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-202-52-00

Подпорин Антон Александрович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

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

**Формулировка задания (Вариант 17):**

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

**Описание алгоритма:**

1. Ввод границ кривой, заданной по функции с ДКР 3.
2. Ввод масштаб объема кривой, фигуры под ним.
3. Ввод количества промежутков, на которые будет делиться фигура.
4. Программа очищает окно.
5. Программа рисует оси координат в окне, размеры которого указаны в коде программы.
6. Задается шаг, который будет делить фигуру на трапеции, находя разность границ фигуры, деленную на количество промежутков.
7. Программа рисует кривую:
8. Начинает рисовать с начальной границы фигуры – а (присваивает x1 – а)
9. В цикле, который будет идти до второй границы фигуры, программа высчитывает y по заданной функции. Далее программа ищет точку на кривой, увеличивая x0 на округленное произведение x1 на масштаб объема кривой, уменьшая y0 на округленное произведение y1, вычисленное через функцию, в которой использовалось x1.
10. Программа дорисовывает фигуру до трапеции, соединяя точки на кривой между промежутками фигуры.
11. Увеличение шага на 1.
12. Программа позволяет изменять масштаб графика с помощью созданной ранее функции mash, которая считывает нажатие соответствующих клавиш и изменяет масштаб.

**Схема алгоритма:**

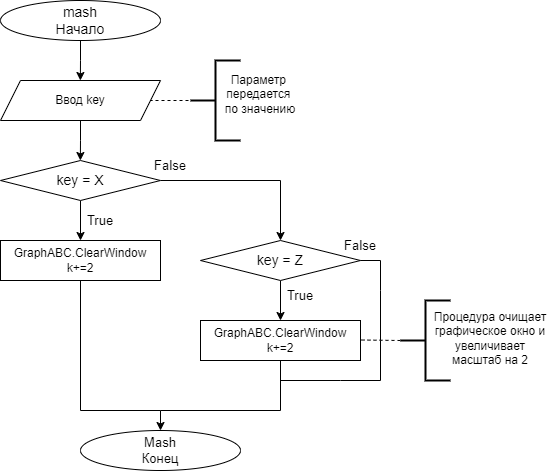


Рисунок 1 – схема алгоритма для процедуры mash

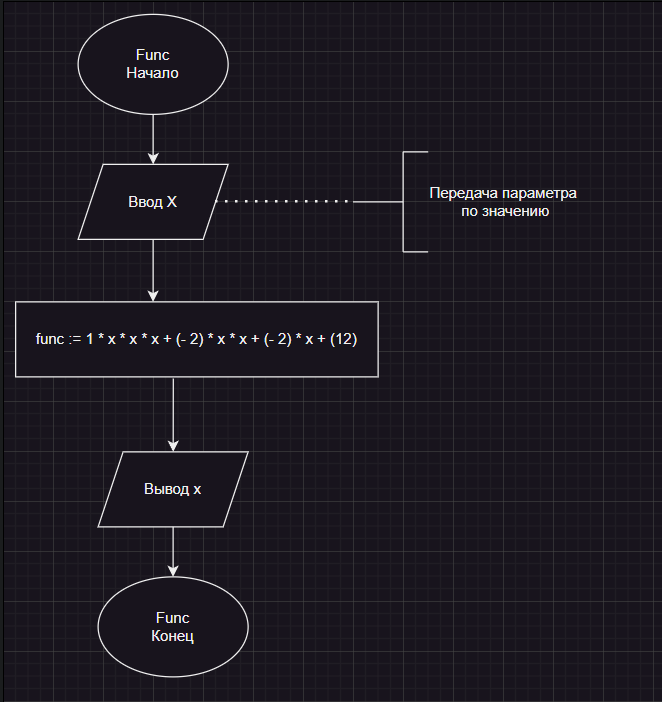


Рисунок 2 – схема алгоритма для функции func

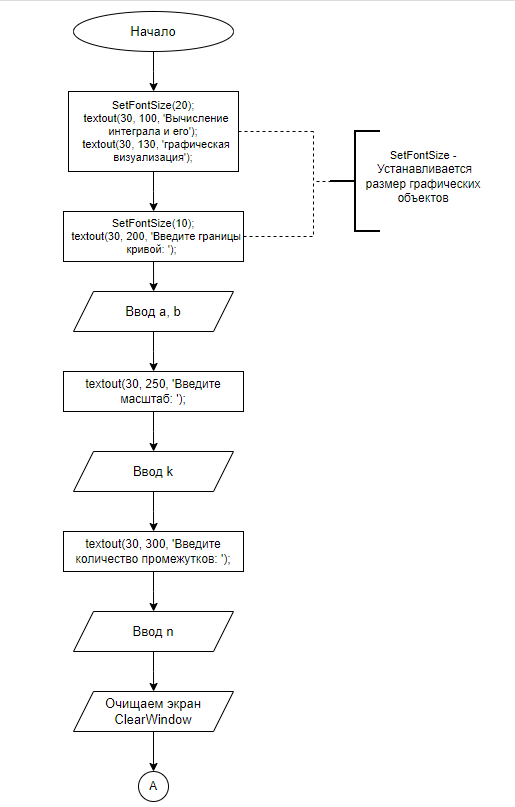


Рисунок 3 – схема алгоритма основной программы (1)

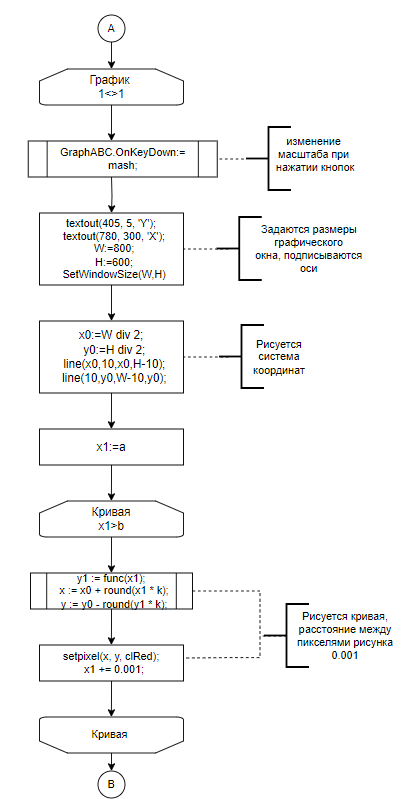


Рисунок 4 – Схема алгоритма основной программы (2)

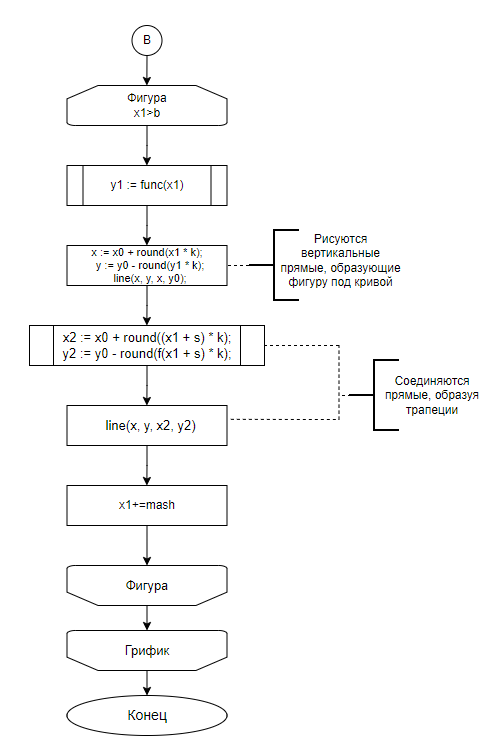
****

Рисунок 5 – Схема алгоритма основной программы (3)

**Код программы:**

**uses** GraphABC;

**var**

W, H, x, y, x0, y0, n, x2, y2:integer;

x1, y1, a, b, k, sh:real;

**procedure** mash(key: integer);

**begin**

**case** key **of**

GraphABC.VK\_X:

**begin**

GraphABC.ClearWindow;

k += 2;

**end**;

GraphABC.VK\_Z:

**begin**

GraphABC.ClearWindow;

k -= 2;

**end**;

**end**;

**end**;

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

**begin**

func:= x \* x \* x - 2 \* x \* x - 2 \* x + 12

**end**;

**begin**

SetFontSize(20);

textout(30, 100, 'Вычисление интеграла и его');

textout(30, 130, 'графическая визуализация');

SetFontSize(10);

textout(30, 200, 'Введите границы кривой: ');

readln(a, b);

textout(30, 250, 'Введите масштаб: ');

readln(k);

textout(30, 300, 'Введите количество промежутков: ');

readln(n);

clearwindow;

**while** 1 = 1 **do**

**begin**

GraphABC.OnKeyDown:= mash;

textout(405, 5, 'Y');

textout(780, 300, 'X');

W:=800;

H:=600;

SetWindowSize(W,H);

x0:=W **div** 2;

y0:=H **div** 2;

line(x0,10,x0,H-10);

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

x1 := a;

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

y1 := func(x1);

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

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

setpixel(x, y, clRed);

x1 += 0.001;

**end**;

sh := abs((b - a) / n);

x1 := a;

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

y1 := func(x1);

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

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

line(x, y, x, y0);

x2 := x0 + round((x1 + sh) \* k);

y2 := y0 - round(func(x1 + sh) \* k);

line(x, y, x2, y2);

x1 += sh;

**end**;

**end**;

**end**.

**Результаты выполнения программы:**

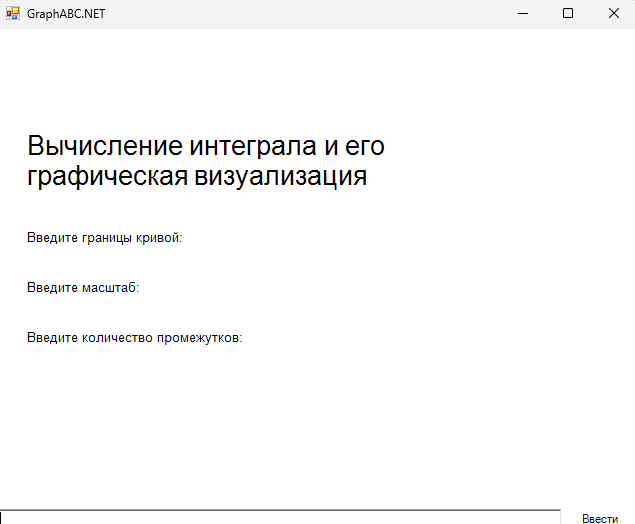


Рисунок 6 – Результат выполнения программы (1)

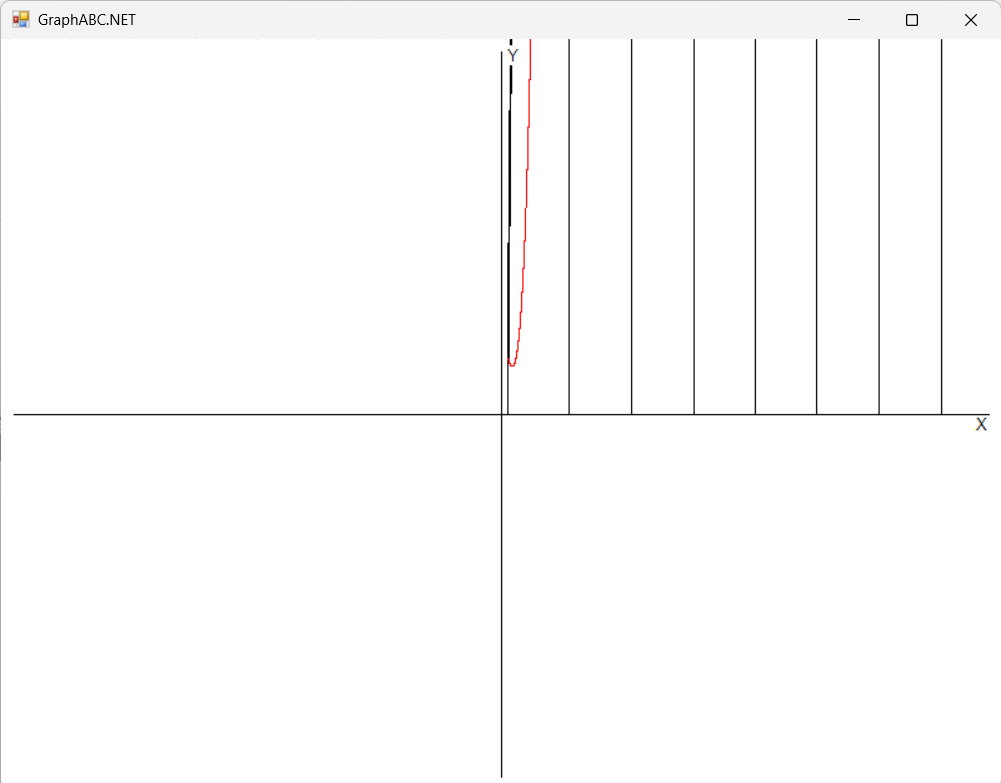
****

Рисунок 7 – Результат выполнения программы (2)

**Вывод:**

В ходе работы автор успешно разработал программу для визуализации математических функций. Программа строит график функции, оси координат, строит кривую, и позволяет масштабировать график и область просмотра.

Перед началом работы потребовалось изучение графических возможностей Паскаля и выбор соответствующих алгоритмов для построения и масштабирования графика.

После тестирования и отладки, программа эффективно визуализирует и анализирует функции, демонстрируя навыки программирования на Паскале и применение полученных знаний в области математической визуализации. Также автор работы получил опыт математического программирования на языке Pascal.