**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-203-52-00

Долгих Дмитрий Константинович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

**Цель работы**

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

**Формулировка задания**

Вариант 9

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 расчёта интеграла.

**Описание алгоритма**

Для решения данной задачи используется модуль *CRT* для вывода консоли на экран пользователя, в которой происходит взаимодействие с самим пользователем. Case-меню имеет выбор из двух пунктов, выбор между которыми осуществлён через оператор выбора *case…of*. Для перемещения по меню используется процедура *GoToXY*, которая переносит курсор в заданное место на экране. Для считывания клавиш с клавиатуры используется функция *ReadKey*. Для визуального оформления case-меню была использована процедура *TextColor*, которая меняет цвет текста.

Данный алгоритм вычисляет определённый интеграл методом правых прямоугольников. В алгоритме используются функции и процедуры для вычисления производной, первообразной и определённого интеграла.

При выборе пункта «Начать интегрирование» пользователю предлагается ввести пределы интегрирования и количество промежутков для изменения точности вычисления определённого интеграла.

Для визуализации графика функции используется дополнительный модуль *GraphABC*. Переключение между консолью и графическим окном используются процедуры *SetConsoleIO* и *SetGraphABCIO*. Для работы с графическими примитивами используются процедуры *SetPixel* (закрашивает пиксель), *Line* (рисует линию), *Rectangle* (рисует прямоугольник), *TextOut* (выводит текст в определенном месте графического окна), *SetBrushColor* (устанавливает цвет кисти), *SetBrushStyle* (устанавливает стиль кисти), *SetBrushHatch* (устанавливает стиль штриховки), *SetFontSize* (устанавливает размер выводимого текста), *SetFontColor* (устанавливает цвет выводимого текста), *SetWindowSize* (устанавливает размеры графического окна).

Чтобы изменить масштаб графика, пользователь вводит параметры графика, границы осей и единичный отрезок по х и по у.

Для отрисовки графика используется цикл, в котором рассчитывается значение функции, данной в предыдущей лабораторной работе, и закрашивает пиксели с координатами х и у.

Если пользователь ввел значения a и b, то на график добавляются прямые x = a и x = b, ограничивающие кривую.

С помощью цикла в промежутке от a до b рисуются прямоугольники по заданному пользователем количеству промежутков.

В результате выполнения программы, при выборе пункта «Смотреть график», в консоли пользователь вводит параметры для масштабирования графика, а в графическом окне выводится криволинейная трапеция в системе координат и визуализация расчета площади фигуры по методу правых прямоугольников, а также информация о результате интегрирования.

**Схема алгоритма с комментариями**

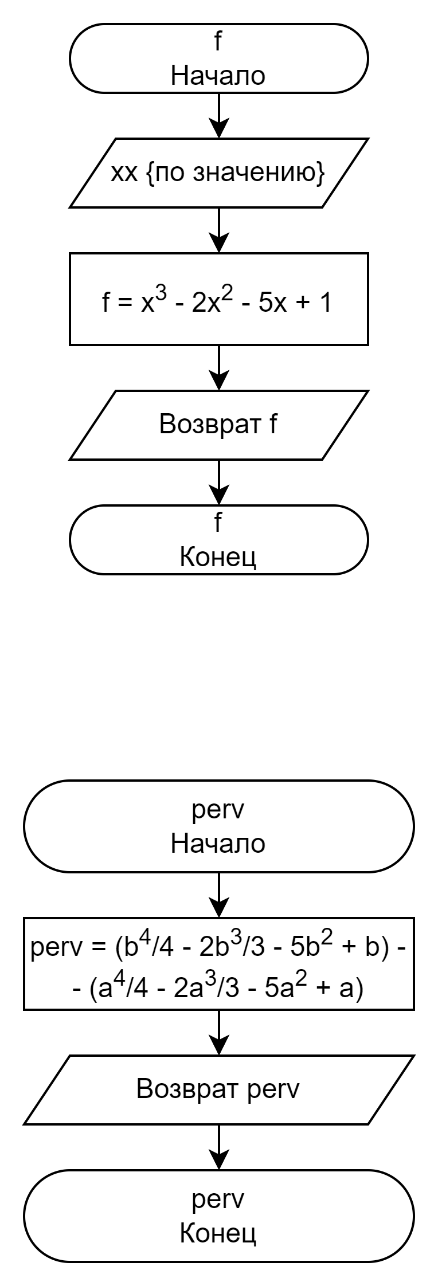


Рис. 1 «Схема функции для вычисления площади криволинейной трапеции по правилу правых прямоугольников и по формуле Ньютона-Лейбница»



Рис. 2 «Схема процедуры вывода меню на экран»



Рис. 3 «Схема процедуры для вывода информации о задании в графическом окне»

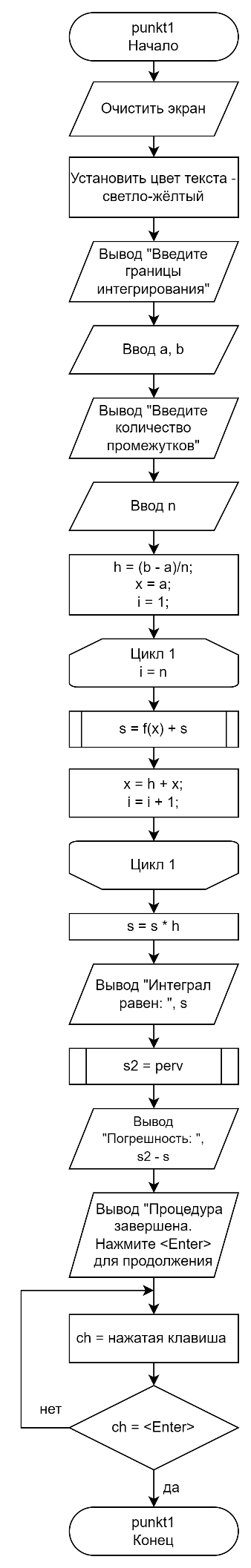


Рис. 4 «Схема вычисления определённого интеграла и погрешности»

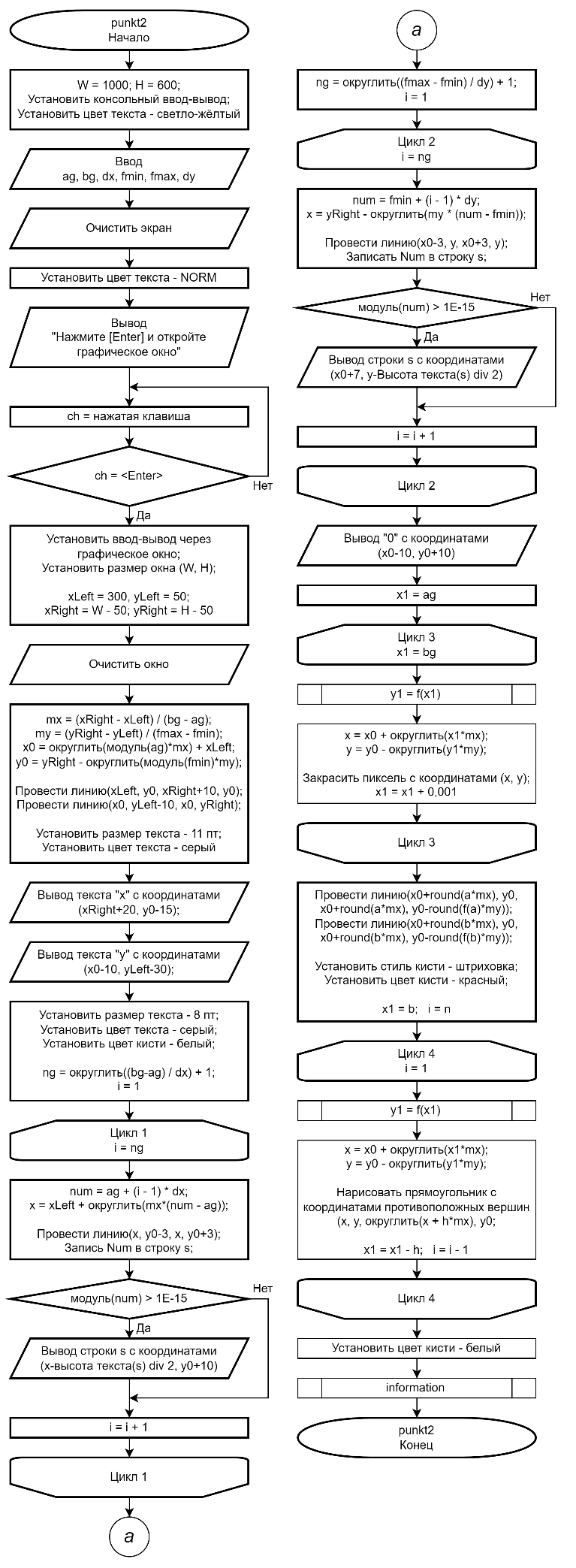


Рис. 5 «Схема процедуры для отрисовки графика»

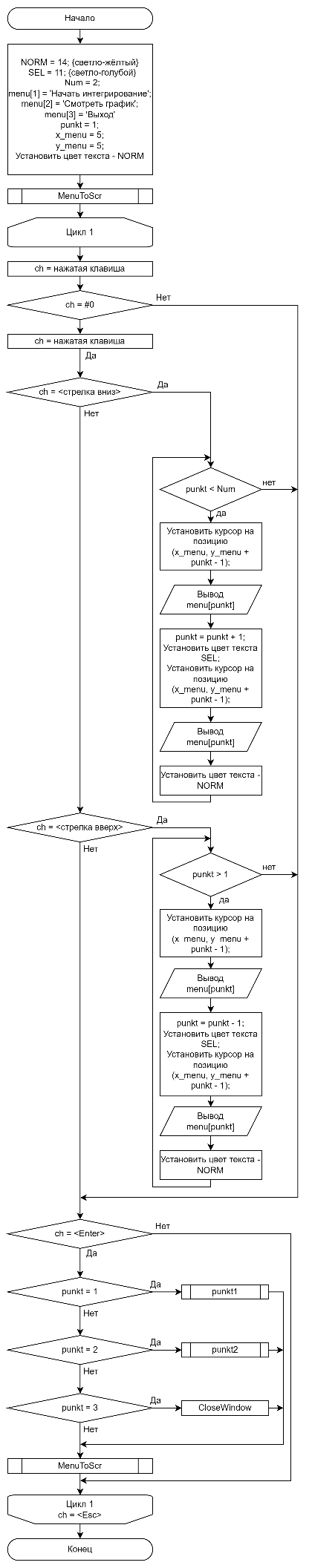


Рис. 6 «Схема алгоритма главной программы»

**Код программы**

**program** dkr4\_casemenu;

**uses** Crt, GraphABC;

**const**

NORM = 14; { цвет невыделеного пункта }

SEL = 11; { цвет выделенного пункта }

Num = 3;

**var**

menu: **array**[1..Num] **of** string[24];{ названия пунктов меню }

punkt: integer;

ch: char;

x\_menu, y\_menu: byte;

a, b, h, x, s, dx, dy: real;

n: integer;

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

**begin**

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

**end**;

**function** perv(xx: real): real;

**begin**

perv:= ((b\*\*4)/4 - (2\*b\*\*3)/3 - (5\*b\*\*2)/2 + b) - ((a\*\*4)/4 - (2\*a\*\*3)/3 - (5\*a\*\*2)/2 + a);

**end**;

**procedure** pravye;

**begin**

SetConsoleIO;

ClrScr;

writeln('Введите границы интегрирования: ');

readln(a, b);

writeln('Введите количество промежутков: ');

readln(n);

h:= (b-a)/n;

x:= a;

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

**begin**

s+= f(x);

x+= h;

**end**;

s\*= h;

writeln('Интеграл равен: ', s:0:3);

writeln('Погрешность: ', perv(x)-s);

**end**;

**procedure** punkt1;

**begin**

ClrScr;

pravye;

writeln;

writeln('Процедура завершена. Нажмите <Enter> для продолжения.');

**repeat**

ch := readkey;

**until** ch = #13;

**end**;

**procedure** information;

**begin**

setfontsize(11);

setfontcolor(clblack);

**if not** ((a = 0) **and** (b = 0)) **then**

**begin**

writeln('Нижний предел: ', a);

writeln('Верхний предел: ', b);

writeln('Площадь фигуры: ', s:0:3);

writeln('Погрешность: ', perv(x)-s);

**end**

**else** writeln('Вы не ввели данные для интегрирования');

**end**;

**procedure** punkt2;

**const**

W = 1000; H1 = 600;//Размеры графического окна

**var**

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

ag, bg, fmin, fmax, x1, y1, mx, my, num: real;

i: byte;

s: string;

**begin**

SetConsoleIO;

textcolor(14);

clrscr;

Writeln('Введите нижнюю границу системы координат по Х: ');

read(ag);

Writeln('Введите верхнюю границу системы координат по Х: ');

read(bg);

Writeln('Введите единичный отрезок по Х: ');

read(dx);

Writeln('Введите нижнюю границу системы координат по Y: ');

read(fmin);

Writeln('Введите верхнюю границу системы координат по Y: ');

read(fmax);

Writeln('Введите единичный отрезок по Y: ');

read(dy);

writeln;

clrscr;

textcolor(norm);

Writeln('Нажмите [Enter] и откройте графическое окно');

**repeat**

ch := readkey;

**until** ch = #13;

SetGraphabcIO;

SetWindowSize(W, H1); //Устанавливаем размеры графического окна

xLeft := 300;

yLeft := 50;

xRight := W - 50;

yRight := H1 - 50;

clearwindow;

mx := (xRight - xLeft) / (bg - ag); //масштаб по Х

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

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

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

line(xLeft, y0, xRight + 10, y0); //ось ОХ

line(x0, yLeft - 10, x0, yRight); //ось ОY

SetFontSize(11);

SetFontColor(clSlateGray);

TextOut(xRight + 20, y0 - 15, 'х');

TextOut(x0 - 10, yLeft - 30, 'у');

SetFontSize(8);

SetFontColor(clgray);

setbrushcolor(clwhite);

ng := round((bg - ag) / dx) + 1; //количество засечек по ОХ

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

**begin**

num := ag + (i - 1) \* dx; //Координата на оси ОХ

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

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

str(Num:0:1, s);

**if** abs(num) > 1E-15 **then** //Исключаем 0 на оси OX

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

**end**;

ng := round((fmax - fmin) / dy) + 1; //количество засечек по ОY

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

**begin**

num := fMin + (i - 1) \* dy; //Координата на оси ОY

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

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

str(num:0:0, s);

**if** abs(num) > 1E-15 **then** //Исключаем 0 на оси OY

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

**end**;

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

x1 := ag;

**while** x1 <= bg **do**

**begin**

y1 := f(x1);

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

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

SetPixel(x, y, clBlack);

x1 := x1 + 0.001

**end**;

line(x0 + round(a\*mx), y0, x0 + round(a\*mx), y0 - round(f(a)\*my), clBlack); // х = а

line(x0 + round(b\*mx), y0, x0 + round(b\*mx), y0 - round(f(b)\*my), clBlack); // х = b

setbrushstyle(bsHatch);

setbrushhatch(bhPercent10);

setbrushcolor(clRed);

x1:= b;

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

**begin**

y1:= f(x1);

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

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

rectangle(x, y, round(x - h\*mx), y0);

x1:= x1 - h;

**end**;

setbrushcolor(clWhite);

information;

**end**;

**procedure** MenuToScr;{ вывод меню на экран }

**var**

i: integer;

**begin**

SetConsoleIO;

ClrScr;

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

**begin**

GoToXY(x\_menu, y\_menu + i - 1);

write(menu[i]);

**end**;

TextColor(SEL);

GoToXY(x\_menu, y\_menu + punkt - 1);

write(menu[punkt]);{ выделим строку меню }

TextColor(NORM);

**end**;

**begin**

SetConsoleIO;

ClrScr;

menu[1] := ' Начать интегрирование ';

menu[2] := ' Смотреть график ';

menu[3] := ' Выход ';

punkt := 1; x\_menu := 5; y\_menu := 5;

TextColor(NORM);

MenuToScr;

**repeat**

ch := ReadKey;

**if** ch = #0 **then begin**

ch := ReadKey;

**case** ch **of**

#40:{ стрелка вниз }

**if** punkt < Num **then begin**

GoToXY(x\_menu, y\_menu + punkt - 1); write(menu[punkt]);

punkt := punkt + 1;

TextColor(SEL);

GoToXY(x\_menu, y\_menu + punkt - 1); write(menu[punkt]);

TextColor(NORM);

**end**;

#38:{ стрелка вверх }

**if** punkt > 1 **then begin**

GoToXY(x\_menu, y\_menu + punkt - 1); write(menu[punkt]);

punkt := punkt - 1;

TextColor(SEL);

GoToXY(x\_menu, y\_menu + punkt - 1); write(menu[punkt]);

TextColor(NORM);

**end**;

**end**;

**end**

**else**

**if** ch = #13 **then begin**{ нажата клавиша <Enter> }

**case** punkt **of**

1: punkt1;

2: punkt2;

3: CloseWindow;{ выход }

**end**;

MenuToScr;

**end**;

**until** ch = #27;{ 27 - код <Esc> }

**end**.

**Результат выполнения программы**

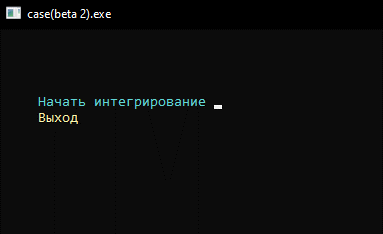


Рис. 7 «Case-меню»

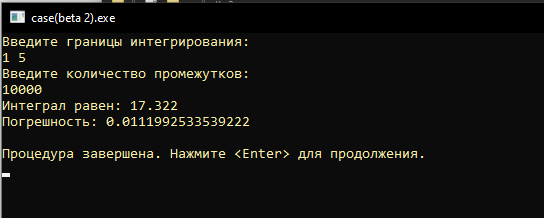


Рис. 8 «Результат интегрирования»

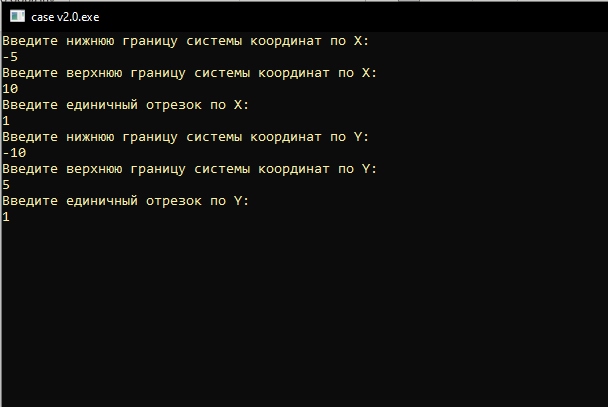


Рис. 9 «Ввод границ системы координат»

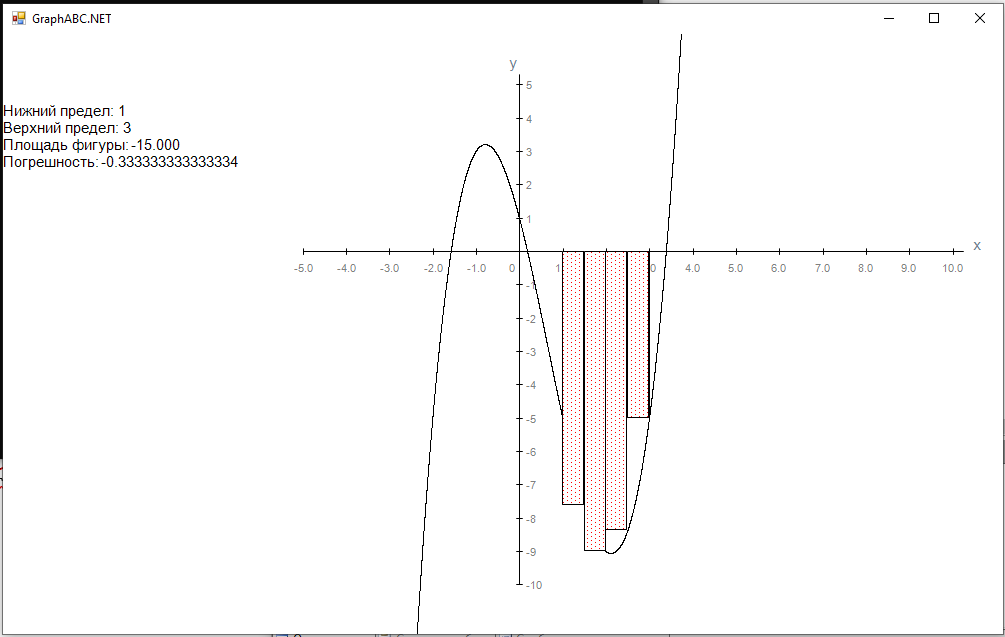


Рис. 10 «График функции»

**Вывод**

Данная работа прошла успешно, в ней мы освоили принципы работы в графическом режиме, получили базовые навыки взаимодействия с графическими примитивами.

Во время выполнения работы возникли небольшие трудности с написанием кода и построением схемы алгоритма. Некоторые из таких проблем:

При выполнении кода всё, что должно было выводиться в консоли, выводилось в графическое окно. В итоге, чтобы решить данную проблему были использованы процедуры *SetConsoleIO* и *SetGraphABCIO*.

Процедура выхода в третьем пункте по какой-то причине перестала работать через присвоение к *ch* нажатие клавиши <Enter>, поэтому пришлось заменить её на процедуру *CloseWindow*.

В итоге, работа улучшила знания в языке программирования Pascal, научила работать с графическим модулем *GraphABC*, некоторыми графическими примитивами. Основная цель работы была достигнута, все задачи выполнены, все ошибки и недочёты исправлены.