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

Носкова Анастасия Владимировна

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

1. **Цель работы**

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

1. **Формулировка задания**

Вариант: 16

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. Создание типа данных bam для функции от одной переменной типа Real;
   2. Определение функции f(x): функция, которую мы хотим проинтегрировать;
   3. Определение процедуры по a, b – границам интегрирования, n – количеству делений, функцию func типа bam (в данном случае функцию f(x)), и переменные s и pogr типа Real по ссылке;
   4. Внутри процедуры находится h – длинна каждого деления, с помощью цикла for находится сумма значений f(x);
   5. После этого рассчитывается площадь фигуры, ограниченной кривой, путем умножения суммы значений функции на шаг h;
   6. Вычисление погрешности с помощью формулы Ньютона-Лейбница;
   7. Процедура "information" устанавливает шрифт и цвет текста, а затем выводит информацию о нижнем и верхнем пределах, если a и b не равны 0.
   8. Процедура построения функции:
   * Задаются константы W (ширина окна), H1 (высота окна).
   * Задаются переменные и считываются границы системы координат и единичный отрезок по X и Y.
   * Очищается экран и ожидается ввод Enter для открытия графического окна.
   * Устанавливаются размеры графического окна, настраиваются оси координат, масштабирование, рисуются засечки, подписываются оси и пр.
   * Вычисляются значения функции и рисуются точки графика.
   * На графике отмечаются основные точки и значения функции, ограниченные по аргументу.
   * Рисуются прямоугольники (под графиком) - для метода прямоугольников.
   1. В основной программе используется цикл repeat-until для выбора действия пользователем. Пользователю предоставляется выбор вычисления площади фигуры (1), вывода графика на экран (2) и выхода из программы (3);
   2. При выборе 1, пользователь вводит границы интегрирования и шаг, после чего на экран выводится площадь и погрешность;
   3. При выборе 2 пользователь вводит необходимый масштаб и переключается из case menu в graphabc для просмотра результата.
   4. При выборе 3, пользователь выходит из программы.
6. **Схема алгоритма**

**Изображение выглядит как текст, снимок экрана, Шрифт, черно-белый

Автоматически созданное описание**

Рисунок 1 – схема алгоритма функции f.

Изображение выглядит как текст, снимок экрана, Шрифт, графический дизайн

Автоматически созданное описание

Рисунок 2 – схема алгоритма процедуры lev.

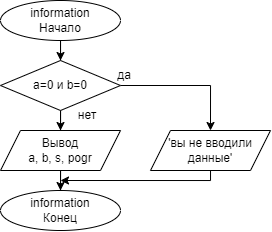


Рисунок 3 – схема алгоритма процедуры information.

Изображение выглядит как текст, снимок экрана, черно-белый, дизайн

Автоматически созданное описание

Рисунок 4 – схема алгоритма процедуры graph.

Изображение выглядит как снимок экрана, текст, черно-белый, Графика

Автоматически созданное описание

Рисунок 5 – схема алгоритма основной программы.

1. **Код программы**

**uses** Crt, GraphABC;

**type**

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

**var**

a, b, dx, dy, h: Real;

n: Integer;

s, pogr: Real;

ch: char;

x, y: integer;

color : Byte;

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

**begin**

f:=2\*(x\*\*3)+(-2)\*(x\*\*2)+(-3)\*x+14;

**end**;

**procedure** lev(a, b: Real; n: Integer; func: bam; **var** s, pogr:real);

**var**

x: Real;

i: Integer;

fa, fb: real; //Значения первообразной

**begin**

h:= (b-a)/n;

x:= a;

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

s:=s+f(x);

x:=x+h;

**end**;

s:=s\*h;

fa:=1/2\*a\*\*4-2/3\*a\*\*3-3/2\*a\*\*2+14\*a;

fb:=1/2\*b\*\*4-2/3\*b\*\*3-3/2\*b\*\*2+14\*b;

pogr:=(fb-fa) - s;

**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 заштрихованной фигуры: ', s:0:2);

writeln('Погрешность: ', pogr);

**end**

**else** writeln('Вы не вводили данные');

**end**;

**procedure** graph;

**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(11);

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(yellow);

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'); //Подписываем ось OX

TextOut(x0 - 10, yLeft - 30, 'у'); //Подписываем ось OY

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); //рисуем засечки на оси OX

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); //рисуем засечки на оси Oy

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); //Вычисляем значение функции

x := x0 + round(x1 \* mx); //Координата Х в графическом окне

y := y0 - round(y1 \* my); //Координата Y в графическом окне

SetPixel(x, y, clred);

x1 := x1 + 0.001 //Увеличиваем абсциссу

**end**;

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

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

setbrushstyle(bsHatch);

setbrushhatch(bhLargeConfetti);

setbrushcolor(clOlive);

x1:=a;

**for** i:=0 **to** n-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**;

**begin**

**repeat**

SetConsoleIO;

ClrScr;

textcolor(Yellow);

writeln('1. Вычисление площади фигуры, ограниченной кривой');

writeln('2. График'); //добавила пункт для графика

writeln('3. Выход');

write('Выберите действие: ');

ch := ReadKey;

**case** ch **of**

'1':

**begin**

ClrScr;

Textcolor(Yellow);

writeln('Введите границы интегрирования: ');

readln(a, b);

Textcolor(Yellow);

writeln('Введите количество делений: ');

readln(n);

lev(a, b, n, f, s, pogr);

Textcolor(LightGreen);

writeln('Площадь фигуры: ', s);

Textcolor(LightGreen);

writeln('Погрешность: ', pogr);

readln;

**end**;

'2': graph;

'3': halt;

**end**;

**until** ch = '3';

**end**.

**Результат выполнения программы:**

**Изображение выглядит как текст, снимок экрана, График, диаграмма

Автоматически созданное описание**

Рисунок 6 – вывод данных 1.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 7 – вывод данных 2.

1. **Вывод**

При выполнении данной работы возникли трудности с реализацией совместимости case menu и GraphABС, но путем проб это получилось исправить.

Одной из самых главных проблем было масштабирование графика, а после независимое масштабирование осей x и y. Для решения понадобилось найти информацию, и реализовать ее на практике с помощью ввода пользователем границ оси x, y и шага у каждой оси.

В остальном программа полностью соответствует заданию, было настроено независимое масштабировании, подписаны оси, выполняется штриховка искомой площади и визуализация численного расчета интеграла.