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»**

Выполнила: студент учебной группы

ИСПк-202-52-00

Долгополов Ярослав Максимович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

**Цель работы**: освоить принципы работы в графическом режиме; получить базовые навыки взаимодействия с графическими примитивами.

1. **Формулировка задания. Вариант 8**

**Постановка задачи**

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.
5. **Описание алгоритма**
   1. Описываем функцию, по которой будем строить график: f = 2x3 + (-1)\*x2 + (-4)\*x + 5.
   2. Строим оси Ox и Oy.
   3. Циклом закрашиваем пиксели, на которых расположен график.
   4. Дан метод средних прямоугольников: с помощью цикла строим прямоугольники на интервале от **a** до **b** с шагом **(b - a)/n**. Циклом штрихуем каждый прямоугольник.
6. **Схема алгоритма с комментариями**

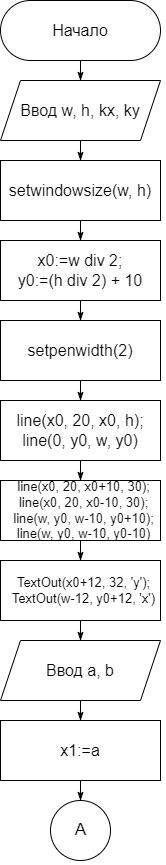
****

Рисунок 1 - Алгоритм решения задачи. Часть 1

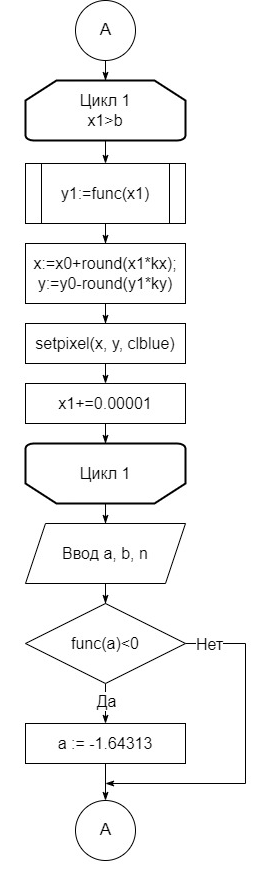
****

Рисунок 2 - Алгоритм решения задачи. Часть 2

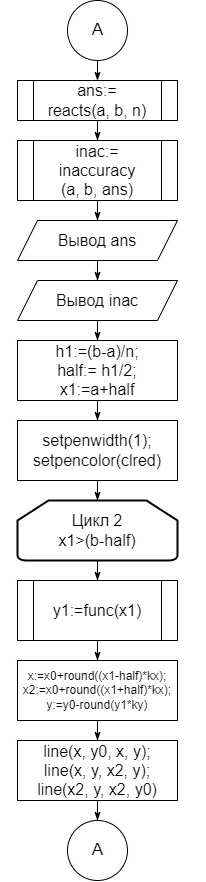


Рисунок 3 - Алгоритм решения задачи. Часть 3

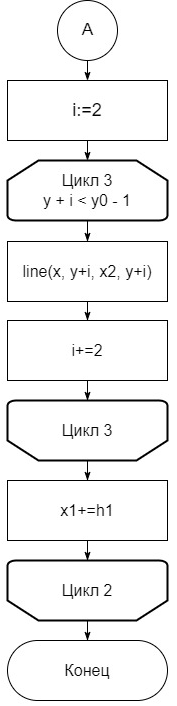
****

Рисунок 4 - Алгоритм решения задачи. Часть 4

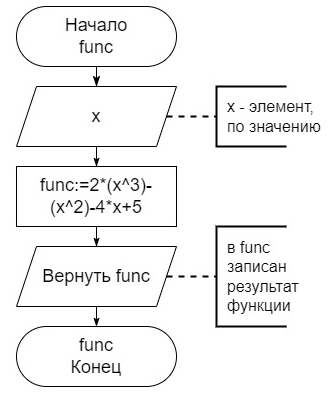


Рисунок 5 - Алгоритм выполнения функции func

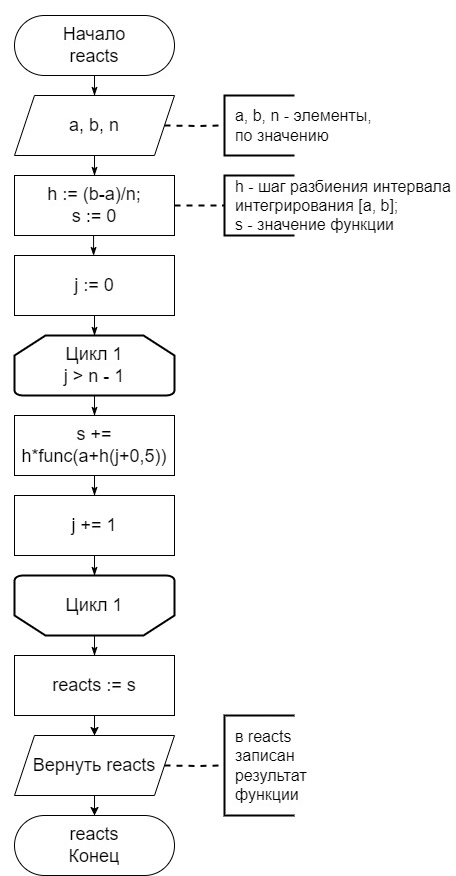


Рисунок 6 - Алгоритм выполнения функции reacts

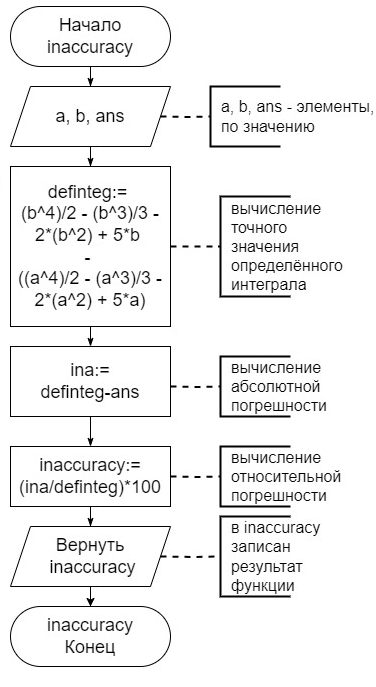


Рисунок 4 - Алгоритм выполнения функции inaccuracy

1. **Код программы**

**uses** GraphABC;

**var**

x0, y0, w, h, x, y, x2, n, i: integer;

x1, y1, a, b, ky, kx, half, h1, ans, inac: real;

**Function** func(x: real): real;

**begin**

func:=2\*x\*sqr(x)-sqr(x)-4\*x+5;

**end**;

**Function** reacts(a, b: real; n: integer): real;

**var**

h, s, j: real;

**begin**

h := (b - a)/n;

j:=0.0;

**while** j <= (n-1) **do**

**begin**

s+=h\*func(a+h\*(j+0.5));

j+=1

**end**;

reacts:=s

**end**;

**Function** inaccuracy(a, b, ans: real): real;

**var**

definteg, ina: real;

**begin**

definteg:=(sqr(sqr(b))/2) - ((b\*sqr(b))/3) - 2\*sqr(b) + 5\*b - ((sqr(sqr(a))/2) - ((a\*sqr(a))/3) - 2\*sqr(a) + 5\*a);

ina:=definteg-ans;

inaccuracy:=(abs(ina)/definteg)\*100

**end**;

**begin**

writeln('Введите размеры окна и коэффициенты масштабирования по осям Ox и Oy');

readln(w, h, kx, ky);

setwindowsize(w, h);

x0:=w **div** 2;

y0:=(h **div** 2) + 10;

setpenwidth(2);

line(x0, 20, x0, h);

line(0, y0, w, y0);

line(x0, 20, x0+10, 30);

line(x0, 20, x0-10, 30);

line(w, y0, w-10, y0+10);

line(w, y0, w-10, y0-10);

TextOut(x0+12, 32, 'y');

TextOut(w-12, y0+12, 'x');

TextOut(2, 0, 'Вычислить площадь фигуры ограниченной кривой 2\*x^3 - x^2 - 4\*x + 5 и осью OX (в положительной части по оси OX) методом средних прямоугольников.');

writeln('Введите границы построения графика');

readln(a,b);

x1:=a;

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

**begin**

y1:=func(x1);

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

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

setpixel(x, y, clblue);

x1+=0.00001

**end**;

writeln('Введите границы интегрирования фигуры');

readln(a, b, n);

**if** func(a)<0 **then** a:=-1.64313;

ans:=reacts(a, b, n);

inac:=inaccuracy(a, b, ans);

writeln('Приблизительный ответ: ', ans);

writeln('Относительная погрешность: ', inac);

h1:=(b-a)/n;

half:= h1/2;

x1:=a+half;

setpenwidth(1);

setpencolor(clred);

**while** x1 <= (b-half) **do**

**begin**

y1:=func(x1);

x:=x0+round((x1-half)\*kx);

x2:=x0+round((x1+half)\*kx);

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

line(x, y0, x, y);

line(x, y, x2, y);

line(x2, y, x2, y0);

i:=2;

**while** y+i<y0-1 **do**

**begin**

line(x, y+i, x2, y+i);

i+=2

**end**;

x1+=h1

**end**;

setpencolor(clgreen);

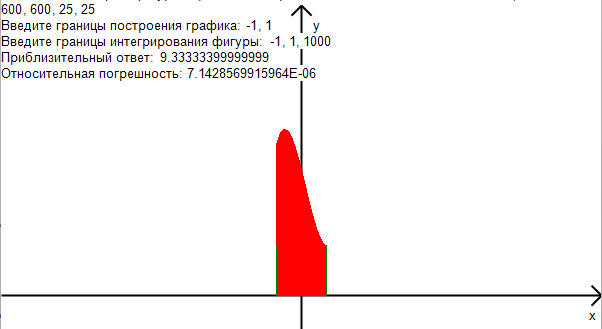
line(x0+round(a\*kx), y0, x0+round(a\*kx), y0-round(func(b)\*ky));

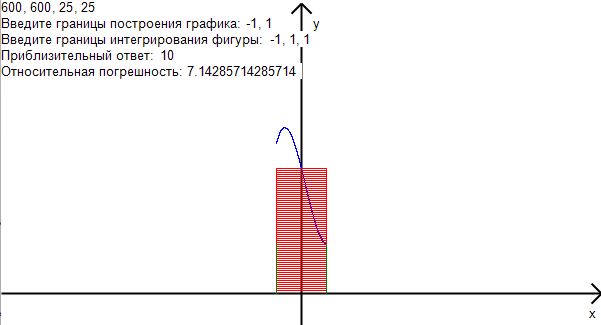
line(x0+round(b\*kx), y0, x0+round(b\*kx), y0-round(func(b)\*ky))

**end**.

1. **Результат выполнения программы**

****

****

****

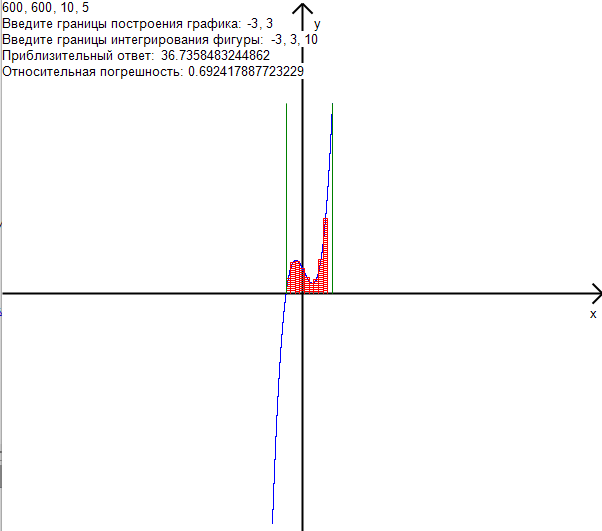
****

Рисунок 7 - Результаты выполнения программы**Вывод**

В процессе работы над домашней контрольной мы изучили принципы построения изображения и отображения функций в графическом виде.

Для создания алгоритмических схем был использован draw.io — бесплатный онлайн-инструмент для создания и совместного редактирования диаграмм и схем.

В ходе работы мы столкнулись с некоторыми трудностями. Например, были проблемы с построением графика, границы которого превышают границы экрана. Данная проблема была решена подбиранием меньшего коэффициента масштабирования.

В итоге, благодаря полученным знаниям и устранённым ошибкам, мы смогли успешно решить задачу, применив код с созданием изображения и построением графика функции и расширили наши знания о языке программирования Pascal.