

Програми за проста анимация

{ Движение на точка (представена с малка окръжност) по хоризонтала }

```
program Running_Ball;
```

```
uses Crt, Graph;
```

```
var
```

```
c:char;
```

```
x,y:integer;
```

```
const
```

```
xl:integer=60;
```

```
xr:integer=570;
```

```
procedure Initialize;      { Инициализира графичната система }
```

```
var GraphDriver, GraphMode, ErrorCode:integer;

begin

GraphDriver:=Detect;

InitGraph(GraphDriver, GraphMode, 'c:bpbgj');

ErrorCode:=GraphResult;

if ErrorCode grOK then

begin

WriteLn('Графична грешка', GraphErrorMsg(ErrorCode));

Halt(1);

end;

end;

begin
```

Initialize;

Circle(xl,100,5); readln;

for x:=xl to xr do

begin

SetColor(GetMaxColor);

Circle(x,100,5);

Delay(20);

SetColor(0);

Circle(x,100,5);

end;

readln;

CloseGraph;

end.

{ Пример за анимация с междинни състояния }

program InBetweens;

uses Graph, Crt;

const

Length=5; { Брой върхове на многоъгълника }

type

ArrayType=array[1..Length]of PointType;

const

NumSteps:integer=100;

StartPoints:ArrayType=

```
((x:0;y:0),(x:100;y:0),(x:100;y:20),
```

```
(x:0;y:20),(x:0;y:0));
```

```
StopPoints :ArrayType=
```

```
((x:400;y:100),(x:500;y:100),(x:500;y:120),
```

```
(x:400;y:120),(x:400;y:100));
```

```
var
```

```
InBetween:ArrayType;
```

```
i:integer;
```

```
procedure InBetweenPoints;
```

```
var
```

```
StepSize,IncAmount:real;
```

```
i,j:integer;

begin

StepSize:=1/(NumSteps-1.0);

for i:=1 to NumSteps do begin

IncAmount:=(i-1)*StepSize;

for j:=1 to Length do begin

InBetween[j].x:=StartPoints[j].x +

round(IncAmount*(StopPoints[j].x-StartPoints[j].x));

InBetween[j].y:=StartPoints[j].y +

round(IncAmount*(StopPoints[j].y-StartPoints[j].y));

end;

DrawPoly(Length,Inbetween);
```

```
delay(100);
```

```
DrawPoly(Length,InBetween);
```

```
end;
```

```
DrawPoly(Length,InBetween);
```

```
end;
```

```
begin {main}
```

```
Initialize; { В програмата Running_Ball по-горе }
```

```
SetWriteMode(XorPut);
```

```
InBetweenPoints;
```

```
readln;
```

```
CloseGraph;
```

end.

{ Ако

StartPoints:ArrayType=

((x:0;y:0),(x:639;y:0),(x:639;y:199),

(x:0;y:199),(x:0;y:0));

StopPoints :ArrayType=

((x:210;y:120),(x:315;y:60),(x:420;y:120),

(x:210;y:120),(x:210;y:120));

правоъгълник се преобразува в триъгълник }

{ Анимация с използване на част от вече растеризирания образ със съхранение на задния план }

program Ball;

{ Пример за анимация с AND и XOR маски }


```
uses Graph, Crt;
```

```
const Step=2;
```

```
DelayTime=50;
```

```
var
```

```
XorMask,AndMask,Covered:pointer;
```

```
procedure DrawBall;
```

```
begin
```

```
GetMem(Covered,ImageSize(35,85,65,115));
```

```
GetMem(XorMask,ImageSize(35,85,65,115));
```

```
GetMem(AndMask,ImageSize(35,85,65,115));
```

```
GetImage(35,85,65,115,Covered^);
```

```
{ Първо се създава маската AndMask }
```

```
SetFillstyle(SolidFill,GetMaxColor);
```

```
Bar(35,85,65,115);
```

```
SetColor(0);
```

```
SetFillStyle(SolidFill,0);
```

```
PieSlice(50,100,0,360,12);
```

```
GetImage(35,85,65,115,AndMask^);
```

```
{ Създаване на маската XorMask }
```

```
PutImage(35,85,Covered^,CopyPut);
```

```
SetColor(GetMaxColor);
```

```
SetFillStyle(SolidFill,GetMaxColor);
```

```
PieSlice(50,100,0,360,12);
```

```
SetFillStyle(HatchFill,1);
```

```
SetColor(1);
```

```
Circle(50,100,6);
```

```
Floodfill(50,100,1);
```

```
GetImage(35,85,65,115,XorMask^);
```

```
{ Изтриване на топката и изчертаване на задния план }
```

```
PutImage(35,85,Covered^,CopyPut);
```

```
SetFillStyle(HatchFill,2);
```

```
Bar3d(100,10,150,199,0,False);
```

```
{ Извеждане на топката върху екрана чрез съхраняване на }
```

```
{ задния план и използване на AndMask и XorMask }
```

```
GetImage(35,85,65,115,Covered^);
```

```
PutImage(35,85,AndMask^, AndPut);
```

```
PutImage(35,85,XorMask^, XorPut);
```

```
end;
```

```
Procedure Moveball;
```

```
var
```

```
i,Move:integer;
```

```
begin
```

```
for i:=0 to 150 do begin
```

```
Move:=i*Step;
```

```
{ Връща задния план на мястото на топката }
```

```
PutImage(35+Move,85,Covered^,CopyPut);
```

```
{ Съхранява задния пран на новото място на топката }
```

```
GetImage(35+Step+Move,85,65+Step+Move,115,Covered^);
```

```
{ Рисува топката на новото място с двете маски }
```

```
PutImage(35+Step+Move,85,AndMask^,AndPut);
```

```
PutImage(35+Step+Move,85,XorMask^,XorPut);
```

```
Delay(DelayTime);
```

```
end;
```

```
end;
```

```
Begin {main}
```

```
Initialize; { В програмата Running_Ball по-горе }
```

```
DrawBall; readln;
```

MoveBall;

readln;

CloseGraph;

end.