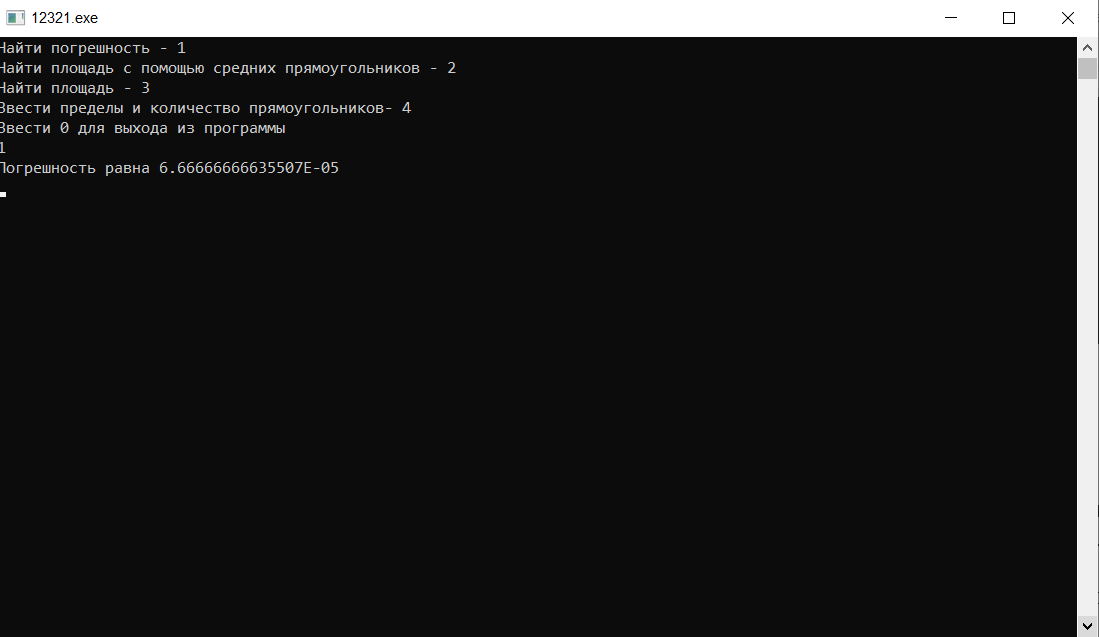
****

Рисунок 10 – Результат выполнения программы 3

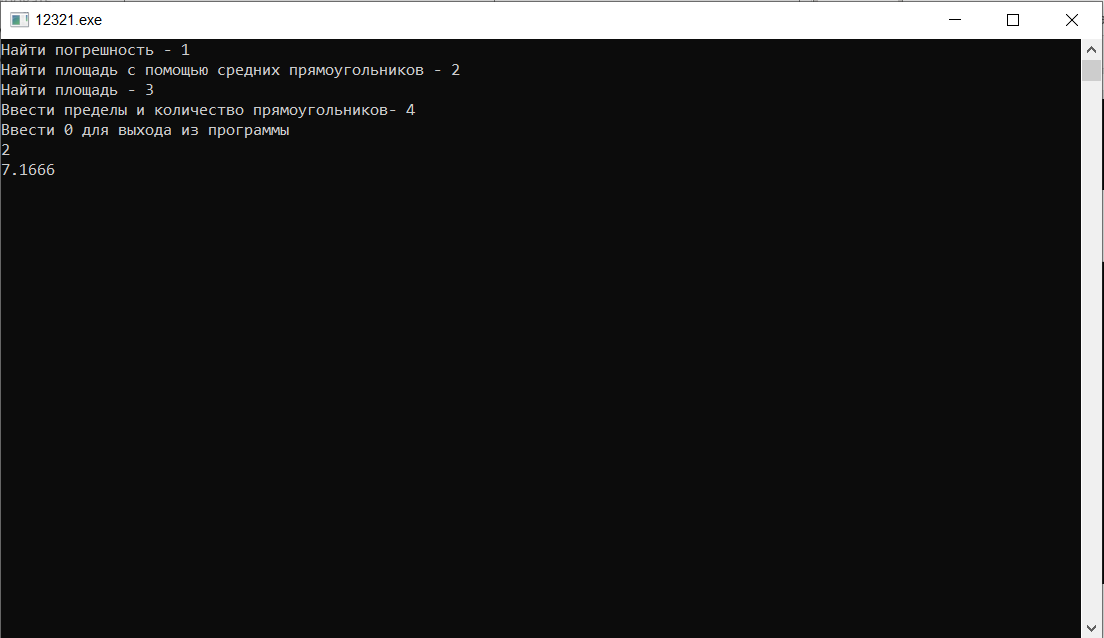


Рисунок 11 – Результат выполнения программы 4

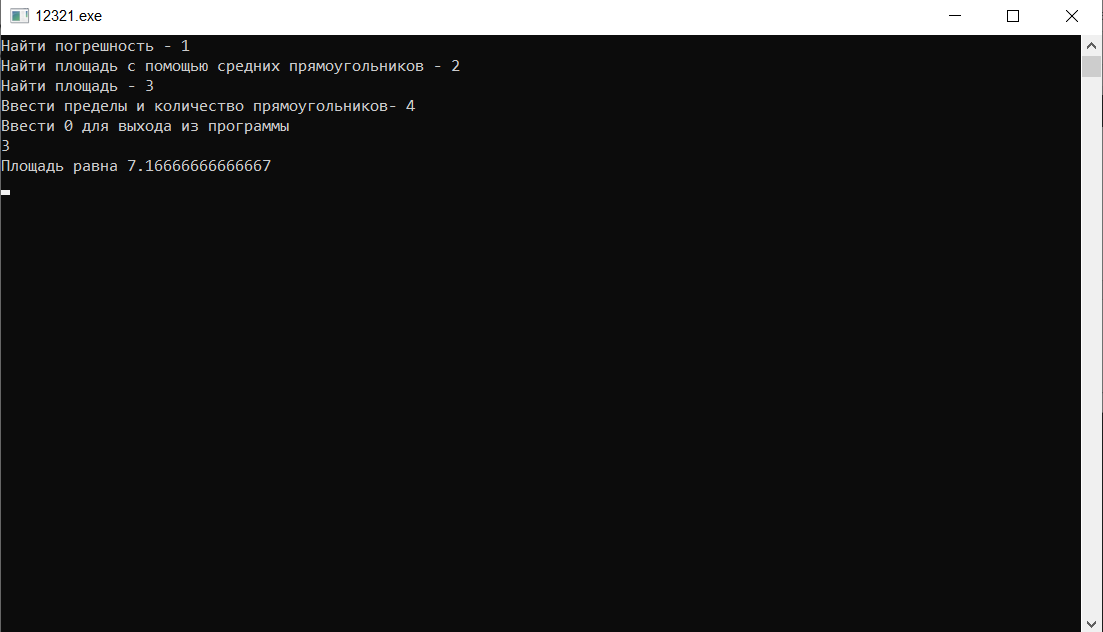


Рисунок 11 – Результат выполнения программы 5

**Вывод**

В ходе выполнения домашней лабораторной работы мы закрепили знания полученные при выполнение лабораторной работы №9-10. При написании кода задачи были использованы конструкции case-меню, изученные при выполнение первых лабораторных работ. При написании и использовании функций возникали сложности с типами данных. Так же на практике была изучена библиотека CRT.

Так же для написания схемы алгоритма был использован интернет -ресурс «draw.io». Его использование упрощает написание схем алгоритмов, уменьшает вероятность ошибок, а также в случае пропуска какого-либо цикла его можно без каких-либо проблем вставить в любое место схемы. Опыт написания схем-алгоритмов будет полезен в будущем для решения новых задач а так же понимание программ, написанных другими пользователями