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 программирования»**

Выполнил: студент учебной группы

ИСПк-208-52-00

Солянов Арсений Алексеевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

В отчете должны отображаться:

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

Цель работы:

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

Задание:

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, штриховка вычисляемой площади, визуализация численного расчетного интервала.

Описание алгоритма:

1.Функция, которая вычисляет значение функции

2.Цикл repeat который выводит визуализированное мен.

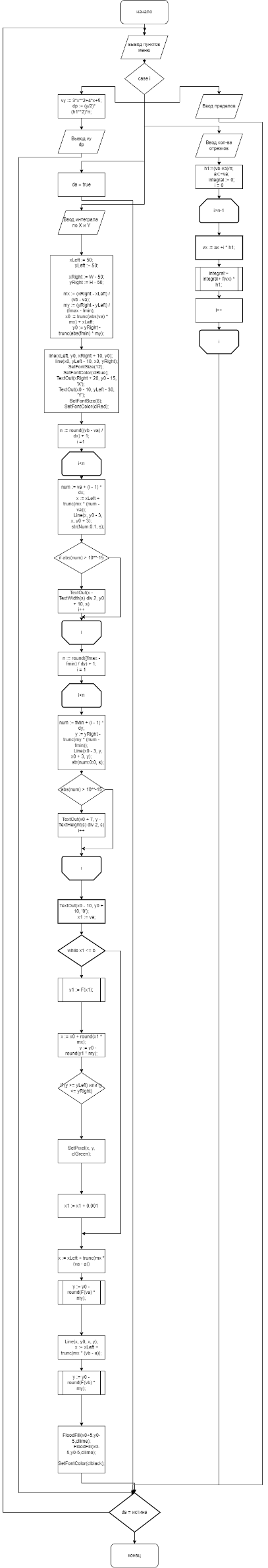
3.Вычисления значения площади по методу левых прямоугольников

4.Вычисление погрешности

5.Выбор параметров визуализации

6.Визуализация графика

Схема алгоритма:



Код программы:

**program** DKR4;

**uses** graphABC;

**const**

W =800; H = 500;

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

**begin**

f:=x\*\*3+(2)\*x\*\*2+(5)\*x+(1);

**end**;

**var**

va,vb,dx,h1,vx,ax,integral,dy,fmin,fmax,dp,a,b,mx,my,num,vy,x1,y1:real;

n,i,xLeft,yLeft,yRight,xRight,x0,y0,x,y:integer;

da:boolean;

s:string;

**begin**

**repeat**

writeln('1.Ввести пределы вычисления интеграла ');

writeln('2.Ввести кол-во отрезков ');

writeln('3.Вычислить интеграл ');

writeln('4.Выбор параметров визуализации ');

writeln('5.Визуализация графика ');

writeln('0.Exit');

write('Выберите! = '); readln(i);

da:=false;

**case** i **of**

1:

**begin**

Write('Введите нижний предел интегрирования = '); readln(va);

Writeln('Введите верхний предел интегрирования = '); readln(vb);

**end**;

2:

**begin**

Write('Введите кол-во отрезков = ');readln(n);

h1:=(vb-va)/n;

ax:=va;

integral := 0;

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

vx := ax +i \* h1;

integral:= integral+ f(vx) \* h1;

**end**;

**end**;

3:

**begin**

vy := 3\*x\*\*2+4\*x+5;

dp := (y/2)\*(h1\*\*2)\*n;

Writeln(dp);

Writeln(integral);

**end**;

4:

**begin**

Writeln('интервал по Х; a и b должны нацело делится на dx');

Writeln('a: '); readln(a);

writeln('b: '); readln(b);

writeln('dx: '); readln(dx);

Writeln('интервал по Y; fmin и fmax должны нацело делится на dy');

writeln('fmin: '); readln(fmin);

writeln('fmax: '); readln(fmax);

writeln('dy: '); readln(dy);

**end**;

5:

**begin**

SetWindowSize(W,H);

xLeft := 50;

yLeft := 50;

xRight := W - 50;

yRight := H - 50;

mx := (xRight - xLeft) / (vb - va);

my := (yRight - yLeft) / (fmax - fmin);

x0 := trunc(abs(va) \* mx) + xLeft;

y0 := yRight - trunc(abs(fmin) \* my);

line(xLeft, y0, xRight + 10, y0);

line(x0, yLeft - 10, x0, yRight);

SetFontSize(12);

SetFontColor(clBlue);

TextOut(xRight + 20, y0 - 15, 'X');

TextOut(x0 - 10, yLeft - 30, 'Y');

SetFontSize(8);

SetFontColor(clRed);

n := round((vb - va) / dx) + 1;

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

**begin**

num := va + (i - 1) \* dx;

x := xLeft + trunc(mx \* (num - va));

Line(x, y0 - 3, x, y0 + 3);

str(Num:0:1, s);

**if** abs(num) > 10\*\*-15 **then**

TextOut(x - TextWidth(s) **div** 2, y0 + 10, s)

**end**;

n := round((fmax - fmin) / dy) + 1;

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

**begin**

num := fMin + (i - 1) \* dy;

y := yRight - trunc(my \* (num - fmin));

Line(x0 - 3, y, x0 + 3, y);

str(num:0:0, s);

**if** abs(num) > 10\*\*-15 **then**

TextOut(x0 + 7, y - TextHeight(s) **div** 2, s)

**end**;

TextOut(x0 - 10, y0 + 10, '0');

x1 := va;

**while** x1 <= b **do**

**begin**

y1 := F(x1);

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

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

**if** (y >= yLeft) **and** (y <= yRight) **then** SetPixel(x, y, clGreen);

x1 := x1 + 0.001

**end**;

x := xLeft + trunc(mx \* (va - a));

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

Line(x, y0, x, y);

x := xLeft + trunc(mx \* (vb - a));

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

Line(x, y0, x, y);

FloodFill(x0+5,y0-5,cllime);

FloodFill(x0-5,y0-5,cllime);

SetFontColor(clblack);

**end**;

0:

da := true;

**end**;

writeln();

**if not** da **then begin**

readln();

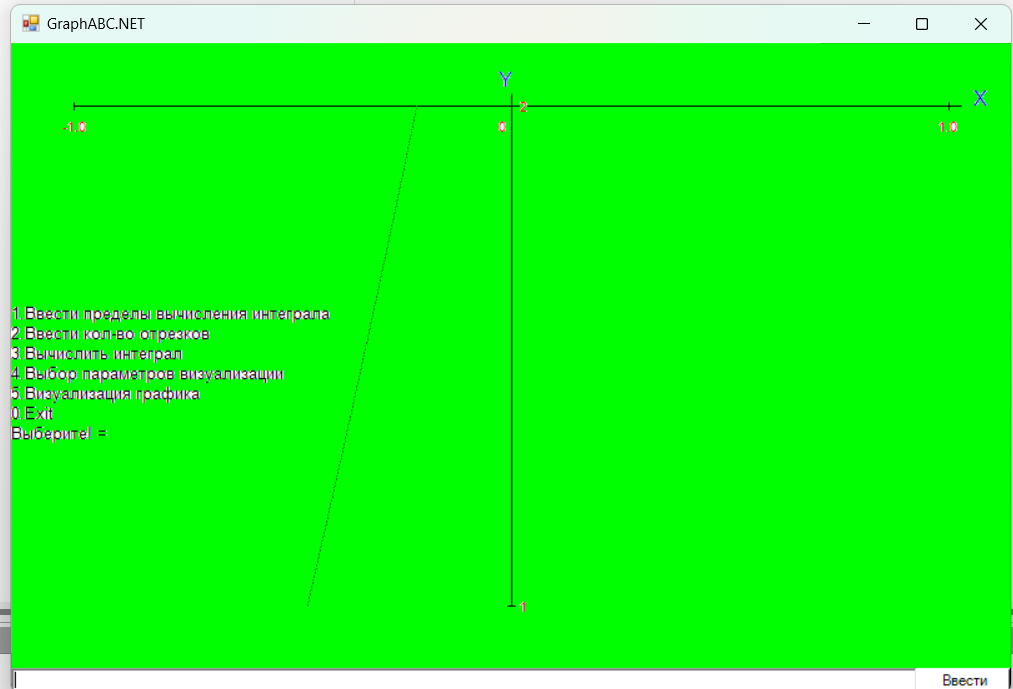
writeln();

ClearWindow;

**end**;

**until** da;

**end**.

Результат выполнения программы: 

Вывод.

В данной работе были сделаны дополнения к предыдущей лабораторной работе, а именно был добавлен режим визуализации функции. Так же были предусмотрены возможности вывода кривой, огранивающие фигуру, на координатную плоскость. Так же были реализованы функции: масштабирование графика, подписи на осях координат, вывод информации о задании.

Проблемы в данной работе могут быть такие как например некорректный ввод данных(ввод не числовых значений).