Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №4**

**«РАБОТА В ГРАФИЧЕСКОМ РЕЖИМЕ»**

**ПО ДИСЦИПЛИНЕ «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЕ»**

Выполнил: студент учебной группы

ИСПк-203-52-00

Вальковская Арина Константиновна

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

Содержание

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

**1.Цель работы:**

Освоить принципы работы в графическом режиме, получить базовые навыки взаимодействия с графическими примитивами.

**2. Формулировка задания (Вариант:4)**

1. Дополнить программу, реализованную в ходе предыдущей лабораторной работы, режимом визуализации.

2. Предусмотреть возможность вывода кривой, ограничивающей фигуру, на координатную плоскость.

3. реализовать следующие возможности и элементы: масштабирование графика, подписи на осях, вывод информации о задании.

4. Реализовать не менее двух возможностей из представленных: независимое масштабирование по осям, штриховка вычисляемой площади, визуализация численного расчета интеграла.

**3. Описание алгоритма**

1. Пользователю предлагается ввести нижний и верхний пределы интегрирования, а также количество подотрезков.

2. Программа использует введенные значения для вычисления ширины каждого подотрезка (шаг интегрирования).

3. Затем программа вычисляет значение функции на каждом конце подотрезка и использует эти значения для вычисления площадей трапеций.

4. Сумма площадей всех трапеций дает приближенное значение определенного интеграла функции.

5. Для визуализации функции и вычисленного интеграла используется графическое представление. Программа рисует график функции на заданном интервале и затем рисует трапеции, которые используются для приближенного вычисления интеграла.

6. Затем программа размещает метки на осях координат и выводит значение вычисленного интеграла.

**4. Схема алгоритма с комментариями**

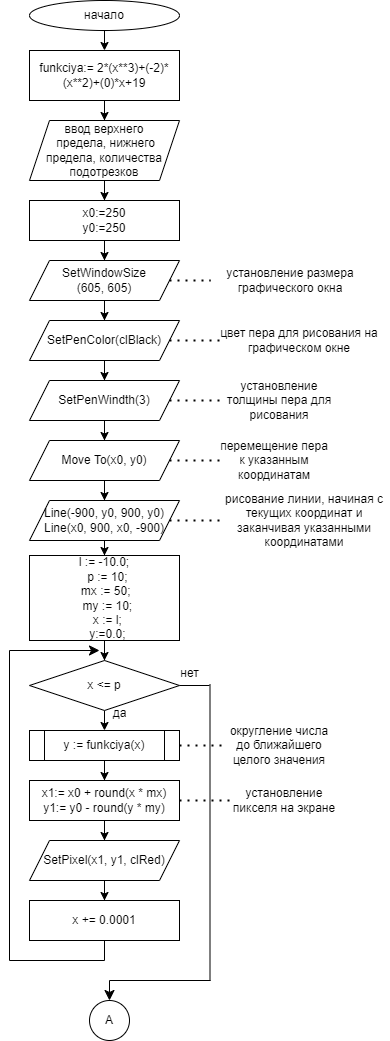
****

Рисунок 1 – Схема алгоритма 1/2

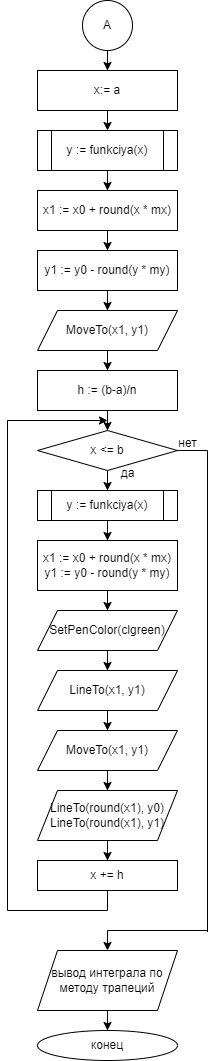


Рисунок 2 – Схема алгоритма 2/2

**5. Код программы**

uses graphABC;

function funkciya(x:real):real;

begin

funkciya:= 2\*(x\*\*3)+(-2)\*(x\*\*2)+(0)\*x+19;

end;

function trap(a, b: Real; n: Integer):real;

var

h, x,sum: Real;

begin

h:= (b-a)/n;

sum := (funkciya(a) + funkciya(b)) / 2;

x:= a+h;

for var i:=1 to n-1 do begin

sum := sum + funkciya(x);

x := x + h;

end;

trap:= h \* sum;

end;

var x1:=0;

var y1:=0;

begin

write('Введите нижний предел: ');

var a:=ReadReal();

writeln(a);

write('Введите верхний предел: ');

var b:=ReadReal();

writeln(b);

write('Введите количество подотрезков : ');

var n:=ReadInteger();

writeln(n);

var sum := trap(a, b, n);

var x0 := 250;

var y0 := 250;

SetWindowSize(605, 605);

SetPenColor(clBlack);

SetPenWidth(3);

MoveTo(x0, y0);

Line(-900, y0, 900, y0);

Line(x0, 900, x0, -900);

SetFontColor(clBlack);

TextOut(x0 + 340, y0 - 20, 'x');

TextOut(x0-12, y0 - 250, 'y');

var l := -10.0;

var p := 10;

var mx := 50;

var my := 10;

var x := l;

var y:=0.0;

while x <= p do begin

y := funkciya(x);

x1 := x0 + round(x \* mx);

y1 := y0 - round(y \* my);

SetPixel(x1, y1, clRed);

x += 0.001;

end;

x := a;

y := funkciya(x);

x1 := x0 + round(x \* mx);

y1 := y0 - round(y \* my);

MoveTo(x1, y1);

var h:=(b-a)/n;

while x <= b do begin

y := funkciya(x);

x1 := x0 + round(x \* mx);

y1 := y0 - round(y \* my);

SetPenColor(clgreen);

LineTo(x1, y1);

MoveTo(x1, y1);

LineTo(round(x1), y0);

LineTo(round(x1), y1);

x += h;

end;

var stepXLabel := 1.0;

var stepYLabel := 10.0;

for var i := Round(l) to Round(p) do

begin

var xPos := x0 + Round(i \* mx);

TextOut(xPos, y0 + 10, i);

end;

for var j := -200 to 200 step Round(stepYLabel\*my) do

begin

var yPos := y0 - j;

TextOut(x0 - 20, yPos, j/10);

end;

writeln('');

writeln('Значение интеграла по методу трапеций: ', sum);

end.

**6. Результат выполнения программы**

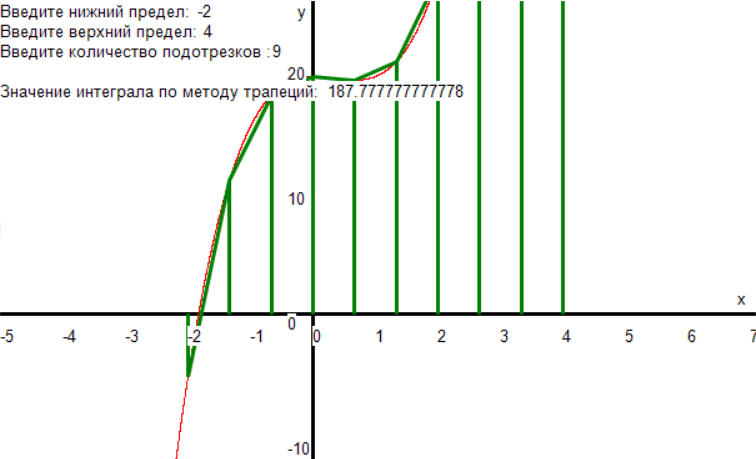
****

Рисунок 3 – Результат выполнения программы

**7. Вывод**

При выполнении данной домашней контрольной работы мы узнали много нового, полезного для нас в будущем. Мы познакомились с модулем GraphABC и с его основными примитивами. Нам стало понятно, как представить функцию в виде графика, так же мы узнали, как выделить на графике необходимую область с помощью штриховки. Мы научились делать масштабирование графика, поняли, что необходимо для того, чтобы были подписи на осях.

В процессе выполнения отчета, для создания блок схемы мы работали с программой draw.io. Draw.io — это бесплатный онлайн-сервис, который помогает создавать алгоритмы, прототипы, инфографику и диаграммы любого вида. Чаще всего его используют именно для построения диаграмм. Сервисом пользуются разные IT-специалисты — от бизнес-аналитиков до разработчиков и дизайнеров.