

Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Чувашский государственный университет имени И. Н. Ульянова»

Факультет информатики и вычислительной техники

Кафедра математического и аппаратного обеспечения информационных  
систем

### **Методические указания**

для выполнения лабораторных работ  
по дисциплине «Математические основы компьютерной графики»

Составители Ильина Л.А., Ильин Д.В.

## Лабораторная работа 1. Создание графических изображений с использованием графических операторов языков программирования

Нарисовать произвольный рисунок с использованием максимального количества графических функций языка программирования.

**Пример.** Нарисуем изображение дома и дерева с использованием максимального количества графических процедур и функций (рисование окружностей, эллипсов, прямых, прямоугольников, дуг, закрашенных фигур).

```
program dom;
uses Graph;
const l=120;h1=75;h2=65;h3=50;lo1=40;a=10;p1=15;

ho1=50;lo2=81;ho2=45;d=15;p2=10;r1=15;r2=20;

ras=100;ld=10;hd=65;hk=100;rd1=25;rd2=55;xn=150;
yn=195;
var
grDriver,grMode,ErrCode,x,y,x1,y1,x2,y2:integer;
begin
grDriver := Detect;
  InitGraph(grDriver, grMode,'egavga.bgi');
  ErrCode := GraphResult;
  if ErrCode = grOk then
    begin

      setcolor(3);
      rectangle(xn,yn,xn+1,yn-h1);
      x:=xn+ 1 div 2-a;
      y:=yn-p1;
      x1:=x-lo1;
      y1:=y-ho1;
      rectangle(x,y,x1,y1);{}
      x2:=x-lo1 div 2;
      y2:=y-ho1 div 2;
      {}
      line(x,y2,x2,y);
```

```

line(x2,y,x1,y2);
line(x1,y2,x2,y1);
line(x2,y1,x,y2);
x:=x+2*a;
x1:=x+lo1;
rectangle(x,y,x1,y1);
x2:=x+lo1 div 2;
{}
line(x,y2,x2,y1);
line(x2,y1,x1,y2);
line(x1,y2,x2,y);
line(x2,y,x,y2);
{}
x:=xn;
y:=yn-h1;
rectangle(x,y,x+1,y-h2);
x:=x+(1-lo2) div 2;
y:=y-p2;
rectangle(x,y,x+lo2,y-ho2);
y1:=y-ho2+d;
line(x,y1,x+lo2,y1);
x:=x+lo2 div 3;
line(x,y,x,y1);
x:=x+lo2 div 3;
line(x,y,x,y1);
y:=yn-(h1+h2);
x:=xn+1 div 2;
ellipse(x,y,0,180,1 div 2,h3);
y:=y-h3 div 2;
ellipse(x,y,0,360,r1,r2);
line(x-r1,y,x+r1,y);
line(x,y-r2,x,y+r2);
x:=xn+1+ras;
y:=yn;
rectangle(x,y,x+ld,y-hd);{}
x1:=x+ld div 2;
y1:=y-hk;
ellipse(x1,y1,0,255,rd1,rd2);
ellipse(x1,y1,285,360,rd1,rd2);

```

```

    setfillstyle(10,1);
    floodfill(x1,y1,3);
    line(0,yn,getmaxx,yn);
    readln;
end;
closegraph;
end.

```

## **Лабораторная работа 2. Создание движущихся графических изображений**

Написать программу движения:

1. Машины по круговой траектории по часовой стрелке.
2. Машины по эллипсоидной траектории по часовой стрелке.
3. Машины слева направо по прямой дороге.
4. Машины справа налево вдоль прямой.
5. Машины сверху вниз вдоль прямой.
6. Машины снизу вверх вдоль прямой.
7. Машины вдоль синусоиды.
8. Машины по прямоугольной траектории.
9. Самолета по прямой траектории слева направо.
10. Самолета по прямой траектории справа налево.
11. Самолета по прямой траектории сверху вниз.
12. Самолета по прямой траектории снизу вверх.
13. Самолета по круговой траектории по часовой стрелке.
14. Самолета по круговой траектории против часовой стрелки.
15. Самолета по диагонали экрана снизу вверх.
16. Самолета по диагонали экрана сверху вниз.
17. Самолета управляемого стрелками вправо влево.
18. Самолета управляемого стрелками вверх и вниз.
19. Автобуса управляемого стрелками вправо влево.
20. Автобуса управляемого стрелками вверх и вниз.
21. Мяча прыгающего на высоту  $h$ , высота задается случайным образом.
22. Удаляющегося самолета (снизу вверх по центру экрана).
23. Приближающегося самолета (сверху вниз по центру экрана).
24. Приближающегося автобуса.

25. Удаляющегося по диагонали (из правого нижнего в левый верхний угол) автобуса.

26. Удаляющегося по диагонали (из левого нижнего в правый верхний угол) автобуса.

27. Падающей снежинки.

28. Облака по небу.

29. Минутной стрелки по циферблату часов.

30. Качающихся качелей.

**Пример.** Написать программу движения ракеты, основанного на операциях поворота и масштабирования. Ракета совершает сначала один полный оборот по круговой траектории по часовой стрелке, а затем движется прямолинейно слева направо, уменьшаясь в размерах. За центр вращения принята центральная точка экрана, за центр масштабирования – точка, имеющая максимальную абсциссу и ординату, отстоящую на расстояние радиуса орбиты от центра.

Обозначения в программе:

$l$  – высота первой ступени ракеты;  $r$  – радиус орбиты;

$d$  – диаметр ракеты;

$ls$  – высота второй ступени ракеты;

$x, y$  – массивы исходных координат вершин ракеты;

$x1, y1$  – массивы преобразованных координат ракеты;

$xn, yn$  – координаты начальной точки (правая верхняя вершина ракеты);

$xs, ys$  – координаты центра орбиты;

$xm, ym$  – координаты центра масштабирования;

$xa, ya$  – коэффициенты, учитывающие разные разрешающие способности вдоль координатных осей;

$t$  – текущее значение угла поворота;

$k$  – текущее значение коэффициента масштабирования.

```
program raketa;  
uses graph;crt;  
const l=60;r=120;d=20;ls=15;
```

```

type mas=array[1..15] of integer;
var x,y,x1,y1:mas;
gd,gm,xn,yn,xc,yc,xm,ym,i,j:integer;
xa,ya:word;
t,k:real;  ErrCode:integer;
{процедура определения координат повернутой точки}
procedure pow(xc,yc,x,y:integer;var
x1,y1:integer;t:real);
begin
x1:=xc+round((x-xc)*cos(t)+(y-yc)*sin(t)*ya/xa);
y1:=yc+round((y-yc)*cos(t)-(x-xc)*sin(t)*xa/ya);
end;
{процедура определения координат промасштабированной точки}
procedure massch(xc,yc,x,y:integer;k:real;var
x1,y1:integer);
var r:real;
begin
r:=1-k;
x1:=round(x*k+r*xc);
y1:=round(y*k+r*yc);
end;
{процедура высвечивания ракеты}
procedure raket(x,y:mas);
var i:integer;
begin
line(x[1],y[1],x[2],y[2]);
line(x[2],y[2],x[3],y[3]);
line(x[3],y[3],x[4],y[4]);
line(x[4],y[4],x[1],y[1]);
line(x[4],y[4],x[5],y[5]);
line(x[1],y[1],x[5],y[5]);
line(x[5],y[5],x[4],y[4]);
for i:=1 to 2 do
begin
line(x[6+(i-1)*5],y[6+(i-1)*5],x[7+(i-1)*5],y[7+(i-1)*5]);

```

```

line(x[7+(i-1)*5],y[7+(i-1)*5],x[8+(i-1)*5],y[8+(i-
1)*5]);
line(x[8+(i-1)*5],y[8+(i-1)*5],x[9+(i-1)*5],y[9+(i-
1)*5]);
line(x[9+(i-1)*5],y[9+(i-1)*5],x[10+(i-1)*5],y[10+(i-
1)*5]);
end;
end;
begin
gd:=detect;
  InitGraph(gd, gm,'egavga.bgi');
  ErrCode := GraphResult;
  if ErrCode = grOk then
begin setcolor(12);
{определение поправочных коэффициентов, учиты-
вающих}
{разные разрешающие способности вдоль осей X и
Y}
GetAspectRatio(xa,ya);
xn:=getmaxx div 2;
yn:=getmaxy div 2-r;
xm:=getmaxx; {координаты центра}
ym:=yn;      {масштабирования}
xs:=xn;      {координаты центра}
ys:=yn+r;    {вращения}
{задание начальных координат точек ракеты}
x[1]:=xn;y[1]:=yn;
x[2]:=xn-1;y[2]:=yn;
x[3]:=xn-1;y[3]:=yn+d;
x[4]:=xn;y[4]:=yn+d;
x[5]:=xn+1s;y[5]:=yn+d div 2;
x[6]:=xn-round(3*1/4);y[6]:=yn;
x[7]:=xn-1;y[7]:=yn-round(d/4);
x[8]:=xn-1-1 div 2;y[8]:=yn-round(d/4);
x[9]:=xn-round(5*1/4);y[9]:=yn+round(d/8);
x[10]:=xn-1;y[10]:=yn+round(d/8);
x[11]:=xn-round(3*1/4);y[11]:=yn+d;
x[12]:=xn-1;y[12]:=yn+d+round(d/4);
x[13]:=xn-1-1 div 2;y[13]:=yn+round(5*d/4);

```

```

x[14]:=xn-1-round(1/4);y[14]:=yn+round(7*d/8);
x[15]:=xn-1;y[15]:=yn+round(7*d/8);
{движение ракеты по круговой траектории вокруг
центра вращения по часовой стрелке один оборот}
for i:=1 to 37 do
begin
t:=(i-1)*10*pi/180;{угол поворота}
for j:=1 to 15 do {10 градусов}
row(xc,yc,x[j],y[j],x1[j],y1[j],-t);
raket(x1,y1);{рисование ракеты}
delay(100);{задержка перед стиранием}
cleardevice;{стирание изображения}
end;
raket(x1,y1);
{движение ракеты по прямой с уменьшением разме-
ров}
for i:=1 to 20 do
begin
k:=1-0.05*i;{коэффициент масштабирования}
for j:=1 to 15 do
massch(xm,ym,x[j],y[j],k,x1[j],y[j]);
raket(x1,y1);{рисование масштабированной ракеты}
delay(100);{задержка перед стиранием}
cleardevice;{стирание изображения}
end;
closegraph;
end else writeln('error');
end.

```

### **Лабораторная работа 3. Использование функций сохранения и выдачи изображений**

Нарисовать орнамент, используя функции сохранения и выдачи изображений.

**Пример.** В левой верхней части экрана построить изображение «строительного блока», представляющего собой прямоугольник, внутри которого находится прямоугольное окно. Внутри прямоугольного окна имеется ромбовидное окно. Построенное изо-



бражение запомнить и использовать для построения фрагмента дома из трех этажей по три блока в каждом.

Обозначения в программе:

h - высота блока;

l - ширина блока;

ras - расстояние от пола до окна;

ho - высота окна;

lo - ширина окна;

xn, yn - координаты левого верхнего угла блока дома

```
program okno;
uses graph;
const h=60;l=80;ras=20;ho=30;lo=40;
var gd,gm,x,y,xn,yn,s,i,j:integer;
    p:pointer;
begin
  gd:=detect;
  initgraph(gd,gm,' ');
  xn:=0; yn:=0;
  rectangle(xn,yn,xn+l,yn+h); {}
  x:=(l-lo)div 2;
  y:=h-ras-ho;
  yn:=yn+y;
  rectangle(xn+x,yn,xn+x+lo,yn+ho); {}
  xn:=xn+(l-lo) div 2;
  yn:=yn+ho div 2;
  moveto(xn,yn);
  linerel(lo div 2,-ho div 2); {}
  linerel(lo div 2,ho div 2);
  linerel(-lo div 2,ho div 2);
  linerel(-lo div 2,-ho div 2);
  xn:=0; yn:=0;
  s:=imagesize(xn,yn,xn+l,yn+h);
  getmem(p,s);
  getimage(xn,yn,xn+l,yn+h,p^);
  xn:=2*l;yn:=2*h;
  for i:=1 to 3 do
  for j:=1 to 3 do
```

```

    putimage (xn+(i-1)*l, yn+(j-1)*h, p^, 0);
    readln;
    closegraph;
end.

```

#### **Лабораторная работа 4. Создание движущихся объемных изображений с использованием матриц преобразования в пространстве**

Написать программу движения и вращения в пространстве двух букв алфавита, соответствующих имени и фамилии.

**Пример.** Написать программу движения объемной буквы и ее вращения в пространстве.

```

uses Graph, crt;
var
h, ox1, oy1, ox2, oy2, x1, y1, z1, x2, y2, z2, i, j, k, l, sx1, sy1: longint;
    omn, mn, a, b, c: real;
    ch: char;
    Tr: array[1..10] of PointType;
    so: array[1..10] of record num, r: longint end;
    page: byte;
const
    grad=3.14/100;
    shag=2;
    pa=50;
    xyz: array[1..48] of record x, y, z: longint end=
    ((x:-30;y:-30;z:10), (x:30;y:-30;z:10), (x:30;y:30;z:10),
    (x:10;y:30;z:10),
    (x:10;y:-10;z:10), (x:-10;y:-10;z:10), (x:-
    10;y:30;z:10), (x:-30;y:30;z:10),
    (x:-30;y:-30;z:-10), (x:30;y:-30;z:-10), (x:30;y:30;z:-
    10), (x:10;y:30;z:-10),
    (x:10;y:-10;z:-10), (x:-10;y:-10;z:-10), (x:-10;y:30;z:-
    10), (x:-30;y:30;z:-10),
    (x:-30;y:-30;z:-10), (x:-30;y:-30;z:10), (x:-
    30;y:30;z:10), (x:-30;y:30;z:-10),

```

```

(x:30;y:-30;z:-10) , (x:30;y:-
30;z:10) , (x:30;y:30;z:10) , (x:30;y:30;z:-10) ,
(x:-30;y:-30;z:-10) , (x:30;y:-30;z:-10) , (x:30;y:-
30;z:10) , (x:-30;y:-30;z:10) ,
(x:-10;y:-10;z:-10) , (x:10;y:-10;z:-10) , (x:10;y:-
10;z:10) , ( x:-10;y:-10;z:10) ,
(x:-30;y:30;z:-10) , (x:-10;y:30;z:-10) , (x:-
10;y:30;z:10) , (x:-30;y:30;z:10) ,
(x:10;y:30;z:-10) , (x:30;y:30;z:-10) , (x:30;y:30;z:10) ,
(x:10;y:30;z:10) ,
(x:-10;y:-10;z:-10) , (x:-10;y:-10;z:10) , (x:-
10;y:30;z:10) , (x:-10; y:30;z:-10) ,
(x:10;y:-10;z:-10) , (x:10;y:-
10;z:10) , (x:10;y:30;z:10) , (x:10;y:30;z:-10) ) ;
k1:array[1..10] of
  record
    p:array[1..8] of longint;
    x,y,z,n,color:longint;
  end=( (p: (1,2,3,4,5,6,7,8);x:0;y:0;z:10;n:8;color:15) ,
        (p: (9,10,11,12,13,14,15,16);x:0;y:0;z:-
10;n:8;color:14) ,
        (p: (17,18,19,20,0,0,0,0);x:-30;y:0;z:-
5;n:4;color:13) ,
        (p: (21,22,23,24,0,0,0,0);x:30;y:0;z:-
5;n:4;color:12) ,
        (p: (25,26,27,28,0,0,0,0);x:0;y:-30;z:-
5;n:4;color:11) ,
        (p: (29,30,31,32,0,0,0,0);x:0;y:-10;z:-
5;n:4;color:10) ,
        (p: (33,34,35,36,0,0,0,0);x:-20;y:30;z:-
5;n:4;color:9) ,
        (p: (37,38,39,40,0,0,0,0);x:20;y:30;z:-
5;n:4;color:8) ,
        (p: (41,42,43,44,0,0,0,0);x:-10;y:10;z:-
5;n:4;color:7) ,
        (p: (45,46,47,48,0,0,0,0);x:10;y:10;z:-
5;n:4;color:6) ) ;
procedure pre(var ax1,ay1,az1,x,y,z:longint);
var x1,y1,z1,x2,y2,z2:longint;

```

```

begin
    x1:=ax1;y1:=ay1;z1:=az1;
    y2:=round(y1*cos(c)+x1*sin(c));
    x2:=round(x1*cos(c)-y1*sin(c));
    x1:=x2;y1:=y2;
    z2:=round(z1*cos(b)+x1*sin(b));
    x2:=round(x1*cos(b)-z1*sin(b));
    x1:=x2;z1:=z2;
    z2:=round(z1*cos(a)+y1*sin(a));
    y2:=round(y1*cos(a)-z1*sin(a));
    y1:=y2;z1:=z2;
    x:=x1;y:=y1;z:=z1;
end;
procedure dorot(j:longint);
begin
    with kl[j] do
        begin
            for i:=1 to n do
                begin
pre(xyz[p[i]].x,xyz[p[i]].y,xyz[p[i]].z,x1,y1,z1);
                    if x1<ox1 then ox1:=x1;
                    if x1>ox2 then ox2:=x1;
                    if y1<oy1 then oy1:=y1;
                    if y1>oy2 then oy2:=y1;
                    tr[i].x:=sx1+round(x1*mn);
                    tr[i].y:=sy1+round(y1*mn);
                end;
                    setcolor(color);setfillstyle(1,color);
                    fillpoly(n,tr);
            end;
        end;
end;
procedure keyb;
begin
    while keypressed do ch:=readkey;
        repeat
            ch:=readkey;if ch=#0 then ch:=readkey;
        until ord(ch) in
[43,45,52,75,54,77,56,72,50,80,55,57,27];

```

```

omn:=mn;
case ord(ch) of
  52,75:b:=b+grad;
  54,77:b:=b-grad;
  56,72:a:=a+grad;
  50,80:a:=a-grad;
  55:c:=c-grad;
  57:c:=c+grad;
  43:if mn<100 then mn:=mn*1.1;
  45:if mn>0.1 then mn:=mn*0.9;
end;
end;
procedure initgr;
var grDriver,grMode,ErrCode:integer;
begin
  grDriver:=vga;grMode:=vgamed;
  InitGraph(grDriver,grMode,' ');
  ErrCode := GraphResult;
  if ErrCode<>grOk then
    Writeln('Graphics error:', GraphErrorMsg(ErrCode));
end;
function np:byte;
begin
  if page=0 then np:=1 else np:=0;
end;
procedure oldclear;
var t:real;
    x1,y1,x2,y2:longint;
begin
  t:=omn*1.5;
  x1:=sx1+round(ox1*t);
  y1:=sy1+round(oy1*t);
  x2:=sx1+round(ox2*t);
  y2:=sy1+round(oy2*t);
  if x1<0 then x1:=0;
  if y1<0 then y1:=0;
  if x2<0 then x2:=0;
  if y2<0 then y2:=0;
  if x1>getmaxx then x1:=640;

```

```

if x2>getmaxx then x2:=640;
if y1>getmaxy then y1:=400;
if y2>getmaxy then y2:=400;
setcolor(0);setfillstyle(1,0);
bar(x1,y1,x2,y2);
ox1:=0;oy1:=0;ox2:=0;oy2:=0;
end;
procedure show;
begin
  setactivepage(np);
  oldclear;
  for i:=1 to 10 do
  with so[i] do
  begin
    num:=i;
    pre(k1[i].x,k1[i].y,k1[i].z,x1,y1,z1);
    r:=sqr(x1)+sqr(y1)+sqr(z1-100000);
  end;
  if so[1].r>so[2].r then so[1]:=so[2] else
so[1]:=so[1];
  if so[3].r>so[4].r then so[2]:=so[4] else
so[2]:=so[3];
  if so[9].r>so[10].r then so[3]:=so[9] else
so[3]:=so[10];
  if so[5].r>so[6].r then
  begin l:=6; so[4]:=so[6]; so[5]:=so[7];
so[6]:=so[8]; end
  else begin l:=4; so[4]:=so[5]; end;
  for i:=1 to l-1 do
  for j:=1 to l-1 do
  if so[j].r<so[j+1].r then
  begin
    k:=so[j].r; so[j].r:=so[j+1].r; so[j+1].r:=k;
    k:=so[j].num; so[j].num:=so[j+1].num;
so[j+1].num:=k;
  end;
  for k:=1 to l do dorot(so[k].num);
  page:=np;setvisualpage(page);
end;
end;

```

```

procedure demo;
begin
  sx1:=-40;sy1:=200;a:=0;b:=0;c:=0;mn:=1;page:=0;omn:=1;
  ch:=#0;ox1:=0;oy1:=0;ox2:=0;oy2:=0;omn:=1;
  while sx1<680 do
  begin
    show; inc(sx1,shag);
    if keypressed then exit;
    b:=b-grad;
    delay(pa);
  end;
  sx1:=300;b:=0;sy1:=-40;
  while sy1<440 do
  begin
    show;inc(sy1,shag);
    if keypressed then exit;
    a:=a-grad;
    delay(pa);
  end;
  sy1:=-40;
  while sy1<440 do
  begin
    show;
    inc(sy1,shag);sx1:=round(sy1*4/3);
    if keypressed then exit;
    c:=c-grad;
    delay(pa);
  end;
  c:=0;sx1:=300;sy1:=200;
  mn:=20;omn:=20;
  for h:=1 to 800 do
  begin
    if keypressed then exit;
    mn:=mn*0.995;
    a:=a-grad*2.5;
    b:=b-grad*1.5;
    show;
    omn:=mn;
    delay(h div 20);
  end;
end;

```

```

end;
end;
begin
  initgr;
  demo;
  setactivepage(0);cleardevice;
  setactivepage(1);cleardevice;
  sx1:=300;sy1:=200;a:=0;b:=0;c:=0;mn:=1;page:=0;omn:=1;
  ch:=#0;ox1:=0;oy1:=0;ox2:=0;oy2:=0;omn:=1;
  while ch<>#27 do begin show; keyb; end;
end.

```

## Лабораторная работа 5. Растровая развертка кривой

Нарисовать растровую развертку:

1. Эллипса.
2. Отрезка толщиной два пикселя.
3. Гиперболы.
4. Параболы.
5. Треугольника.
6. Четырехугольника.
7. Пятиугольника.
8. Эллипса толщиной два пикселя.
9. Отрезка толщиной два пикселя.
10. Гиперболы толщиной два пикселя.
11. Параболы толщиной два пикселя.
12. Треугольника толщиной два пикселя.
13. Четырехугольника толщиной два пикселя.
14. Пятиугольника толщиной два пикселя.
15. Эллипса толщиной три пикселя.
16. Отрезка толщиной три пикселя.
17. Гиперболы толщиной четыре пикселя.
18. Параболы толщиной четыре пикселя.
19. Треугольника толщиной три пикселя.
20. Четырехугольника толщиной три пикселя.
21. Пятиугольника толщиной три пикселя.
22. Шестиугольника толщиной три пикселя.
23. Шестиугольника толщиной два пикселя.
24. Семиугольника толщиной три пикселя.



25. Семиугольника толщиной два пикселя.
26. Шестиугольника.
27. Восьмиугольника толщиной три пикселя.
28. Девятиугольника.
29. Восьмиугольника толщиной два пикселя.
30. Девятиугольника толщиной два пикселя.

**Пример.** Нарисовать растровую развертку окружности

```
uses crt, graph;
```

```
var xc, yc, gd, gm: integer;
```

```
procedure circlepoints(x, y, color: integer);
```

```
begin
```

```
  putpixel(xc+x, yc+y, color);
```

```
  putpixel(xc+y, yc+x, color);
```

```
  putpixel(xc+y, yc-x, color);
```

```
  putpixel(xc+x, yc-y, color);
```

```
  putpixel(xc-x, yc-y, color);
```

```
  putpixel(xc-y, yc-x, color);
```

```
  putpixel(xc-y, yc+x, color);
```

```
  putpixel(xc-x, yc+y, color);
```

```
end;
```

```
procedure MyCircle(xCenter, yCenter, r, color: integer);
```

```
var x, y, d, delta1, delta2: integer;
```

```
begin
```

```
  x:=0; y:=r;
```

```
  d:=1-r;
```

```
  delta1:=3;
```

```
  delta2:=-2*r+5;
```

```
  xc:=xcenter;
```

```
  yc:=ycenter;
```

```
  circlePoints(x, y, color);
```

```
  while y>x do
```

```
  begin
```

```
    if d<0 then
```

```
    begin
```

```
      d:=d+delta1;
```

```
      delta1:=delta1+2;
```

```

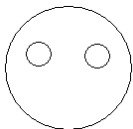
        delta2:=delta2+2;
        x:=x+1;
    end
    else
    begin
        d:=d+delta2;
        delta1:=delta1+2;
        delta2:=delta2+4;
        x:=x+1;
        y:=y-1;
    end;
    circlepoints(x,y,color);
end;
begin
    gd:=detect;
    initgraph(gd,gm,'');
    mycircle(150,100,50,11);
end.

```

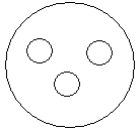
### Лабораторная работа 6. Заполнение областей

**Заполните внутреннюю часть фигуры:**

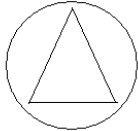
1. Эллипса.
2. Десятиугольника.
3. Окружности.
4. Одиннадцатиугольника.
5. Двух пересекающихся треугольников.
6. Четырехугольника.
7. Пятиугольника.
8. Эллипса толщиной два пикселя.
- 9.



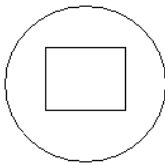
10.



11.



12.



13. Треугольника толщиной два пикселя.

14. Четырехугольника толщиной два пикселя.

15. Пятиугольника толщиной два пикселя.

16. Эллипса толщиной три пикселя.

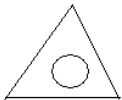
17.



18.



19.



20. Треугольника, внутри которого размещен квадрат.

21. Четырехугольника, внутри которого размещен круг.

22. Пятиугольника внутри которого размещен круг.

23. Шестиугольника, внутри которого размещен квадрат.
24. Шестиугольника, внутри которого размещен круг.
25. Семиугольника, внутри которого размещен квадрат.
26. Семиугольника, внутри которого размещен круг.
27. Шестиугольника, внутри которого размещен треугольник.
28. Восьмиугольника, внутри которого размещен треугольник.
29. Восьмиугольника, внутри которого размещен круг.
30. Восьмиугольника, внутри которого размещен квадрат.

**Пример.** Закрашивание внутренней части треугольника

```

program myfill1;
uses graph,crt;
type pstek =^stek;
   stek=record   {описание элемента стека}
   x,y:integer;  {координаты точки}
   link:pstek;  {указатель на предыдущий элемент
стека}
   end;
var
   beg:pstek;      {начало стека}
   fillcolor:word; { цвет закрашки}
   framcolor:word; { цвет границы}

procedure push(x,y:integer; var home:pstek);
var enter:pstek; {указатель на новый элемент
стека}
begin
{ if home=nil then
   begin
       new(home);
       home^.link:=nil;
       home^.x:=x;
       home^.y:=y;
   end
   else
begin
   new(enter);
   enter^.link:=home;

```

```

        enter^.x:=x;
        enter^.y:=y;
        home:=enter;
    end;}
    new(enter);
    enter^.link:=home;
    enter^.x:=x;
    enter^.y:=y;
    home:=enter;
end;
procedure pop(var x,y:integer;var home:pstek);
var enter:pstek;
begin
    if home=nil then {если в стеке нет элементов,
    to}
        begin          {выходим из программы}
            writeln('ошибка:стек уже пуст ');
            halt(1);
        end
        else
        begin
            enter:=home;
            x:=home^.x;{забираем координаты точки из
стека}
            y:=home^.y;
            home:=home^.link;
            dispose(enter);{освобождаем память}
        end;
    end;
end;
procedure
fillmy(x,y:integer;newcolor,framcolor:word);
begin
    beg:=nil;          {указатель на стек пуст}
    PutPixel (X, Y, newcolor);
    push(x,y,beg); {помещаем пиксел в Stack;}
    while (beg<>nil) do {пока стек не пуст}
    begin
        pop(x,y,beg);
        if not((GetPixel(X+1,Y) = newcolor) or

```

```

        (GetPixel(X+1,Y) = framcolor)
then
    begin
        PutPixel (X+1, Y, newcolor);
        push(X+1,Y,beg);
        end;
    if not((GetPixel(X,Y+1) = newcolor) or
        (GetPixel(X,Y+1) = framcolor))
        then
            begin
                PutPixel (X, Y+1, newcolor);
                push(X,Y+1,beg);
            end;
if not((GetPixel(X-1,Y) = newcolor) or
        (GetPixel(X-1,Y) =
framcolor))
        then
            begin
                PutPixel (X-1, Y, newcolor);
                push(X-1,Y,beg);
            end;
    if not((GetPixel(X,Y-1)=newcolor) or
        (GetPixel(X,Y-1) = framcolor))
        then
            begin
                PutPixel (X, Y-1, newcolor);
                push(X,Y-1,beg);
            end;
    end
end;
procedure
Triagle(x1,y1,x2,y2,x3,y3,fillColor,framColor:in
teger);
begin
    setColor(framColor); {устанавливаем цвет гра-
ницы}
    Line(x1,y1,x2,y2);    {рисует треугольник}
    Line(x2,y2,x3,y3);
    Line(x3,y3,x1,y1);

```

```

    {закрашиваем треугольник за начальную точку
закраски берем средне арифметическую точку от
трех граничных точек треугольника}
fillmy((x1+x2+x3)div 3, (y1+y2+y3)div 3, fillColor,
framColor);
end;
procedure MyInitGraph;
var gd, gm, er: integer;
begin
    gd:=DETECT;
    InitGraph(gd, gm,
'd:\turbo\bgi'); {инициализируем графический ре-
жим}
    if (GraphResult <> grOk) then begin
        writeln('Ошибка:', GraphErrorMsg(Er));
        halt(1)
    end;
end;
begin
    myInitGraph; {инициализируем графический ре-
жим}
    fillColor:=green; {цвет закраски}
    framColor:=red; {цвет границы}
    Triangle(100,100,150,70,120,170,fillColor,
framColor);
    readln;
    CloseGraph; {закрываем графический режим}
end.

```

### **Список рекомендованной литературы**

1. Петров М.Н., Молочков В.П. Компьютерная графика. СПб. Питер, 2003. 736 с.
2. Шикин Е.В., Боресков А.В. Компьютерная графика. Полигональные модели. М.: ДИАЛОГ-МИФИ, 2001. 464 с.
3. Глушаков С.В., Кнабе Г.А. Компьютерная графика. Харьков: Фолио, 2002. 500 с.
4. Б. Мандельброт. Фрактальная геометрия природы. М.: Институт компьютерных исследований, 2002. 656 с.

5. Ласло М. Вычислительная геометрия и компьютерная графика на C++. М, 1997. 396 с.
6. Роджерс Д.Ф. Алгоритмические основы машинной графики. М, 1989. 456 с.
7. Артемьев И.Т., Ильина Л.А., Ильин Д.В. Программирование на алгоритмических языках: лабораторный практикум. Чебоксары: Изд-во Чуваш. ун.-та, 2005. 100 с.
8. Шикин Е.В., Боресков А.В., Зайцев А.А. Начала компьютерной графики. М.: ДИАЛОГ-МИФИ, 1993. 138 с.
9. Артемьев И.Т., Новикова С. В. Программирование на языке Turbo Pascal: лабораторный практикум. Чебоксары: Изд-во Чуваш. ун.-та, 2000. 160с.



## Оглавление

Лабораторная работа 1. Инструменты выделения и их использование для создания коллажа.....	1
Лабораторная работа 2. Инструменты ретуширования и тонирования. Ретушь черно-белой фотографии.....	<b>Ошибка! Закладка не определена.</b>
Лабораторная работа 3. Цветная ретушь изображений (коррекция цвета, устранение эффекта красных глаз). ..	<b>Ошибка! Закладка не определена.</b>
Лабораторная работа 4. Работа со слоями ..	<b>Ошибка! Закладка не определена.</b>
Лабораторная работа 5. Создание текстовых эффектов с использованием фильтров ..	<b>Ошибка! Закладка не определена.</b>
Лабораторная работа 6. Использование текстур для создания фона страницы .....	<b>Ошибка! Закладка не определена.</b>
Лабораторная работа 7. Добавление к изображению природных явлений .....	<b>Ошибка! Закладка не определена.</b>
Лабораторная работа 8. Создание художественной рамки .....	<b>Ошибка! Закладка не определена.</b>
Лабораторная работа 9. Раскраска черно-белой фотографии .....	<b>Ошибка! Закладка не определена.</b>
Лабораторная работа 10. Фотомонтаж ...	<b>Ошибка! Закладка не определена.</b>
Лабораторная работа 11. Создание графических изображений с использованием графические операторов языков программирования.....	2
Лабораторная работа 12. Создание движущихся графических изображений.....	4
Лабораторная работа 13. Использование функций сохранения и выдачи изображений .....	8
Лабораторная работа 14. Создание движущихся объемных изображений с использованием матриц преобразования в пространстве .....	10

Лабораторная работа 15. Растровая развертка кривой.....	16
Лабораторная работа 16. Заполнение областей.....	18
Список рекомендованной литературы .....	23

КОМПЬЮТЕРНАЯ ГРАФИКА

Лабораторный практикум

Редактор

Компьютерная верстка правка

Подписано в печать      Формат 60×84/16. Бумага газетная.

Гарнитура Times. Печать офсетная.

Усл. печ.л.    Уч.-изд. л.    Тираж    экз. Заказ №