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

Савватеев Аркадий Сергеевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

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

**1.Формулировка задания 1. Вариант 0**

**Постановка задачи**

Дополнить программу, реализованную в ходе предыдущей работы, режимом визуализации. Предусмотреть возможность вывода кривой, ограничивающей фигуру, на координатную плоскость. Реализовать следующие возможности и элементы: масштабирование графика, подписи на осях, вывод информации о задании. Реализовать не менее двух возможностей из представленных: штриховка вычисляемой площади, независимое масштабирование по осям, визуализация численного расчета интеграла.

**2.Описание алгоритма**

1. Определение функции: пишем функцию, которую хотим интегрировать, то есть данную нам в условии задачи

2. Функция с методом трапеций: реализуем основной алгоритм для вычисления интеграла с помощью нахождения высоты трапеции по формуле и цикла for для подсчета их количества.

**3.Схема алгоритма**

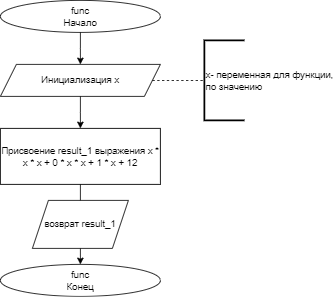
****

Рисунок 1.1 – схема алгоритма

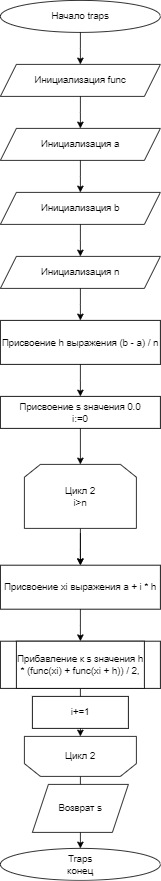


Рисунок 1.2 – схема алгоритма

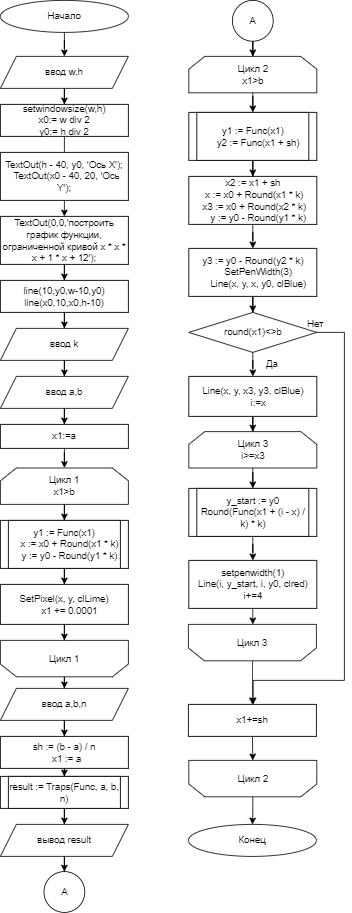


Рисунок 1.3 – схема алгоритма

**4.Код программы**

**Uses** GraphABC;

**var**

x, y, x0, y0, w, h, n, y3, x3: integer;

x1, y1, a, b, k, Sh, y2, x2,result: real;

**type**

FuncType = **function**(x: Real): Real;

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

**begin**

Func := x \* x \* x + 1 \* x + 12;

**end**;

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

**begin**

Primitive := (x \* x \* x \* x) / 4 + (x \* x) / 2 + 12 \* x;

**end**;

**function** ExactIntegral(a, b: Real): Real;

**begin**

ExactIntegral := Primitive(b) - Primitive(a);

**end**;

**function** Traps(Func: FuncType; a, b: Real; n: Integer): Real;

**var**

h, s, xi: Real;

i: Integer;

**begin**

h := (b - a) / n;

s := 0.0;

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

**begin**

xi := a + i \* h;

s := s + h \* (Func(xi) + Func(xi + h)) / 2;

**end**;

Traps := s;

**end**;

**begin**

writeln('Введите ширину: ');

readln(w);

writeln('Введите высоту: ');

readln(h);

SetWindowSize(w, h);

x0 := w **div** 2;

y0 := h **div** 2;

TextOut(h - 40, y0, 'Ось Х');

TextOut(x0 - 40, 20, 'Ось Y');

TextOut(0,0,'построить график функции, ограниченной кривой x \* x \* x + 1 \* x + 12');

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

Line(x0, 10, x0, h - 10);

writeln('Введите масштаб графика(лучше 25)');

readln(k);

writeln('Введите пределы участка для графика: ');

readln(a, b);

x1 := a;

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

**begin**

y1 := Func(x1);

x := x0 + Round(x1 \* k);

y := y0 - Round(y1 \* k);

SetPixel(x, y, clLime);

x1 += 0.0001;

**end**;

writeln('Введите данные о трапециях: ');

readln(a, b, n);

sh := (b - a) / n;

x1 := a;

result := Traps(Func, a, b, n);

Writeln('Приближенное значение интеграла: ', result);

// Рисуем трапеции

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

**begin**

y1 := Func(x1);

y2 := Func(x1 + sh);

x2 := x1 + sh;

x := x0 + Round(x1 \* k);

x3 := x0 + Round(x2 \* k);

y := y0 - Round(y1 \* k);

y3 := y0 - Round(y2 \* k);

SetPenWidth(3);

Line(x, y, x, y0, clBlue);

**if** round(x1) <> b **then**

**begin**

Line(x, y, x3, y3, clBlue);

// Штриховка трапеций

**var** i: integer;

i:=x;

**while** i<x3 **do**

**begin**

**var** y\_start := y0 - Round(Func(x1 + (i - x) / k) \* k);

setpenwidth(1);

Line(i, y\_start, i, y0, clred);

i+=4;

**end**;

**end**;

x1 += sh;

**end**;

**end**.

**5.Результат выполнения программы**

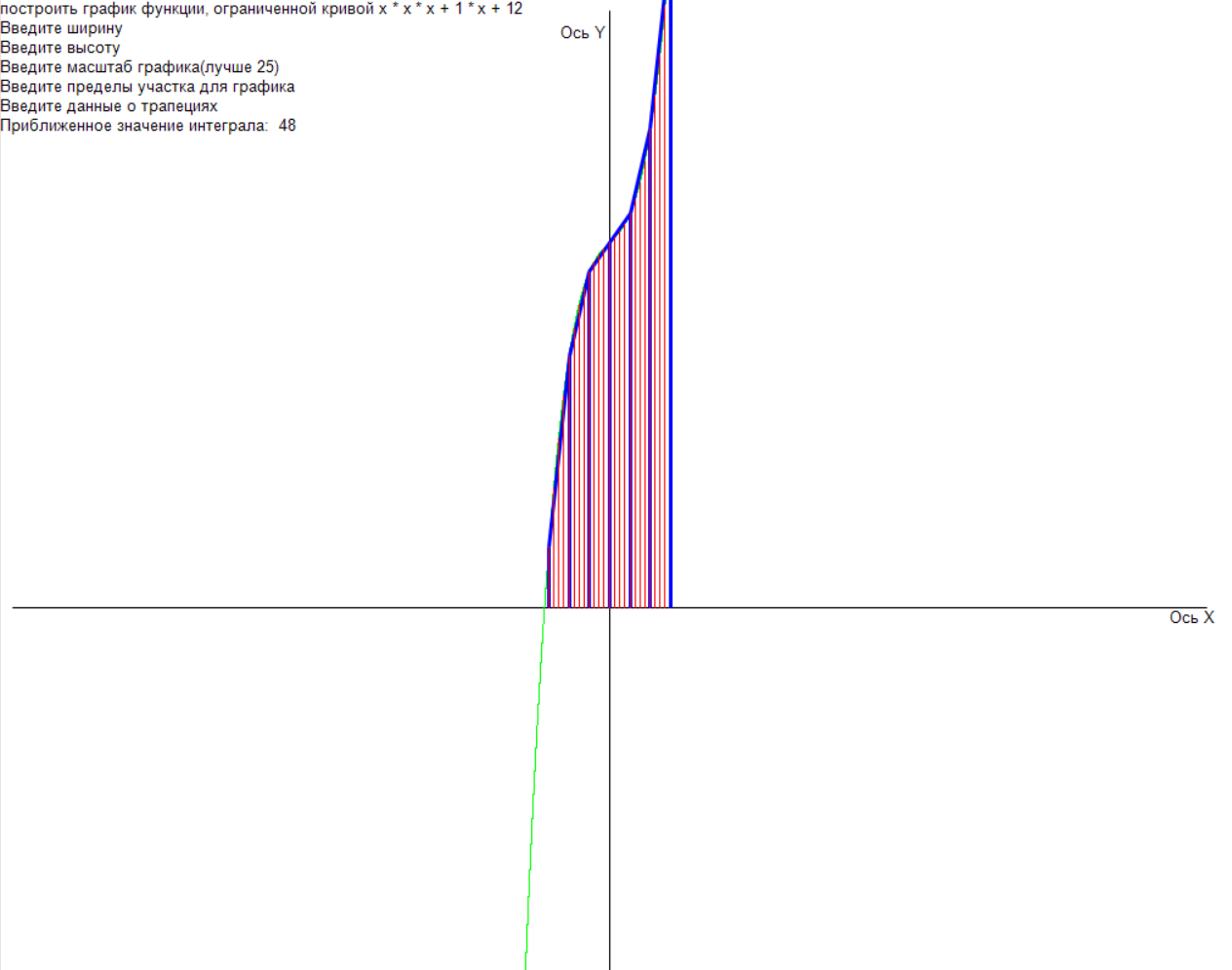
****

Рисунок 2 – примеры вводимых значений

1. **Вывод**

В ходе работы мы освоили принципы работы в графическом режиме; получили базовые навыки взаимодействия с графическими примитивами.

Для создания алгоритмических схем было изучено правильное построение письменного алгоритма для достижения верного решения задачи, а также и само построение схем в программе «draw.io».

Однако в ходе работы мы столкнулись и с проблемами. Например, при создании штриховки получались далеко не все способы, так как в нашем случае фигура - трапеция. Решением стало создание вертикальной штриховки, параллельной основным линиям, чтобы график был читаемым.

В конечном итоге при помощи полученных знаний и исправленных ошибок у нас получилось добиться поставленной цели – решить задачу при помощи кода с условиями и циклами, а также узнать о языке программирования Pascal новые знания.