Лабораторная работа №6

по вычислительной практике

Тема:

«Графические возможности языка Pascal»

Вариант 15

Выполнил: студент гр. ПМ-12

Орловский A.C.

Проверила: Кузьменкова Е.Ю.

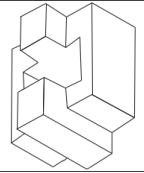
Гомель, 2021

**Постановка задачи**

Построить график функции y = f(x) согласно варианту на отрезке [a, b] (a, b – задаются произвольно).

y = 

Написать программу для рисования заданного изображения, согласно варианту, с заливкой каждой области разным цветом.



**Изученные теоретические вопросы**

1. Порядок перехода в графический режим.

**Uses** Graph;

**Var**

Driver, Mode: Integer;

**Begin**

Driver:=Detect;

{режим автоопределения типа графического драйвера}

InitGraph(Driver,Mode,’’);

{графический режим}

**Repeat** **Until** KeyPressed; {задержка экрана}

CloseGraph; {закрыли графический режим}

**End**.

1. Процедуры рисования линий, окружностей, прямоугольников.

Line (x1,y1,x2,y2) – процедура вычерчивает линию с координатами начала (x1,y1) и конца (x2,y2). Линия вычерчивается текущим стилем и текущим цветом.

LineTo (x,y) – процедура вычерчивает линию от текущего положения указателя до точки с заданными координатами (x,y). Линия вычерчивается текущим стилем и текущим цветом.

LineRel (dx,dy) – процедура вычерчивает линию от текущего положения указателя до положения, заданного приращениями координат (dx,dy). Линия вычерчивается текущим стилем и текущим цветом.

Rectangle (x1,y1,x2,y2) – процедура вычерчивает прямоугольник с заданными координатами углов. Здесь (x1,y1) – координаты левого верхнего, (x2,y2) – правого нижнего углов прямоугольника. Прямоугольник вычерчивается с использованием текущего цвета и стиля линий.

Circle (x,y,*радиус*) – процедура вычерчивает окружность с центром в точке (x,y) заданного радиуса.

1. Вывод текстовой информации в графическом режиме.

OutText (*текст*) – процедура выводит текстовую строку, начиная с текущего положения указателя.

OutTextXY (x,y,*текст*) – процедура выводит строку, начиная с позиции (x,y).

SetTextStyle(*шрифт*,*направление*,*размер*) – процедура устанавливает стиль текстового вывода на графический экран. Здесь *шрифт –* код (номер) шрифта; *направление* –код направления; *размер –* код размера шрифта.

*Код* *шрифта* задается одной из следующих предварительно определенных констант:

**Const**

DefaultFont=0; {Точечный шрифт 8x8}

TriplexFont=1; {Утроенный шрифт TRIP.CHP}

SmallFont=2; {Уменьшенный шрифт LITT.CHR}

SansSerifFont=3; {Прямой шрифт SANS.CHR)

GothicFont=4; {Готический шрифт GOTH.CHR}

1. Процедуры установки окна, цвета фона и символов.

SetColor (*цвет*) – процедура устанавливает цвет выводимых линий и символов. Цвет определяется одной из следующих констант, находящихся в модуле Graph:

SetColor (*цвет*) – процедура устанавливает цвет выводимых линий и символов. Цвет определяется одной из следующих констант, находящихся в модуле Graph:

**Const**

Black=0; {черный}

Blue=1; {синий}

Green=2; {зеленый}

Cyan=3; {голубой}

Red=4; {красный}

Magenta=5; {сиреневый}

Brown=6; {коричневый}

LightGray=7; {светло-серый}

DarkGray=8; {темно-серый}

LightBlue=9; {голубой}

LightGreen=10; {светло-зеленый}

LightCyan=11; {светло-голубой}

LightRed=12; {светло-красный}

LightMagenta=13; {светло-сиреневый}

Yellow=14; {желтый}

White=15; {белый}

GetColor – функция возвращает значение типа Word, содержащее код текущего цвета.

GetMaxColor – функция возвращает значение типа Word, содержащее максимальный доступный код цвета, который можно использовать для обращения к SetColor.

SetBkColor (*цвет*) – процедура устанавливает цвет фона.

GetBkColor – функция возвращает значение типа Word, содержащее текущий цвет фона.

SetPalette (n, Color) – процедура заменяет один из цветов палитры на новый цвет. Здесь n – номер цвета в палитре, Color – номер вновь устанавливаемого цвета.

GetPalette (PaletteInfo) – процедура возвращает размер и цвета текущей палитры. Здесь PaletteInfo – переменная типа PaletteType, возвращающая размер и цвета палитры. В модуле Graph определена константа

**Const**

1. Процедуры заполнения произвольной области заданным цветом.

FloodFill(x,y,цвет\_границы) – заполняет произвольную замкнутую фигуру, используя текущий стиль заполнения до границы заданного цвета, (x,y) – произвольная точка внутри фигуры

**Программа и результаты**

**program** uvp6\_1;

{y:=sin(x)+sqr(x)+3}

**uses** Graph;

**const**

a=-7; b=7;

wm=50; wn=40;

**var**

driver,mode: Integer;

i,wx,wy: Integer;

x,y: Real;

**begin**

driver:=Detect;

initgraph(driver,mode,'');

wx:=getmaxX **div** 2;

wy:=getmaxY **div** 2;

setcolor(red);

setlinestyle(1,0,3);

line(0,wy,getmaxX,wy);

line(wx,0,wx,getmaxY);

setcolor(white);

outtextXY(wx+7,15,'Y');

outtextXY(getmaxX-25,wy+6,'X');

setcolor(green);

setlinestyle(0,0,1);

i:=1;

**while** (wx-i\*wm>0) **or**

(wx+i\*wm <getmaxX) **do**

**begin**

line(wx-i\*wm,0,

wx-i\*wm,getmaxY);

line(wx+i\*wm,0,wx+i\*wm,getmaxY);

inc(i);

**end**;

i:=1;

**while** (wy-i\*wn>0) **or**

(wy+i\*wn <getmaxX) **do**

**begin**

line(0,wy-i\*wn,getmaxX,

wy-i\*wn);

line(0,wy+i\*wn,getmaxX,

wy+i\*wn);

inc(i);

**end**;

i:=a\*wm;

**while** i<=b\*wm **do**

**begin**

x:=i/wm;

y:=sqrt(x+3\*abs(x));

putpixel(round(wx+wm\*x),

round(wy-wn\*y),white);

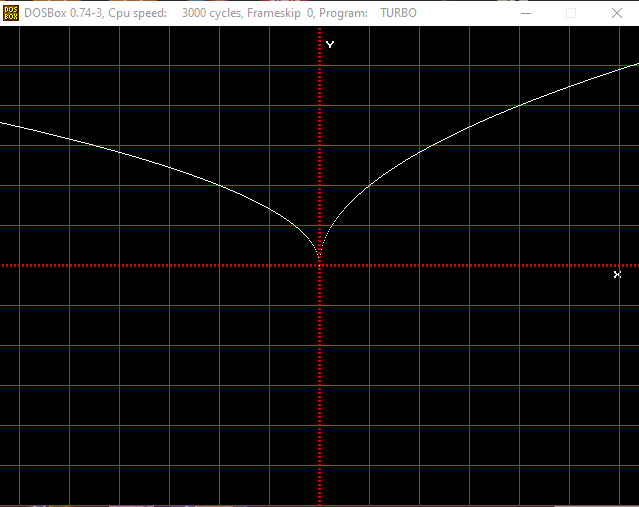
i:=i+1;

**end**;

readln;

closegraph;

**end**.



**program** uvp6\_2;

**uses** graph;

**var**

driver, mode: Integer;

**begin**

driver:=Detect;

initgraph(driver,mode,'');

setcolor(white);

setlinestyle(0,0,1);

line(30,60,60,30);

line(60,30,60,80);

line(60,80,30,100);

line(30,100,30,60);

line(60,30,90,50);

line(90,50,90,90);

line(90,90,60,80);

line(90,50,130,20);

line(130,20,180,60);

line(180,60,180,170);

line(180,170,130,150);

line(130,150,130,20);

line(130,150,100,170);

line(100,170,100,140);

line(100,140,110,135);

line(110,135,110,70);

line(110,70,90,90);

line(90,90,90,50);

line(100,140,50,130);

line(50,130,40,125);

line(40,125,55,115);

line(55,115,30,100);

line(50,130,30,150);

line(30,150,30,190);

line(30,190,50,170);

line(50,170,50,130);

line(50,170,150,220);

line(150,220,130,240);

line(130,240,30,190);

line(150,220,150,190);

line(150,190,180,170);

line(150,190,100,170);

line(90,220,60,250);

line(60,250,10,220);

line(10,220,35,195);

line(10,220,10,100);

line(10,100,30,80);

setfillstyle(1,blue);

floodfill(45,200,15);

setfillstyle(1,green);

floodfill(50,60,15);

setfillstyle(1,red);

floodfill(80,60,15);

setfillstyle(1,brown);

floodfill(90,100,15);

setfillstyle(1,white);

floodfill(80,150,15);

setfillstyle(1,green);

floodfill(160,80,15);

setfillstyle(1,red);

floodfill(80,200,15);

readln;

closegraph;

**end**.

