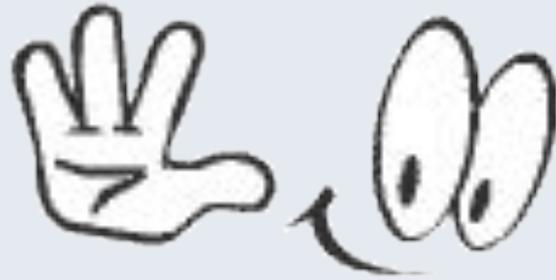


Программирование – вторая грамотность.

А.П.Ершов

Графические ВОЗМОЖНОСТИ Pascal



Примеры графических программ

```

File Edit Search Run Compile Debug Tools Options Window Help
\DOCUME~2\STUDENTS\MI-02\PASCAL\ПРОГРА~1\LAB10\LAB102.PAS 2
\DOCUME~2\STUDENTS\MI-02\PASCAL\ПРОГРА~1\LAB10\LAB101.PAS:2 3=[↑]
program lab101;
uses crt,graph;
var dr,md:integer;
    x,y,dx,dy:real;
    x0,y0,d,h:integer;
begin clrscr;
    dr:=detect;
    initgraph(dr,md,'c:\programs\tp\hgi');
    d:=getmaxx;
    h:=getmaxy;
    line(round(d/2),0,round(d/2),h);
    line(0,round(h/2),d,round(h/2));
    x:=-3;
    y:=x/exp(x);
    dx:=0.1;
    x0:=round(d/2+x*10);
    y0:=round(h/2-y*10);
    moveto(x0,y0);
    while x<3 do
    begin
        x:=x+dx;
        y:=x/exp(x);
        x0:=round(d/2+x*10);
        y0:=round(h/2-y*10);
        lineto(x0,y0);
    end;
    x:=-3;
    y:=exp((1/3)*ln(x*(3-sqr(x)))));
    dx:=0.1;
    x0:=round(d/2+x*50);
    y0:=round(h/2-y*50);
    moveto(x0,y0);
    while x<3 do
    begin
        x:=x+dx;
        if (x*(3-sqr(x)))<0 then
            y:=-exp((1/3)*ln(abs(x*(3-sqr(x)))))
        else
            if (abs(x*(3-sqr(x))))=0 then
                x:=0
            else
                y:=exp((1/3)*ln(x*(3-sqr(x))));
        x0:=round(d/2+x*50);
        y0:=round(h/2-y*50);
        lineto(x0,y0);
    end;
end;
26:58
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu

```

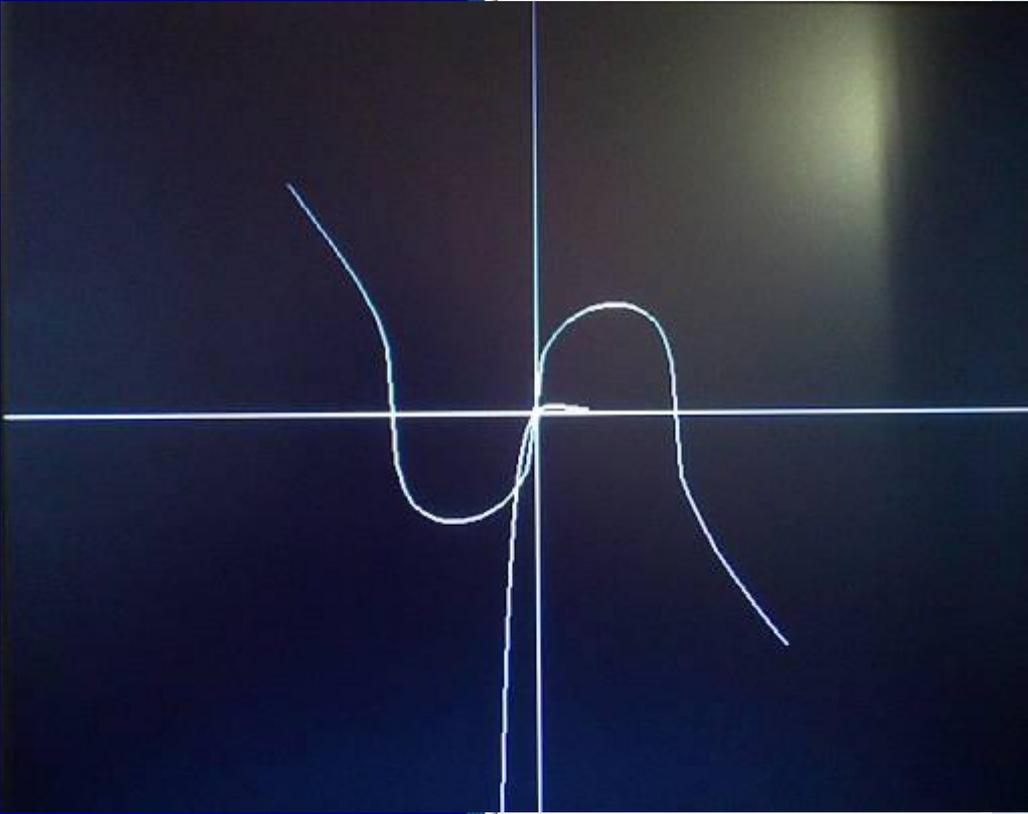


График функции

```
TeachPro Программирование на TurboPascal
File Edit Search Run Compile Debug Tools Options Window Help
\DOCUME~2\STUDENTS\МИ-02\PASCAL\ПРОГРА~1\LAB10\LAB101.PAS 1
\DOCUME~2\STUDENTS\МИ-02\PASCAL\ПРОГРА~1\LAB10\LAB102.PAS 2=[↑]
program abc;
uses crt,graph;
var dr,md:integer;
    x,y,r,i:integer;
procedure face;
begin
    setfillstyle(1,14);
    setcolor(14);
    fillellipse(x,y,r,r);
    delay(65000);
    setcolor(0);
    setfillstyle(1,0);
    fillellipse(x,y,r,r);
    y:=y+20;
end;
begin clrscr;
    r:=70;
    x:=100;
    y:=100;
    dr:=detect;
    initgraph(dr,md,'c:\programs\tp\bgi');
    ellipse(x,round(y+3*r),0,180,round(x),round(r));
    ellipse(300,round(x+3*r),0,180,round(y),round(r));
    ellipse(500,round(x+3*r),0,180,round(y),round(r));
face;
    y:=100;
    for i:=1 to 20 do
        begin face;
            {delay(65000);
            setfillstyle(1,0);}
            {y:=y+15;}
        end;
    setcolor(1);
    line(500,170,480,180);
    line(490,160,490,190);
    line(483,167,502,189);
    line(300,90,280,100);
    line(290,79,290,105);
    line(279,94,298,99);
    line(300,170,280,180);
    line(290,160,290,190);
    line(283,167,302,189);
    line(500,90,480,100);
    line(490,79,490,105);
    line(479,94,498,99);
1:1
```

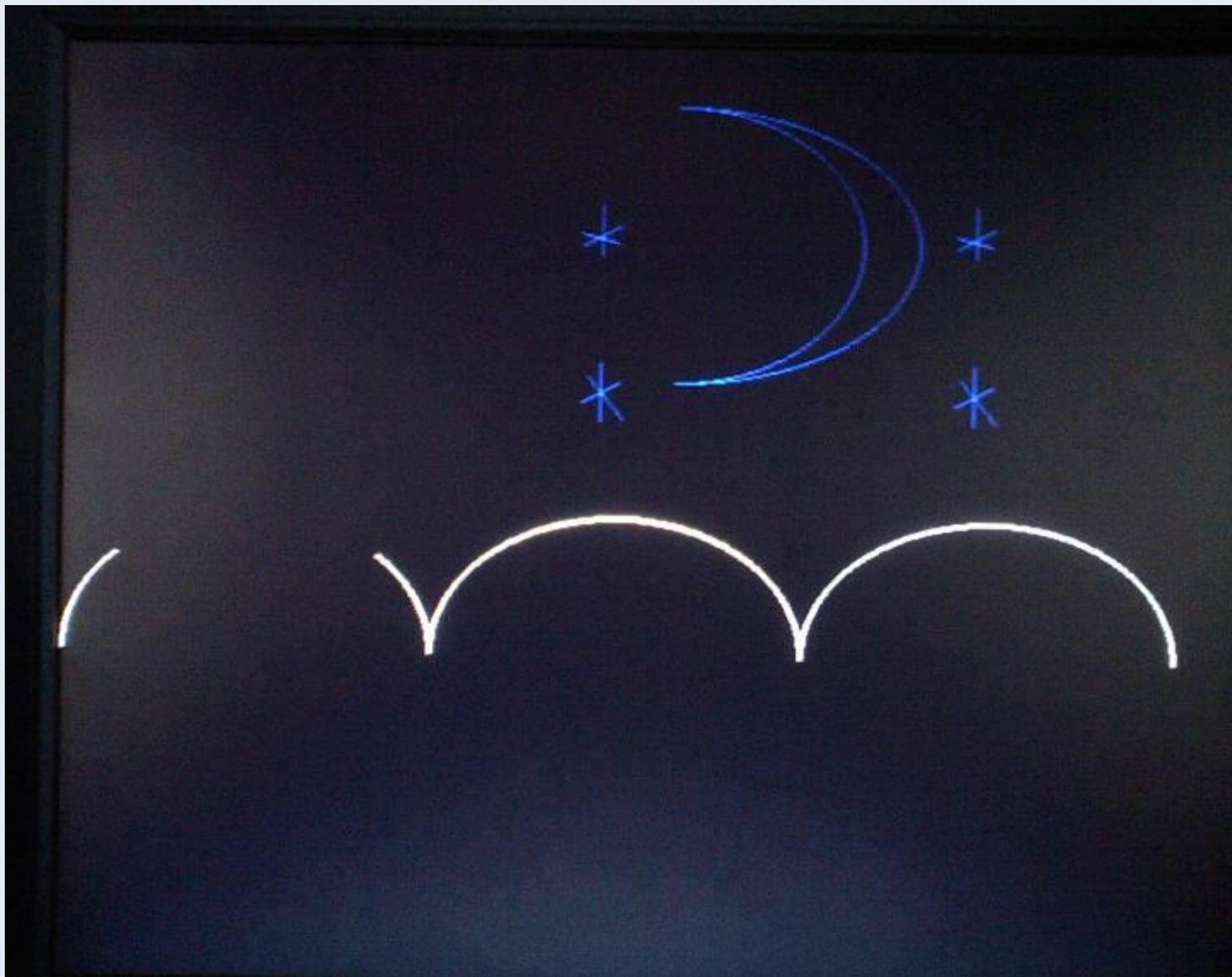
Солнце садится



F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10



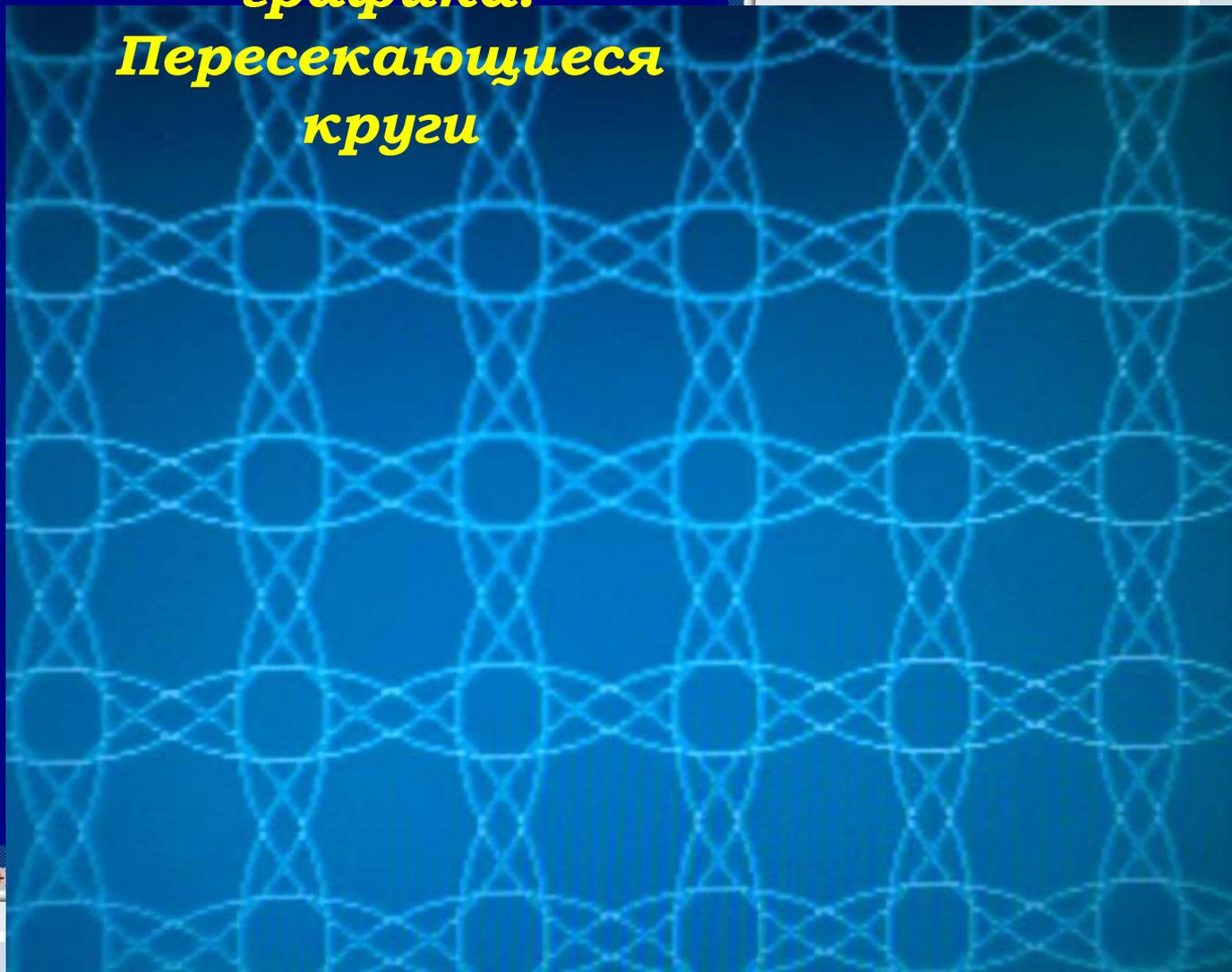
Появятся звезды и луна



```
TeachPro Программирование на TurboPascal
File Edit Search Run Compile Debug Tools Options Window Help
NONAME00.PAS
program abc;
uses graph;
var d,m,x,y:integer;
begin
d:=0; m:=0;
initgraph(d,m,'c:\pascal');
setbkcolor(3);
x:=0;
while x<=650 do
begin
y:=0;
while y<=500 do
begin
circle(x,y,60);
y:=y+50;
end;
x:=x+50;
end;
readln;
closegraph;
end.
```

Фрактальная графика.

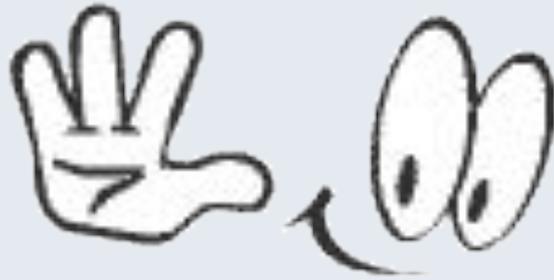
Пересекающиеся круги



```
TeachPro Программирование на TurboPascal
File Edit Search Run Compile Debug Tools Options Window Help
NONAME00.PAS
program okr;
uses graph;
var gd, gm, i, r: integer;
begin
gd:=0;
initgraph(gd, gm, 'c:\bp\bgi');
r:=5;
for i:=1 to 23 do
begin
setcolor(i);
circle(300, 240, i*10);
end;
readln;
end.
```

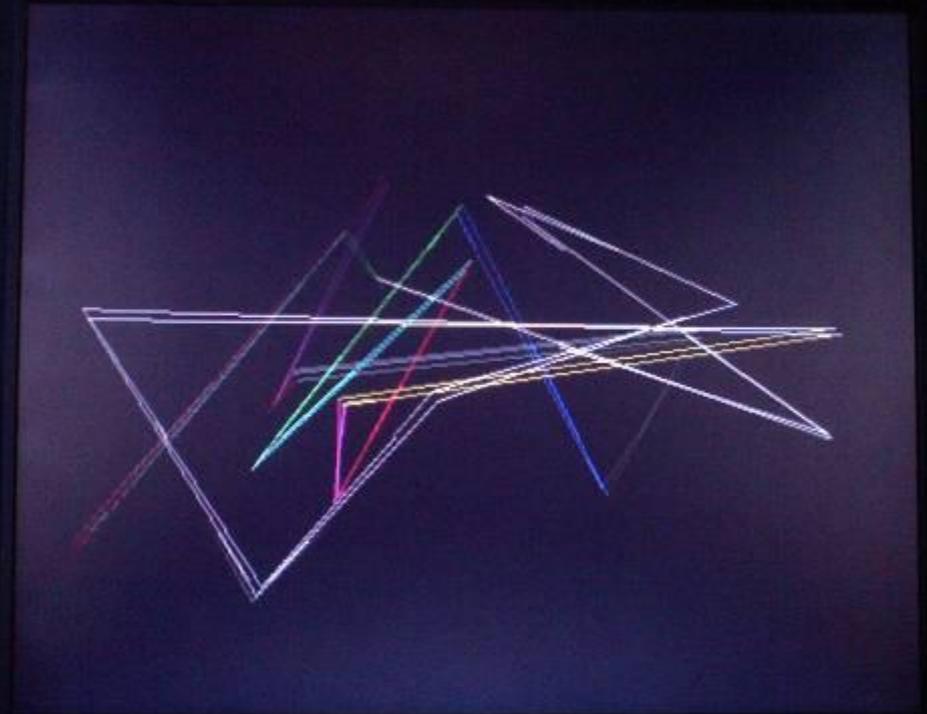
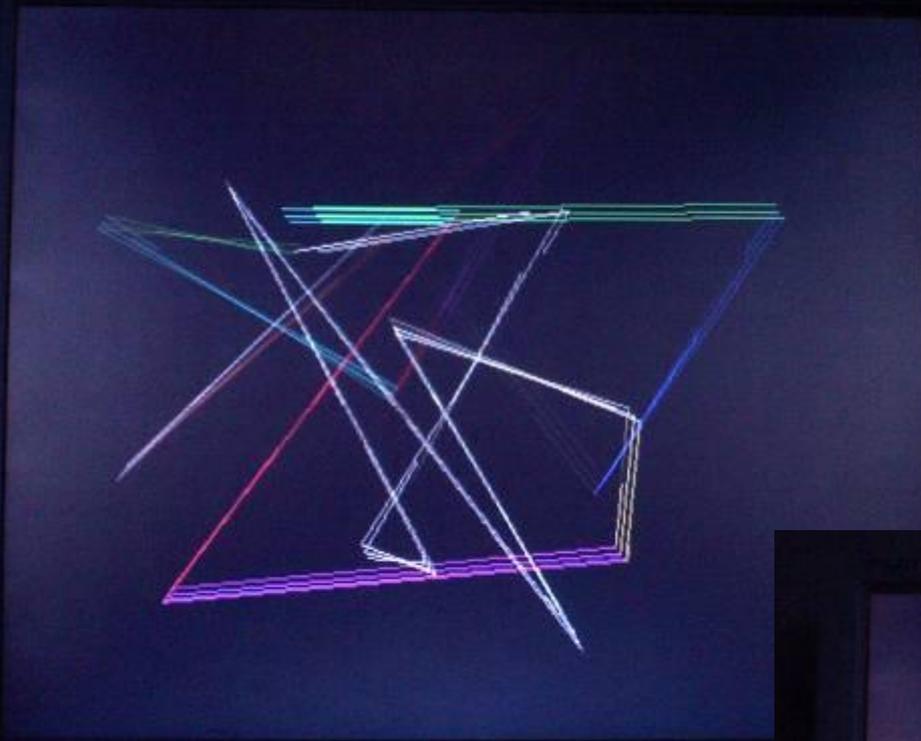
Концентрические окружности



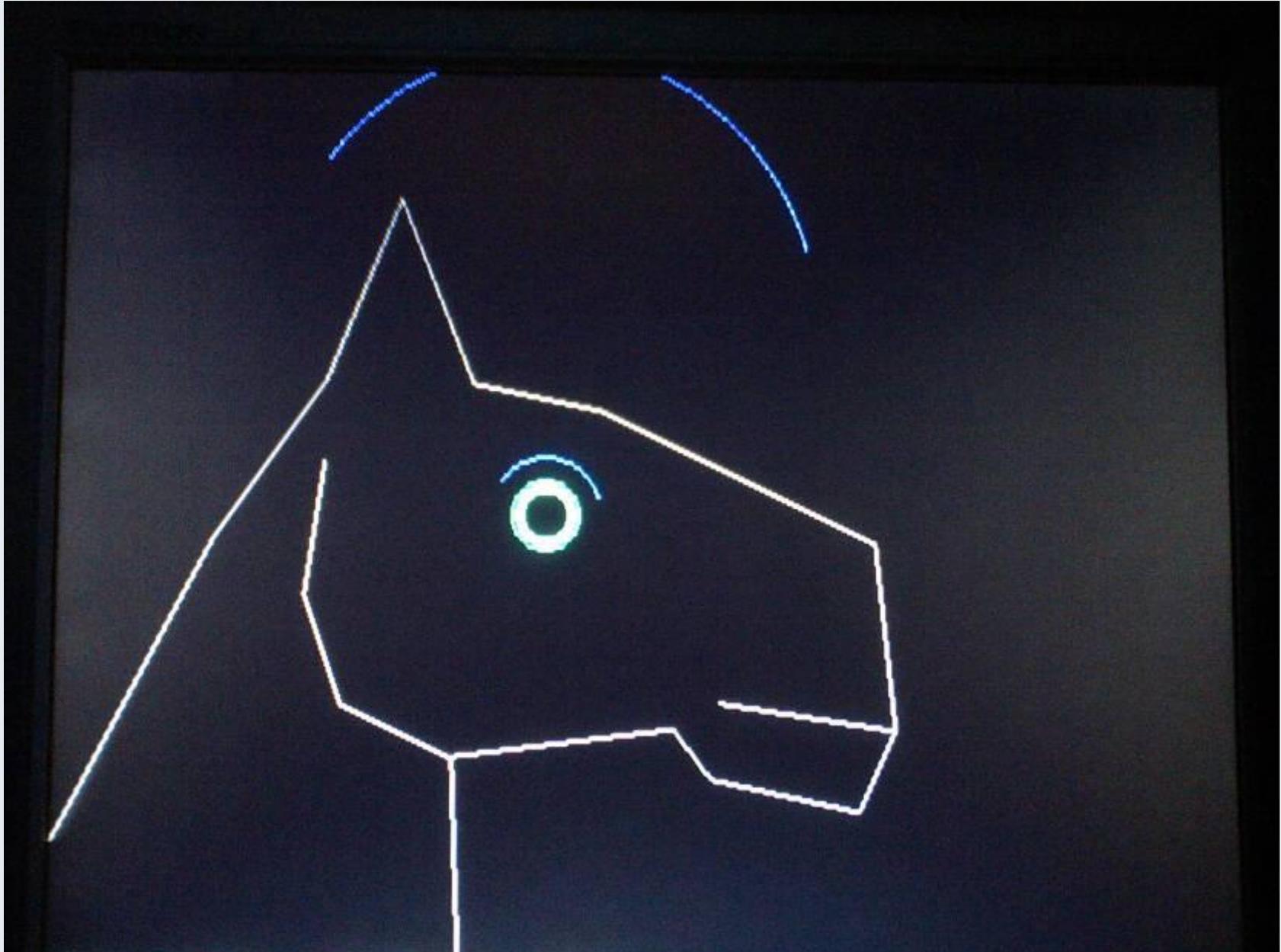


***Примеры графических
объектов,
созданных в Pascal***

*Движение
разноцветных
ломанных*



Лошадь



Говорит: «Лошадь, лошадь, лошадь»

o
E

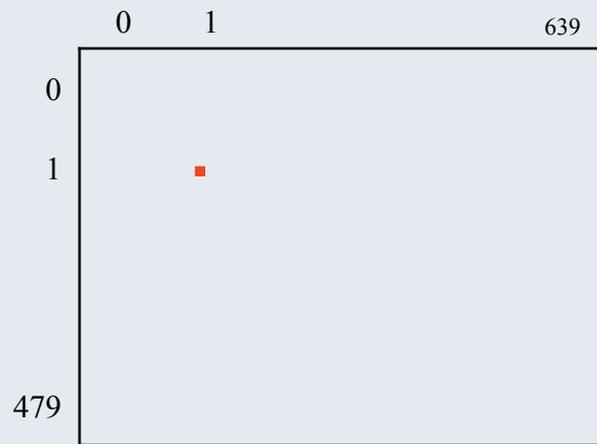
it

Режимы работы монитора

Текстовый режим

Графический режим

	1	2				80
1						
2			A			
25						



Знакомство

Пиксель

Библиотека GRAPH

Библиотека (модуль) GRAPH содержит константы, процедуры и функции для управления графическим режимом работы монитора.

Константы цвета

Black = 0; {Черный}	DarkGray = 8; {Темносерый}
Blue = 1; {Синий}	LightBlue = 9; {Яркосиний}
Green = 2; {Зеленый}	LightGreen = 10; {Яркозеленый}
Cyan = 3; {Голубой}	LightCyan = 11; {Яркоголубой}
Red = 4; {Красный}	LightRed = 12; {Розовый}
Magenta = 5; {Фиолетовый}	LightMagenta = 13; {Малиновый}
Brown = 6; {Коричневый}	Yellow = 14; {Желтый}
LightGray = 7; {Светлосерый}	White = 15; {Белый}

Константы типов и толщины линий

SolidLn = 0;
{Сплошная}

DottedLn = 1;
{Точечная}

CenterLn = 2;
{Штрихпунктирная}

DashedLn = 3;
{Пунктирная}

NormWidth=1;
{Нормальная толщина}

ThickWidth = 3;
{Тройная толщина}

Константы шаблона штриховки

EmptyFill = 0;	{Заполнение цветом фона}
SolidFill = 1;	{Сплошная штриховка}
LineFill = 2;	{Горизонтальная штриховка}
LtSlashFill = 3;	{/// штриховка}
SlashFill = 4;	{/// штриховка толстыми линиями}
BkSlashFill = 5;	{\\ \\ штриховка толстыми линиями}
LtBkSlashFill = 6;	{\\ \\ штриховка}
HatchFill = 7;	{Заполнение прямой клеткой}
XHatchFill = 8;	{Заполнение косой клеткой}
InterleaveFill = 9;	{Заполнение частой сеткой}
WideDotFill = 10;	{Заполнение редкими точками}
CloseDotFill = 11;	{Заполнение частыми точками}
UserFill = 12.	{Тип задается пользователем}

Установка режима

InitGraph (Var Driver, Mode: Integer; Path: String)

Организует переход в графический режим. Переменные Driver и Mode содержат тип графического драйвера и его режим работы. Третий параметр определяет маршрут поиска графического драйвера. Если строка пустая (т. е. равна ""), считается, что драйвер находится в текущем каталоге.

Практически:

```
uses Graph;  
  
var  
  
    Driver, Mode : integer;  
  
begin  
  
    Driver:=Detect;  
  
    InitGraph(Driver, Mode, '');  
  
    . . .  
  
    CloseGraph;  
  
end.
```

Установка режима

CloseGraph

Закрывает графический режим, т.е. освобождает память, распределенную под драйверы графики и файлы шрифтов, и восстанавливает текстовый режим работы экрана.

ClearDevice

Очищает графический экран, закрашивает его в цвет фона.

Функции

GetMaxX

и

GetMaxY

Возвращает значения максимальных координат экрана в текущем режиме работы, соответственно, по горизонтали и вертикали.

Установка цвета, координат и стилей

MoveTo (X, Y: Integer)

Смещает текущий указатель к точке X,Y.

SetColor (Color: Word)

Устанавливает основной цвет, которым будет осуществляться рисование.

SetBkColor (Color: Word)

Устанавливает цвет фона.

SetFillStyle (Pattern, Color: Word)

Устанавливает образец штриховки и цвет.

SetLineStyle (LineStyle, Pattern, Thickness: Word)

Устанавливает толщину и стиль линии.

SetTextStyle (Font, Direction, CharSize: Word)

Устанавливает текущий шрифт, направление (горизонтальное или вертикальное) и размер текста.

Вывод на графический экран

OutTextXY(X, Y: Integer; TextString: String)

Выводит текст в заданное место экрана.

PutPixel(X, Y: Integer; Color: Word)

Выводит точку цветом Color с координатами X, Y.

Line(X1, Y1, X2, Y2: Integer)

Рисует линию от точки X1, Y1 до точки X2, Y2.

LineTo(X, Y: Integer)

Рисует линию от текущего указателя к точке X, Y.

Rectangle(X1, Y1, X2, Y2)

Рисует контур прямоугольника, используя текущий цвет и тип линии. X1, Y1 - координаты левого верхнего угла прямоугольника, X2, Y2 - координаты правого нижнего угла прямоугольника.

Вывод на графический экран

Circle(X, Y: Integer; R: Word)

Рисует текущим цветом окружность радиуса R с центром в точке (X,Y).

Arc(X, Y: Integer; U1, U2, R: Word)

Строит дугу окружности текущим цветом с текущими параметрами линии. X, Y - координаты центра дуги, U1 - угол до начальной точки дуги, отсчитываемый против часовой стрелки от горизонтальной оси, направленной слева направо, U2 - угол до конечной точки дуги, отсчитываемый так же, как U1, R - радиус дуги.

FillEllipse(X, Y: Integer; XR, YR: Word)

Рисует заштрихованный эллипс, используя X,Y как центр и XR,YR как горизонтальную и вертикальную полуоси эллипса.

Ellipse(X, Y: Integer; U1, U2, XR, YR: Word)

Рисует дугу эллипса текущим цветом; X, Y - координаты центра эллипса; U1, U2 - углы до начальной и конечной точек дуги эллипса (см. процедуру Arc); XR, YR - горизонтальная и вертикальная полуоси эллипса.

Вывод на графический экран

Sector ($X, Y: \text{Integer}; U1, U2, XR, YR: \text{Word}$)

Рисует и штрихует сектор эллипса радиусами XR, YR с центром в X, Y от начального угла $U1$ к конечному углу $U2$.

PieSlice ($X, Y: \text{Integer}; U1, U2, \text{Radius}: \text{Word}$)

Строит сектор круга, покрашенный текущей штриховкой и цветом заполнения. X, Y - координаты центра сектора круга; $U1$ и $U2$ - начальный и конечный углы сектора, отсчитываемые против часовой стрелки от горизонтальной оси, направленной вправо; Radius - радиус сектора.

Вывод на графический экран

Var (X1, Y1, X2, Y2: Integer)

Строит прямоугольник, закрашенный текущим цветом с использованием текущего стиля (орнамента, штриховки). X1, Y1, X2, Y2 - координаты левого верхнего и правого нижнего углов прямоугольника.

Var3D (X1, Y1, X2, Y2: Integer; Glubina: Word; Top: Boolean)

Строит параллелепипед, используя текущий стиль и цвет. X1, Y1, X2, Y2 - координаты левого верхнего и правого нижнего углов передней грани; Glubina - ширина боковой грани (отсчитывается по горизонтали), Top - признак включения верхней грани (если True - верхняя грань вычерчивается, False - не вычерчивается).

FillPoly (N: Word; Var PolyPoints)

Рисует и штрихует многоугольник, содержащий N вершин с координатами в PolyPoints (целочисленный массив с 2N элементами).

FloodFill (X, Y: Integer; Border: Word);

Заполняет произвольную замкнутую фигуру, используя текущий стиль заполнения (узор и цвет). X, Y - координаты любой точки внутри замкнутой фигуры; Border - цвет граничной линии.

Алгоритм рисования фигуры

1. Задать параметры (координаты)
2. Выбрать цвет фигуры
3. Нарисовать фигуру
4. Закрасить фигуру
5. Задержать изображение

Заполнение экрана окружностями

```
uses graph,crt;
var Driver,Mode,x1,y1,r,c,gm:integer;
begin
  Driver:=Detect;
  InitGraph(Driver, Mode, '');
  randomize;
  repeat
    x1:= random(640);
    r:=random(100);
    y1:=random(480);
    c:=random(16);
    setcolor(c);
    circle(x1,y1,r);
    delay(100);
  until keypressed;
end.
```

Заполнение экрана закрасенными кругами

```
uses graph,crt;
var Driver,Mode,x1,y1,r,c,gm:integer;
begin
    Driver:=Detect;
    InitGraph(Driver, Mode, '');
    randomize;
    repeat
        x1:= random(640);
        r:=random(100)+2;
        y1:=random(480);
        c:=random(16);
        setcolor(c);
        circle(x1,y1,r);
        setfillstyle(10,c);
        floodfill(x1,y1,c);
        delay(100);
    until keypressed;
end.
```

Построение графика функций

1. Построить оси в центре экрана
2. Построить координатную сетку
 - a) Сетка по оси X (от центра)
 - b) Сетка по оси Y (от центра)
3. Построить график

Анимация объекта

1. Установить начальное положение (координаты)
- 2. Нарисовать фигуру:
 - a) Выбрать цвет фигуры
 - b) Нарисовать фигуру
 - c) Закрасить фигуру
3. Задержать изображение
4. Нарисовать фигуру цветом фона
5. Изменить координаты