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-205-52-00

Веселов Алексей Денисович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

1 Цель работы

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

2 Формулировка задания (с вариантом)

Вариант: 4

Задание:

1. Реализовать программу вычисления площади фигуры, ограниченной кривой 1 \* x ^ 3 + (2) \* x ^ 2 + (-2) \*x + (8) и осью OY (в положительной части по оси OX).

2. Вычисление определенного интеграла должно выполняться численно, с применением метода Симпсона.

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.

3 Описание алгоритма

Задача не из простых, для удобства разделим её на 6 частей. Одной для взаимодействия с пользователем (case меню), и 5 вычислительных частей.

Первая часть, она же продпрограмма, вычисление y = x ^ 3 + (2) \* x ^ 2 + (-2) \*x + (8). В дальнейшем она будет использоваться в нашем методе.

Вторая подпрограмма – вычисление первообразной от функции.

Третья подпрограмма находит крайнюю левую точку пересечения графика с осью.

Четвёртая подпрограмма — вычисление Формулы Ньютона-Лейбница.

Пятая подпрограмма — вычисление методом Симпсона, тут нужно ввести количество промежутков, чем их больше, тем точнее значение получиться. Далее путём не хитрых махинаций “рисуем” параболы и считаем их площадь.

Шестая подпрограмма — это кейс меню для взаимодействия с пользователем. В нем есть всё выше перечисленные подпрограммы.

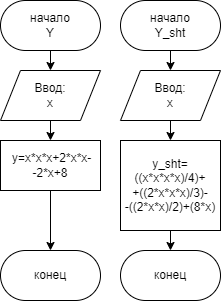
4. Схема алгоритма с комментарием

Рисунок 1 Процедуры, Y и Y\_sht

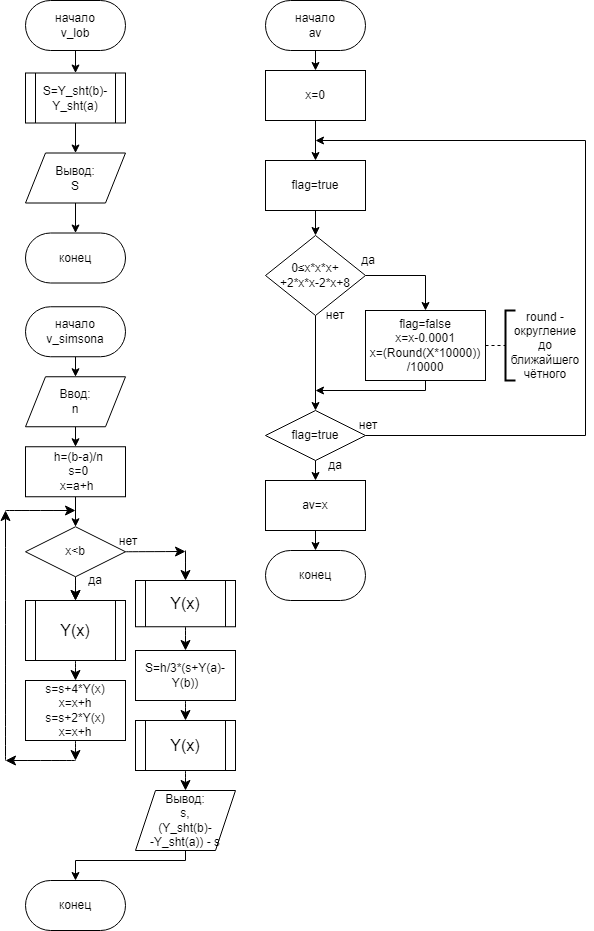


Рисунок 2 Процедуры v\_lob, av, v\_simsona

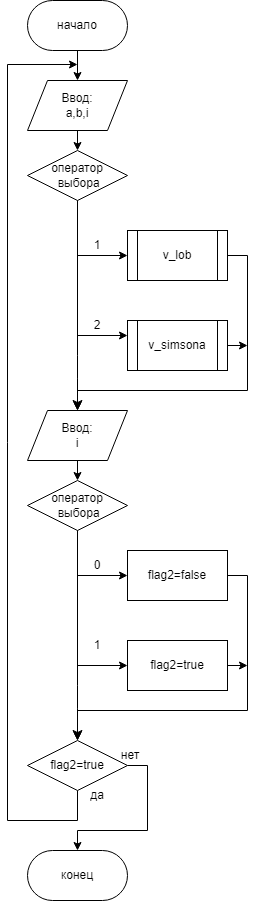


Рисунок 3 case меню

5 Код программы

**program** z1; ///f=x\*x\*x+2\*x\*x-2\*x+8

**uses** crt;

**var**

a,b,S : Real;

h,x : real;

i : integer;

flag2 : boolean;

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

**begin**

y:=x\*x\*x+2\*x\*x-2\*x+8;

**end**;

**function** Y\_sht (x:real) :real;

**begin**

y\_sht:=((x\*x\*x\*x)/4)+((2\*x\*x\*x)/3)-((2\*x\*x)/2)+(8\*x);

**end**;

**function** av:real;

**var** flag:boolean;

x:real;

**begin**

x:=0;

**repeat**

flag:=true;

**if** 0<=x\*x\*x+2\*x\*x-2\*x+8 **then begin**

flag:=false;

x-=0.0001;

X:=(Round(X\*10000))/10000;

**end**;

**until** flag=true;

av:=x;//-3.3251

**end**;

{procedure pregel(var a,b :real);

var flag:boolean;

begin

repeat

flag:=true;

writeln('Введите пределы интегрирования ф-ии [-3.3251; ]');

read(a,b);

if (av<a) and (a>0) then begin

flag:=false;

writeln('предел а недопустим');

end;

if (av<b) and (b>0) then begin

flag:=false;

writeln('предел b недопустим');

end;

if a>b then begin

flag:=false;

writeln('предел a и b недопустим');

end;

until flag=true;

end;}

**procedure** v\_lob;

**begin**

//в лоб

writeln('Вы выбрали бызовый метод');

S:=Y\_sht(b)-Y\_sht(a);

writeln('Интеграл = ',s);

**end**;

**procedure** v\_simsona;

**var** n : integer;

**begin**

//в симсонa

writeln('Вы выбрали метод Симпсона');

writeln('На сколько частей делим промежуток?');

Readln(n);

h:=(b-a)/n;// мб спросить на сколько частей?(1000=n)

s:=0;

x:=a+h;

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

s:=s+4\*Y(x);

x:=x+h;

s:=s+2\*Y(x);

x:=x+h;

**end**;

S:=h/3\*(s+Y(a)-Y(b));

writeln('Интеграл = ',s);

writeln('Погрешность = ', (Y\_sht(b)-Y\_sht(a)) - s);

**end**;

**begin**

**repeat**

flag2:=true;

writeln('Нахождение интеграла y=x^(3)+2x^(2)-2\*x+8');

{pregel(a,b);}

writeln('Введите пределы интегрирования ф-ии [-3.3251;∞]');

read(a,b);

writeln('Выберите метод интегрирования');

writeln('1:базовый метод интегрирования');

writeln('2:метод Симпсона');

readln(i);

**case** i **of**

0:flag2:=false;

1:v\_lob;

2:v\_simsona;

**end**;

writeln('Хотите повторить ввод?');

writeln('1:Да');

writeln('2:Нет');

readln(i);

**case** i **of**

1:flag2:=false;

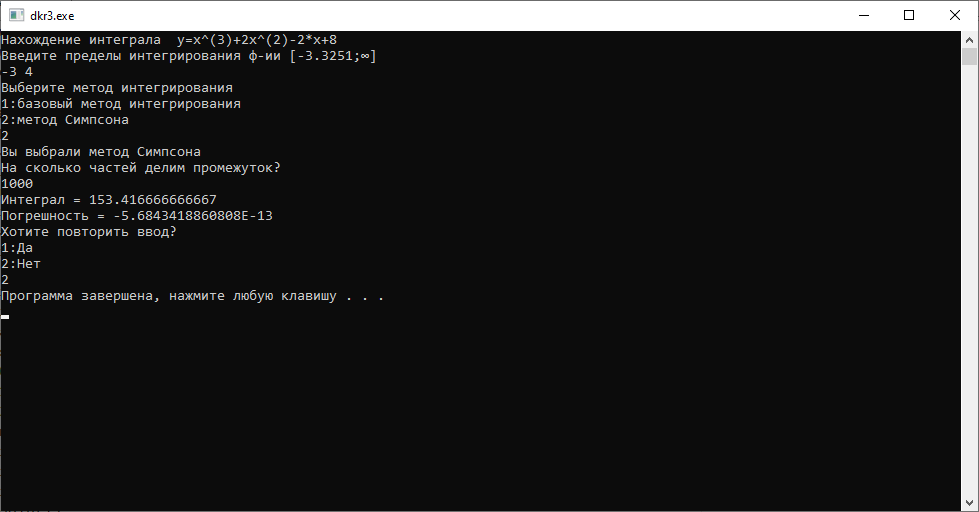
2:flag2:=true;

**end**;

**until** flag2=true ;

**end**.

6 Результат выполнения программы



7 Вывод

Данная работа оказалась сложной.

Был изучен метод Симпсона также называемый метод парабол. Изначально было не понятно, как реализовать вычисление этим методом. Но решение было найдено благодаря сети интернет.

Так же была реализована, затем удалена подпрограмма для ввода корректных ограничений, но так как такого задания не дано, было принято решение удалить данную часть кода (но она ещё осталась в виде пояснения).

Изначально была идея не писать количество промежутков, но как выяснилось это потребуется в следующей домашней контрольной работе.

Само собой цель работы была достигнута.