

Графика в Pascal

Prezentatii.com

Графические операторы в Pascal'e

- ✓ Задание графического режима
- ✓ Линия
- ✓ Прямоугольник
- ✓ Окружность
- ✓ Эллипс
- ✓ Дуга

выход

содержание

Тест №1

Тест №2

Графический режим

Для вывода графических изображений на экран монитора в языке Pascal существует стандартная библиотека Graph, подключение которой осуществляется стандартным способом

uses Graph;

В графическом режиме работы любое изображение на экране дисплея состоит из множества мельчайших светящихся точек – пикселей. Экран дисплея представляет собой прямоугольную сетку координат.



Графический режим

$(0,0)$

$(640,0)$ **x**

$(0,480)$

y

Размер экрана 640x480

Инициализация графического режима

```
Program pr;  
Uses crt, graph;    {использование графического режима}  
Var driver,         {графический драйвер}  
    mode:integer;    {графический режим}  
Begin  
    Driver:=detect;    {автораспознавание драйвера}  
    Initgraph(driver, mode,'путь к файлу egavga.bgi');  
    {операторы и процедуры графических построений}  
    Readkey;  
    Closegraph;  
End.
```



Номера цветов

0 – черный

1 – синий

2 – зеленый

3 – голубой

4 – красный

5 – фиолетовый

6 – коричневый

7 – светло-серый

8 – темно-серый

9 – ярко-синий

10 – ярко-зелёный

11 – ярко-голубой

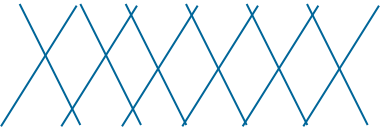
12 – ярко-красный

13 – ярко-розовый

14 – жёлтый

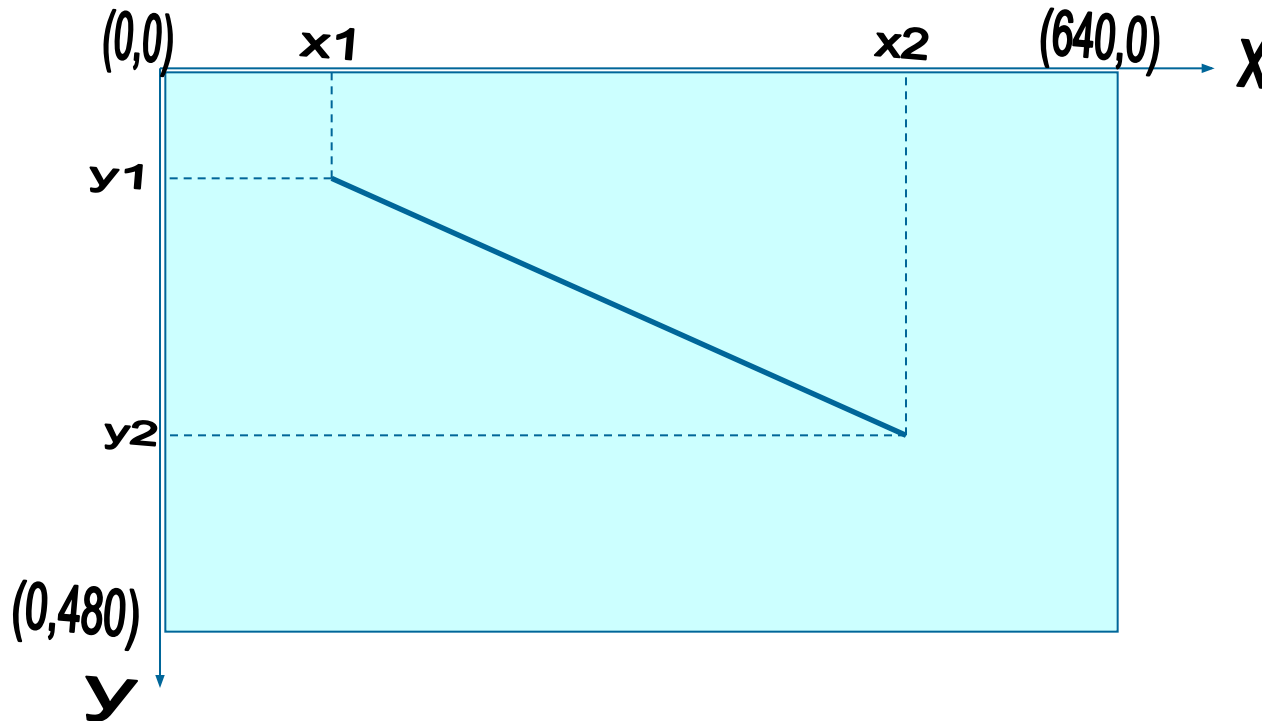
15 – белый



№ стиля	Вид стиля заливки	№ стиля	Вид стиля заливки
0	Заполнение (закрашивает цветом фона)	6	
1	Заполнение текущим цветом	7	
2		8	
3		9	
4		10	
5		11	



Линия



Оператор **LINE(X1,Y1,X2,Y2);** рисует на экране отрезок прямой из точки с координатами $(x1,y1)$ до точки с координатами $(x2,y2)$.



Установка цвета

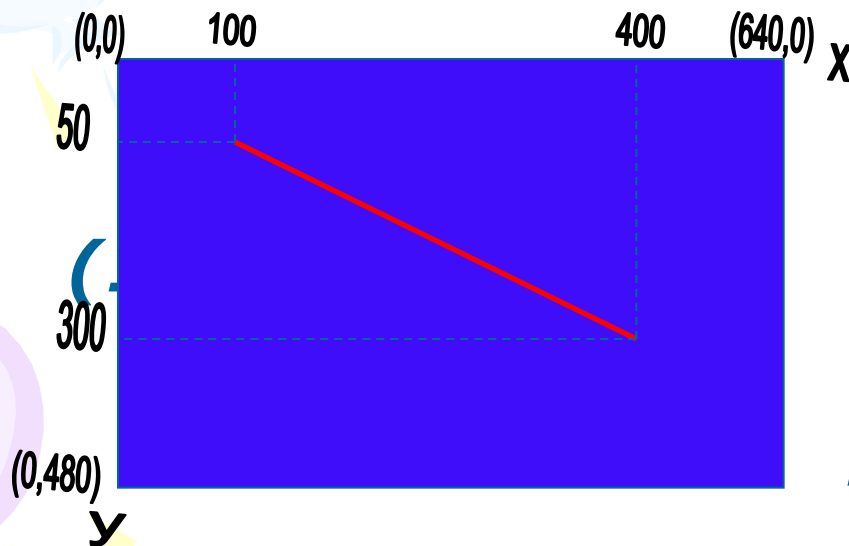
SetColor(Color:integer);

Устанавливает основной цвет рисования

SetBkColor(Color:integer);

Устанавливает текущий фоновый цвет

Например: нарисовать на экране синего цвета отрезок прямой из точки с координатами(100,50) в точку с координатами(400,300) красного цвета:

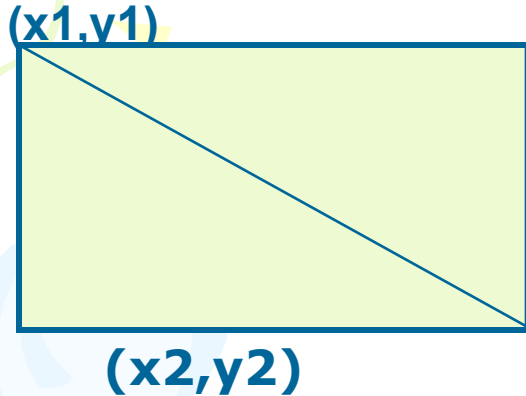


SetBkColor

SetColor(4);

LINE(100,50,400,300);

Прямоугольник



Процедура

Var(x1,y1,x2,y2)

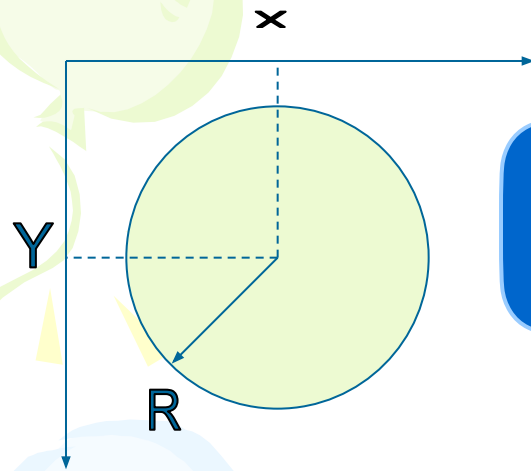
выводит на экран закрашенный
прямоугольник с координатами
левого верхнего угла (x1,y1) и
правого нижнего угла (x2,y2).

Чтобы начертить закрашенные фигуры нужно:

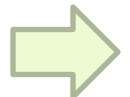
- Установить цвет фигуры:
SETCOLOR(№ цвета);
 - Установить цвет и стиль закрашки:
SETFILLSTYLE(№ стиля, № цвета);
 - Нарисовать фигуру: **{название фигуры}** (данные нужные для закрашки);
 - Закрасить фигуру: **FLOODFILL**(x,y,№ цвета границы);
- Где **x,y** – координаты точки в замкнутой области.



ОКРУЖНОСТЬ

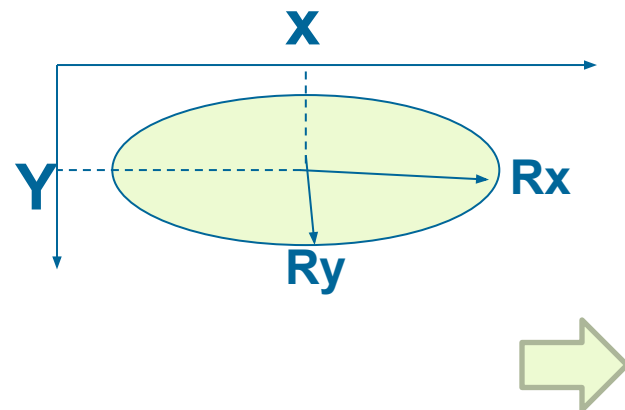
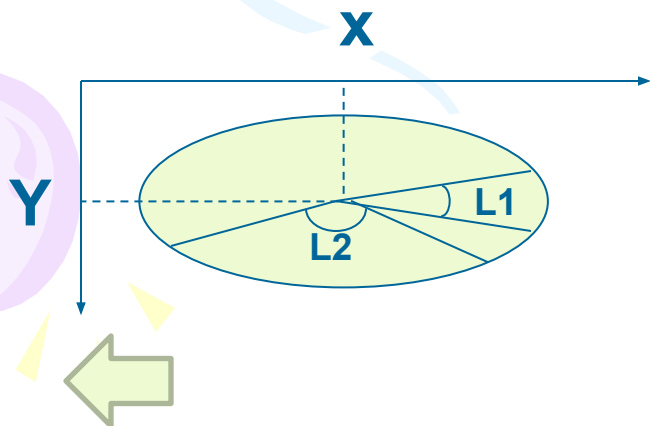


Процедура **CIRCLE(x,y,r)** выводит на экран окружность с центром (**x**, **y**) и радиусом **r**.

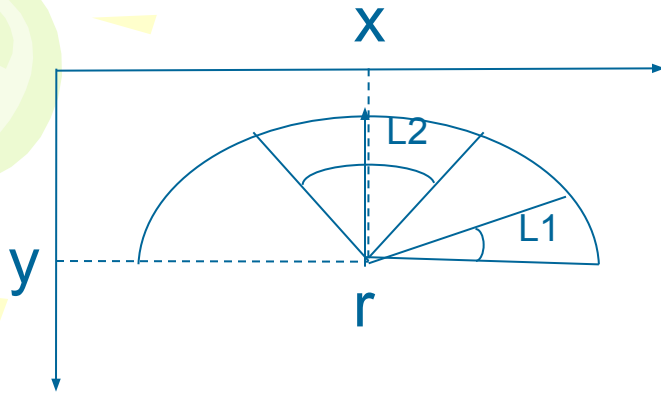


ЭЛЛИПС

Процедура **ELLIPSE**($x, y, L1, L2, Rx, Ry$) рисует эллиптическую дугу от начального угла **L1** до конечного **L2**, используя X, Y как координаты центра, как горизонтальный (**Rx**) и вертикальный (**Ry**) радиус.



ДУГА



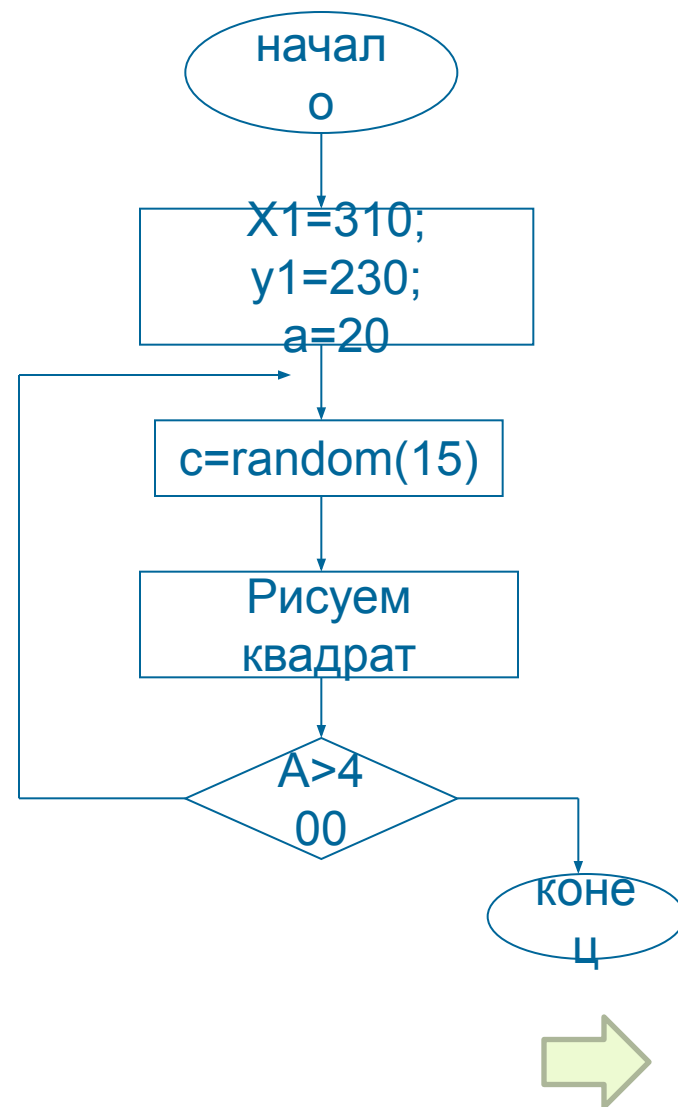
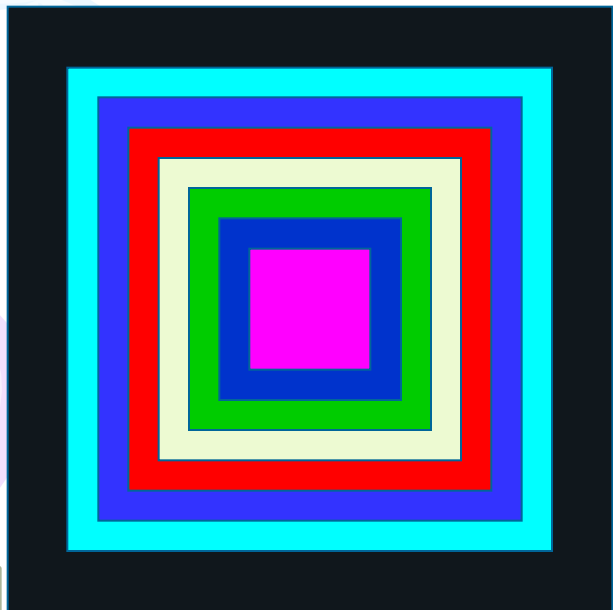
Процедура **Arc(x,y,L1,L2,r)**
рисует дугу с центром в точке (x,
y) от начального угла L1 до
конечного L2 радиусом r.



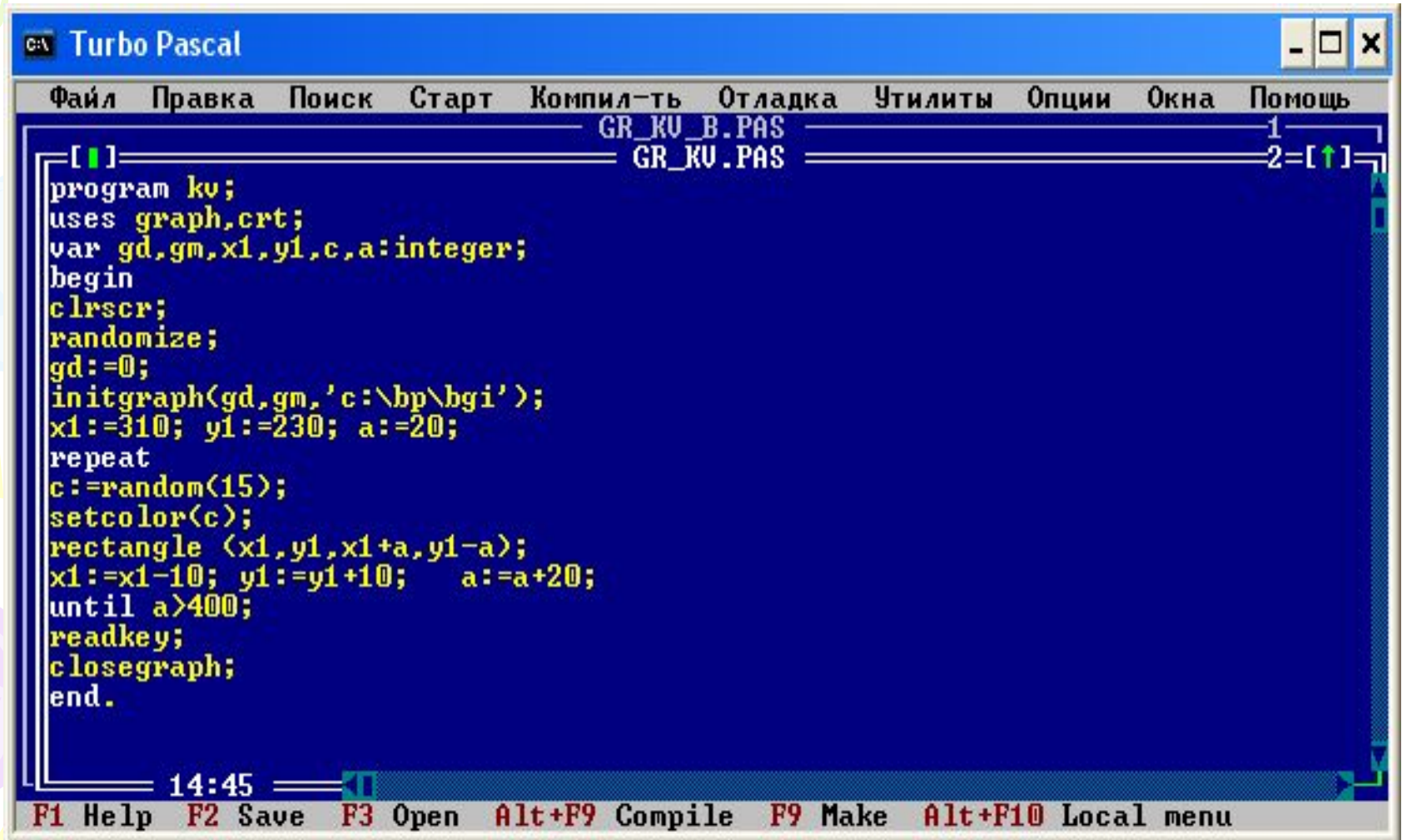
Примеры решения задач:

Забавные квадраты

- Нарисовать квадраты, стороны которых увеличиваются от 20 до 400 ppi, с шагом 20 ppi. Центр экрана совпадает с центром квадрата. Цвет квадратов задаётся случайным числом. Для решения задачи используется цикл с постусловием.



Текст программы



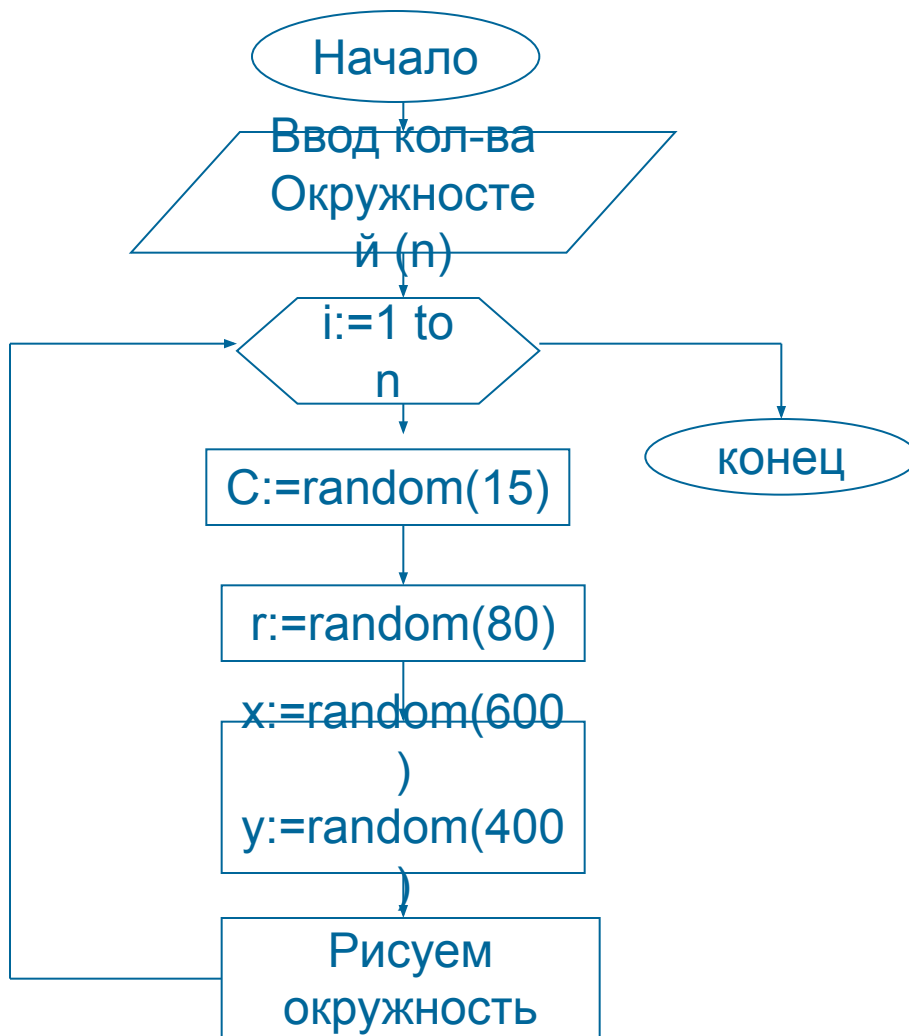
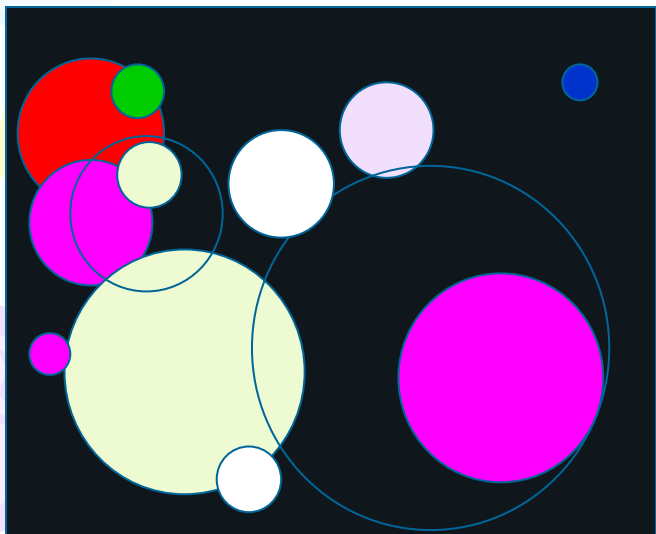
```
program kv;  
uses graph,crt;  
var gd,gm,x1,y1,c,a:integer;  
begin  
  clrscr;  
  randomize;  
  gd:=0;  
  initgraph(gd,gm,'c:\bp\bgi');  
  x1:=310; y1:=230; a:=20;  
  repeat  
    c:=random(15);  
    setcolor(c);  
    rectangle (x1,y1,x1+a,y1-a);  
    x1:=x1-10; y1:=y1+10; a:=a+20;  
  until a>400;  
  readkey;  
  closegraph;  
end.
```

14:45

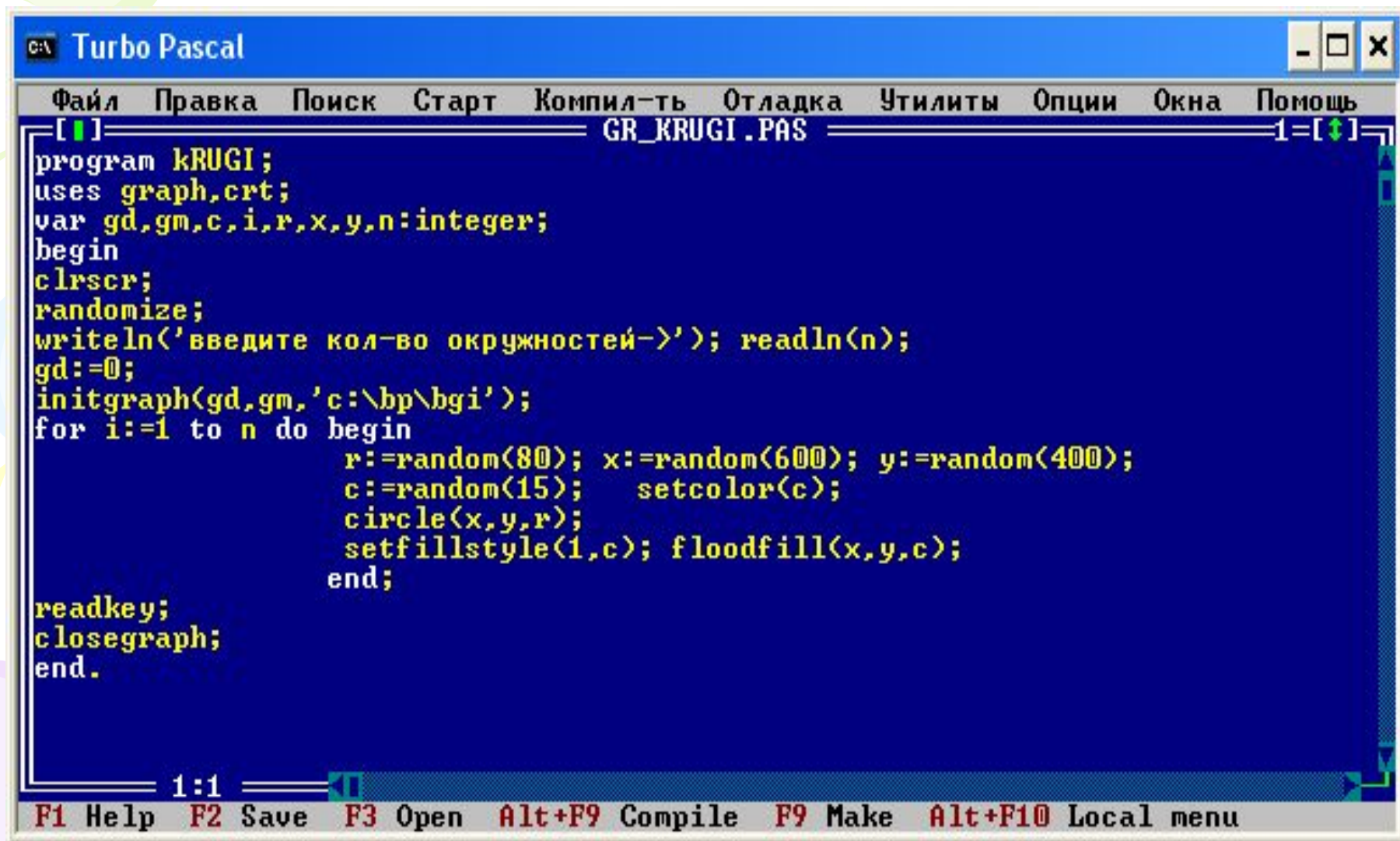
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu

Воздушные пузыри

- Нарисовать окружности, кол-во которых задается пользователем. Центр, радиус и цвет окружностей задаётся случайным числом.



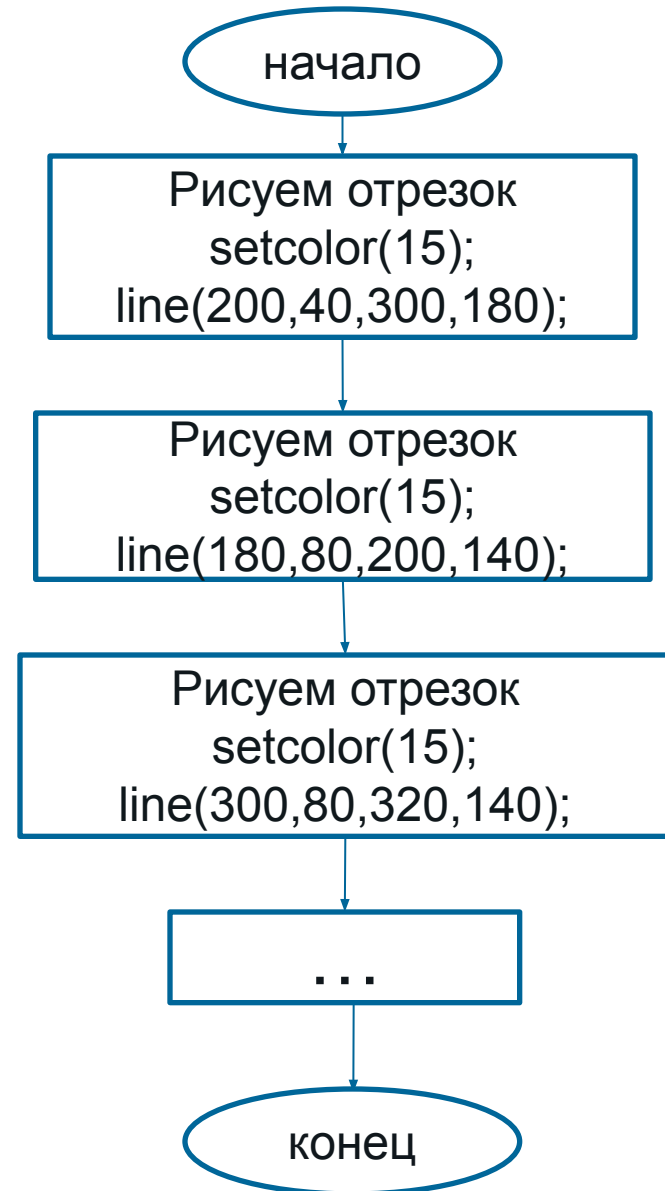
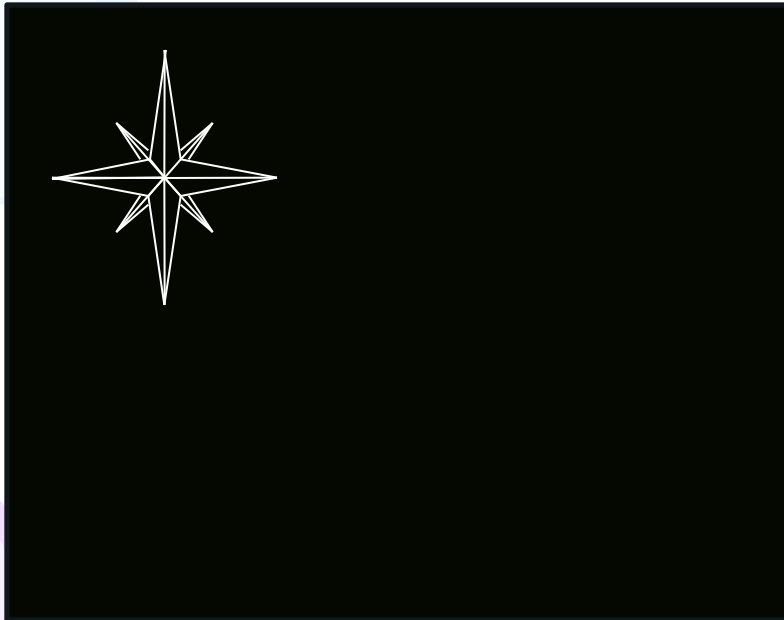
Текст программы



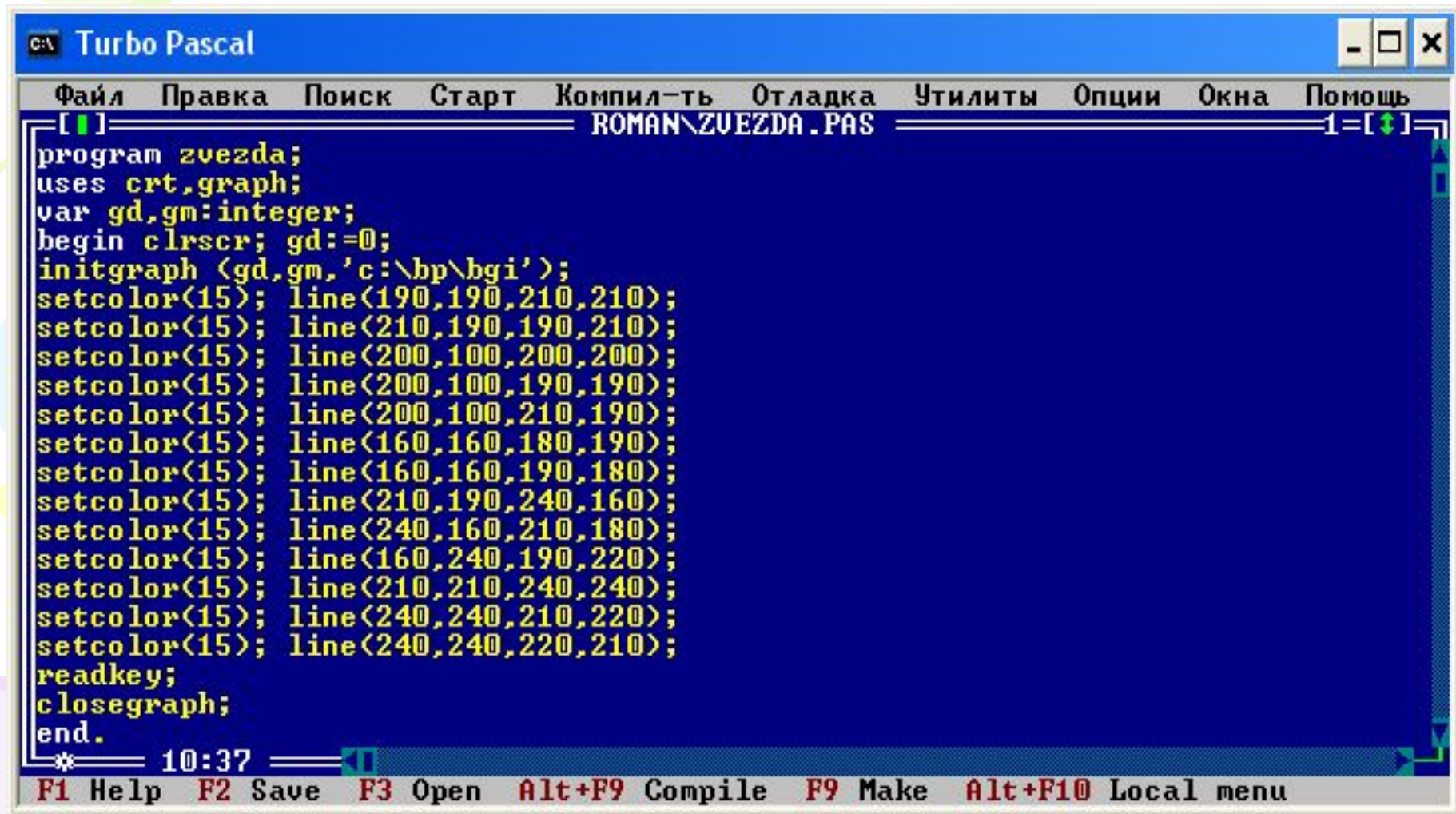
```
program kRUGI;  
uses graph,crt;  
var gd,gm,c,i,r,x,y,n:integer;  
begin  
  clrscr;  
  randomize;  
  writeln('введите кол-во окружностей->'); readln(n);  
  gd:=0;  
  initgraph(gd,gm,'c:\bp\bgi');  
  for i:=1 to n do begin  
    r:=random(80); x:=random(600); y:=random(400);  
    c:=random(15); setcolor(c);  
    circle(x,y,r);  
    setfillstyle(1,c); floodfill(x,y,c);  
  end;  
  
  readkey;  
  closegraph;  
end.
```

Звезда

- Нарисовать звезду прямыми линиями.



Текст программы

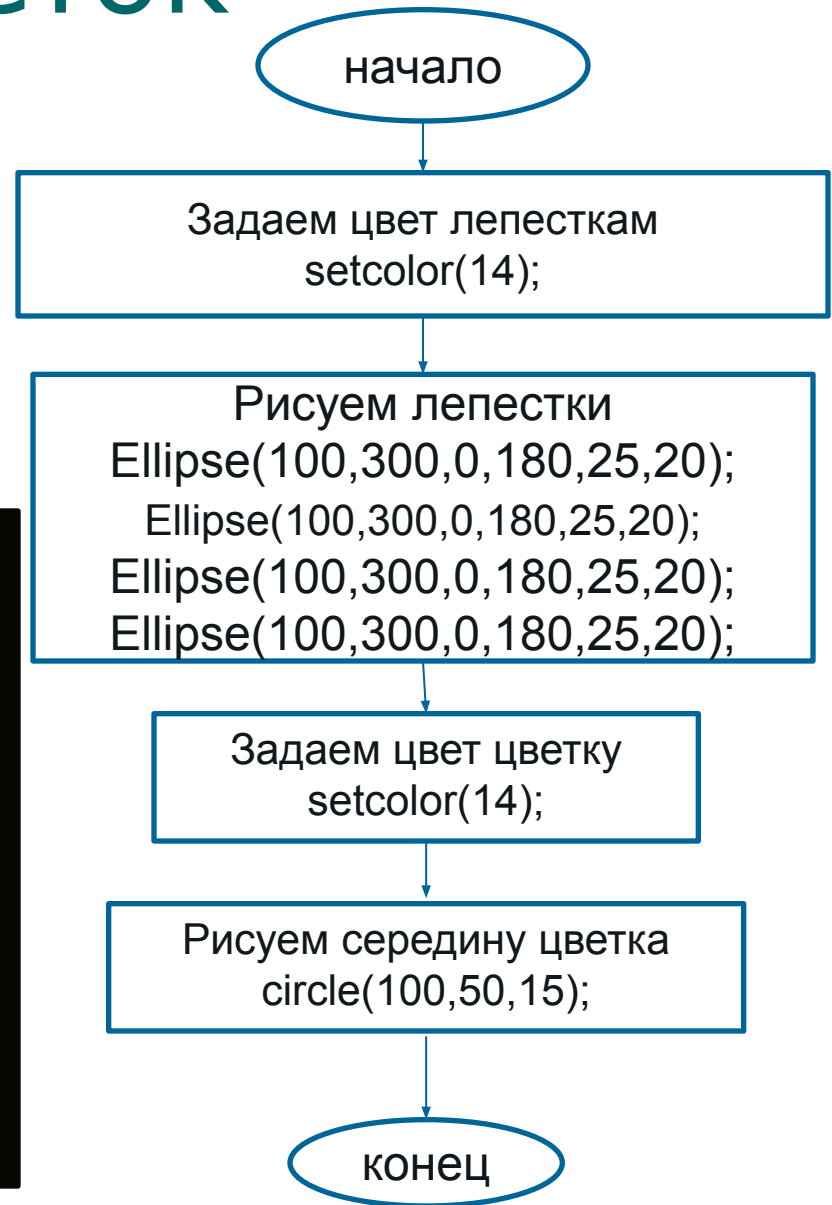
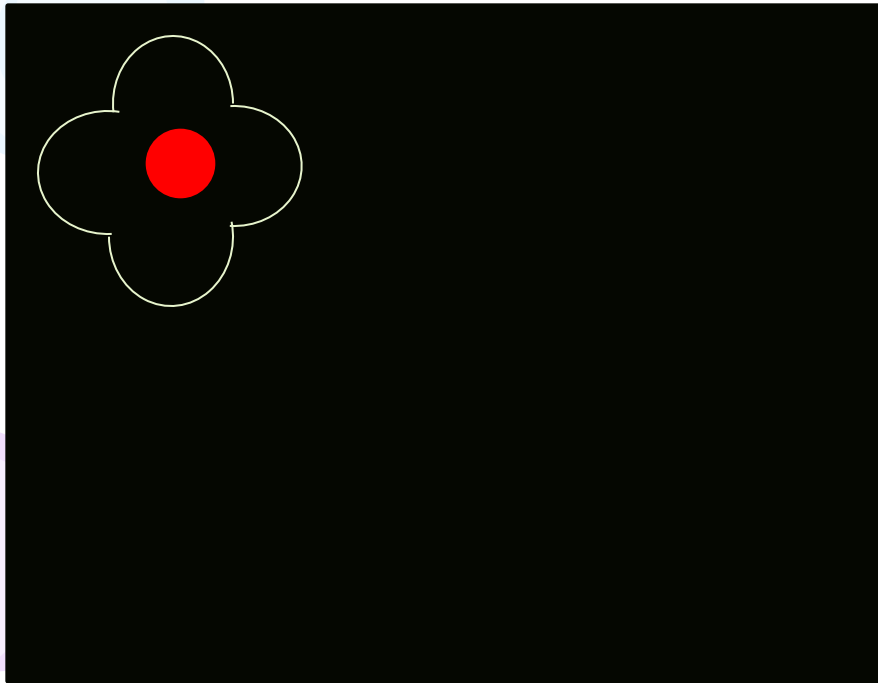


The screenshot shows the Turbo Pascal development environment. The title bar reads 'Turbo Pascal'. The menu bar includes 'Файл', 'Правка', 'Поиск', 'Старт', 'Компил-ть', 'Отладка', 'Утилиты', 'Опции', 'Окна', and 'Помощь'. The status bar at the bottom displays function key shortcuts: F1 Help, F2 Save, F3 Open, Alt+F9 Compile, F9 Make, Alt+F10 Local menu, and the time 10:37. The main window shows the source code for 'ROMAN\ZVEZDA.PAS'.

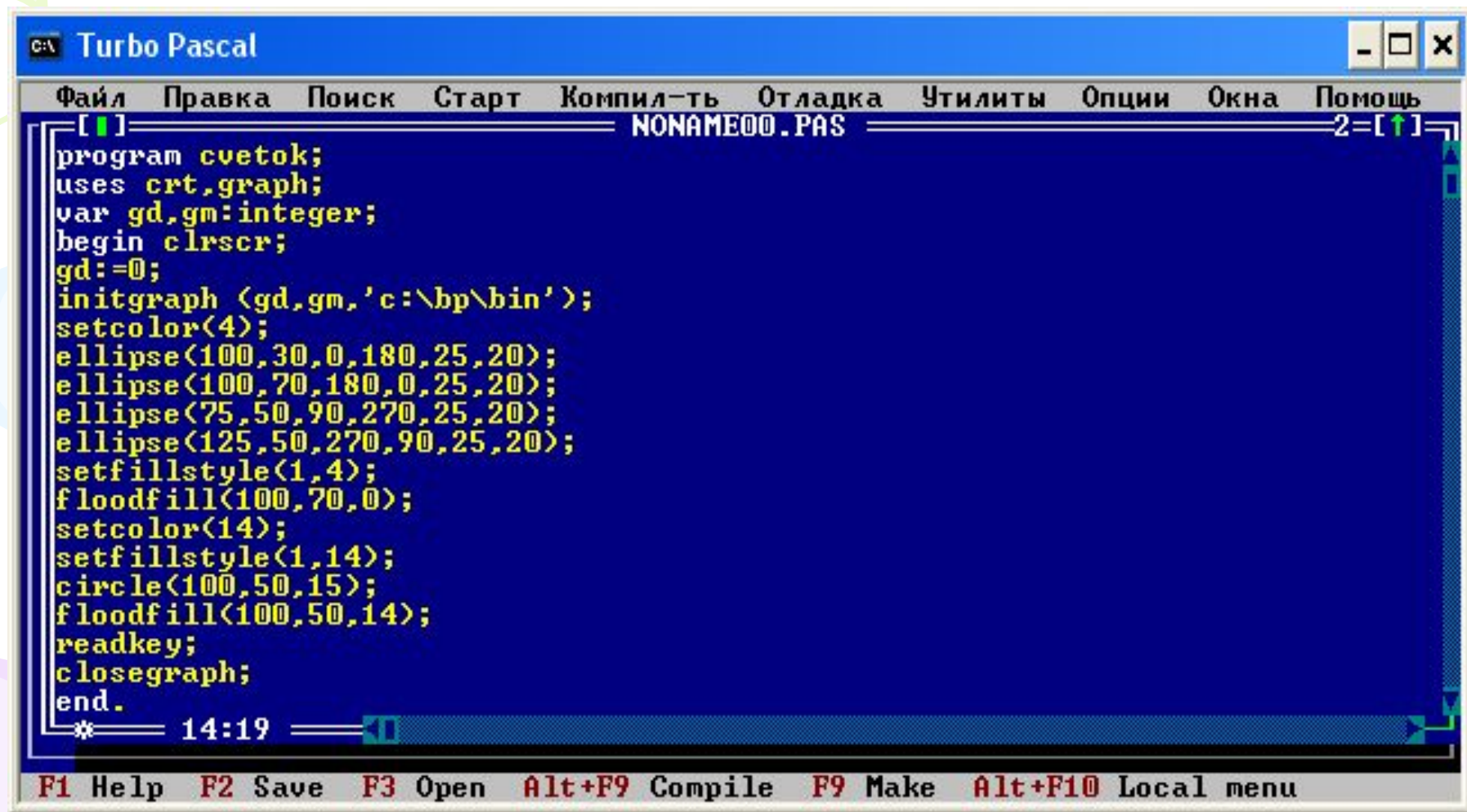
```
program zvezda;
uses crt, graph;
var gd, gm: integer;
begin clrscr; gd:=0;
initgraph (gd, gm, 'c:\bp\bgi');
setcolor(15); line(190,190,210,210);
setcolor(15); line(210,190,190,210);
setcolor(15); line(200,100,200,200);
setcolor(15); line(200,100,190,190);
setcolor(15); line(200,100,210,190);
setcolor(15); line(160,160,180,190);
setcolor(15); line(160,160,190,180);
setcolor(15); line(210,190,240,160);
setcolor(15); line(240,160,210,180);
setcolor(15); line(160,240,190,220);
setcolor(15); line(210,210,240,240);
setcolor(15); line(240,240,210,220);
setcolor(15); line(240,240,220,210);
readkey;
closegraph;
end.
```

Цветок

- Нарисовать цветок используя оператор **Ellipse**.



Тест программы



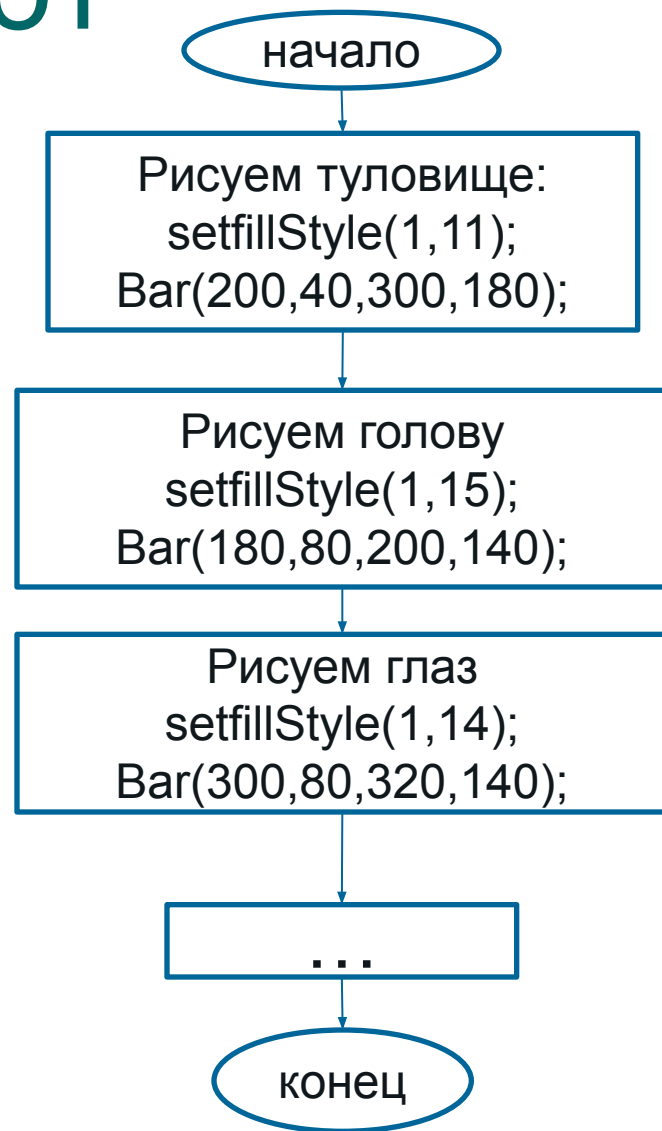
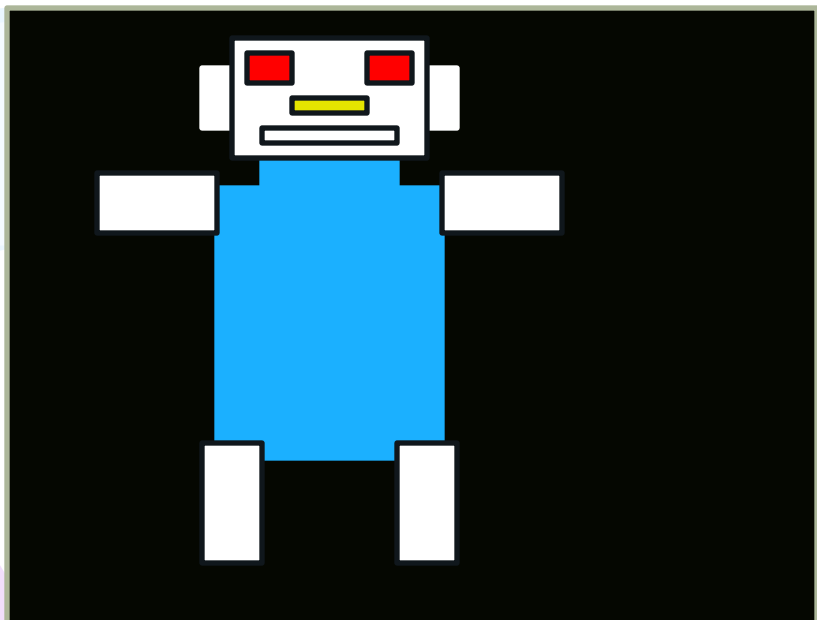
```
program cvetok;  
uses crt,graph;  
var gd,gm:integer;  
begin clrscr;  
gd:=0;  
initgraph (gd,gm,'c:\bp\bin');  
setcolor(4);  
ellipse(100,30,0,180,25,20);  
ellipse(100,70,180,0,25,20);  
ellipse(75,50,90,270,25,20);  
ellipse(125,50,270,90,25,20);  
setfillstyle(1,4);  
floodfill(100,70,0);  
setcolor(14);  
setfillstyle(1,14);  
circle(100,50,15);  
floodfill(100,50,14);  
readkey;  
closegraph;  
end.
```

* 14:19

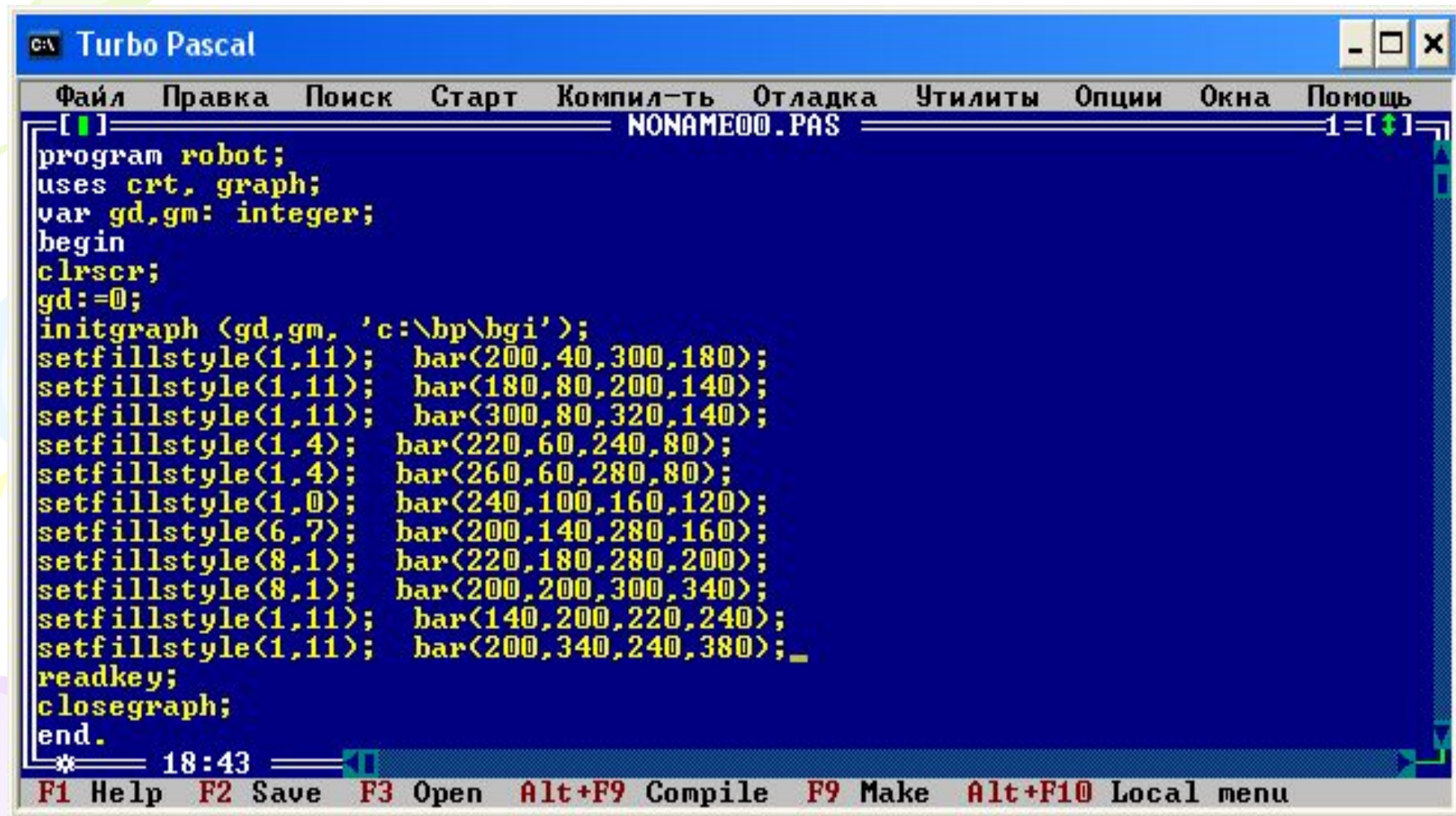
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu

Робот

- Нарисовать робота из разноцветных прямоугольников.



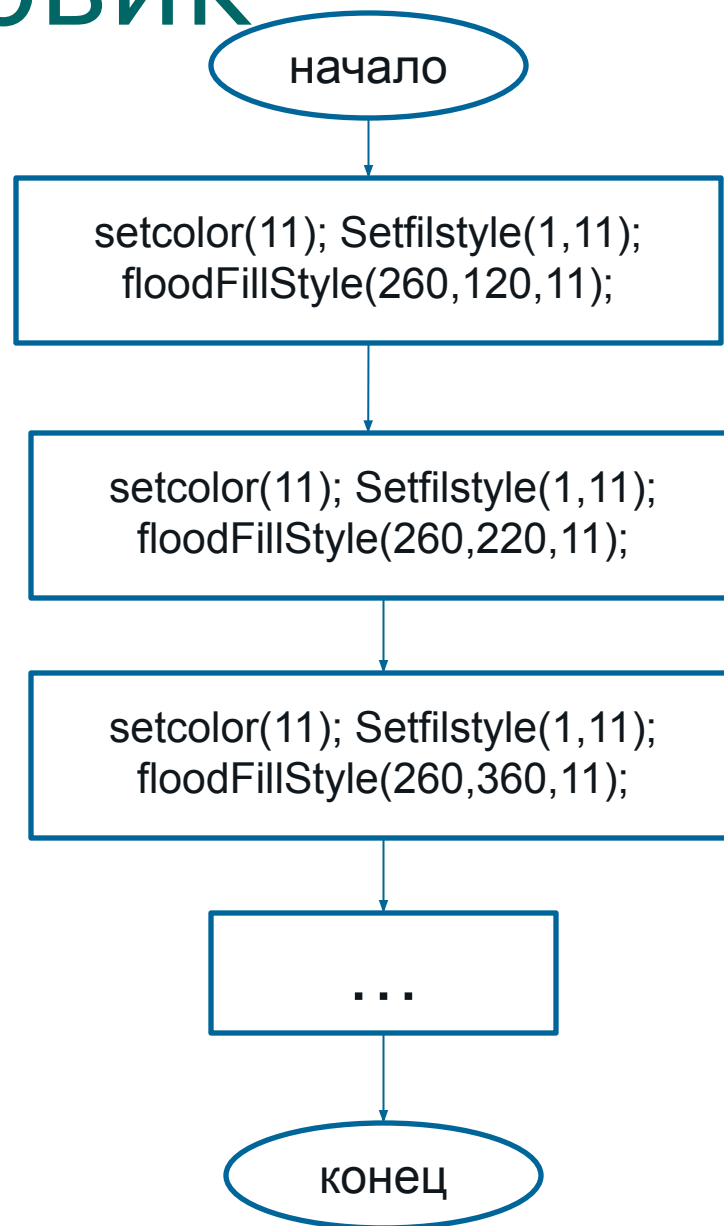
Текст программы



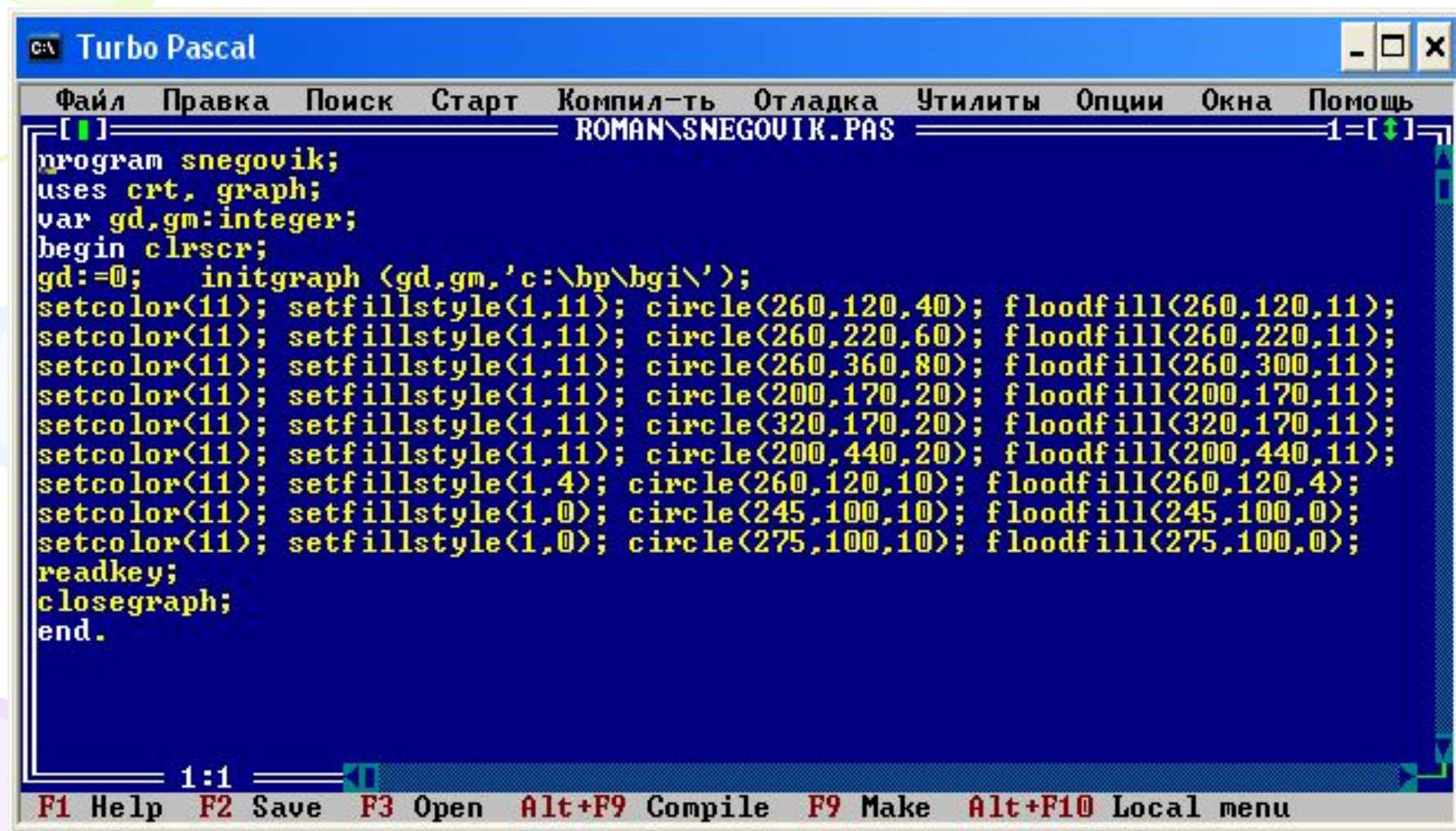
```
c:\ Turbo Pascal
Файл  Правка  Поиск  Старт  Компил-ть  Отладка  Утилиты  Опции  Окна  Помощь
[ ] NONAME00.PAS 1=[↑]
program robot;
uses crt, graph;
var gd,gm: integer;
begin
clrscr;
gd:=0;
initgraph (gd,gm, 'c:\bp\bgi');
setfillstyle(1,11);  bar(200,40,300,180);
setfillstyle(1,11);  bar(180,80,200,140);
setfillstyle(1,11);  bar(300,80,320,140);
setfillstyle(1,4);   bar(220,60,240,80);
setfillstyle(1,4);   bar(260,60,280,80);
setfillstyle(1,0);   bar(240,100,160,120);
setfillstyle(6,7);   bar(200,140,280,160);
setfillstyle(8,1);   bar(220,180,280,200);
setfillstyle(8,1);   bar(200,200,300,340);
setfillstyle(1,11);  bar(140,200,220,240);
setfillstyle(1,11);  bar(200,340,240,380);
readkey;
closegraph;
end.
* 18:43
F1 Help  F2 Save  F3 Open  Alt+F9 Compile  F9 Make  Alt+F10 Local menu
```

Снеговик

- Нарисовать снеговик одними разноцветными окружностями.



Текст программы



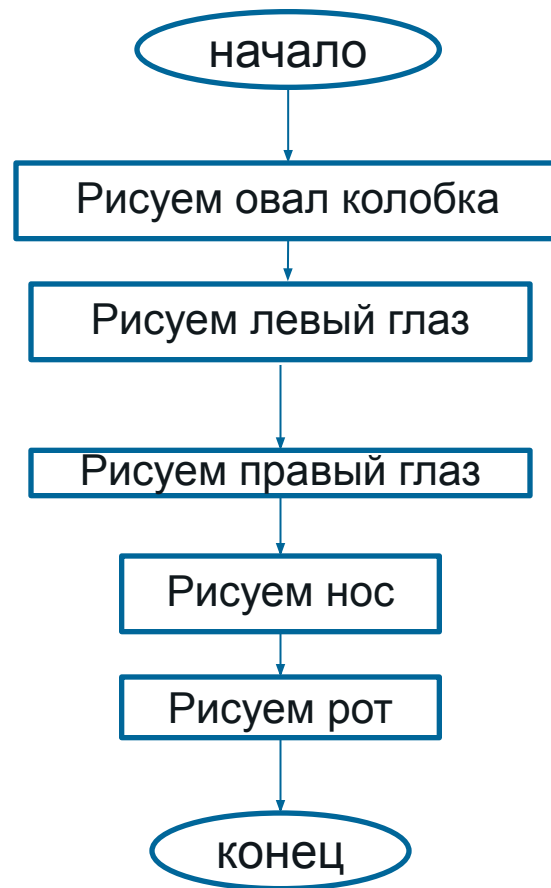
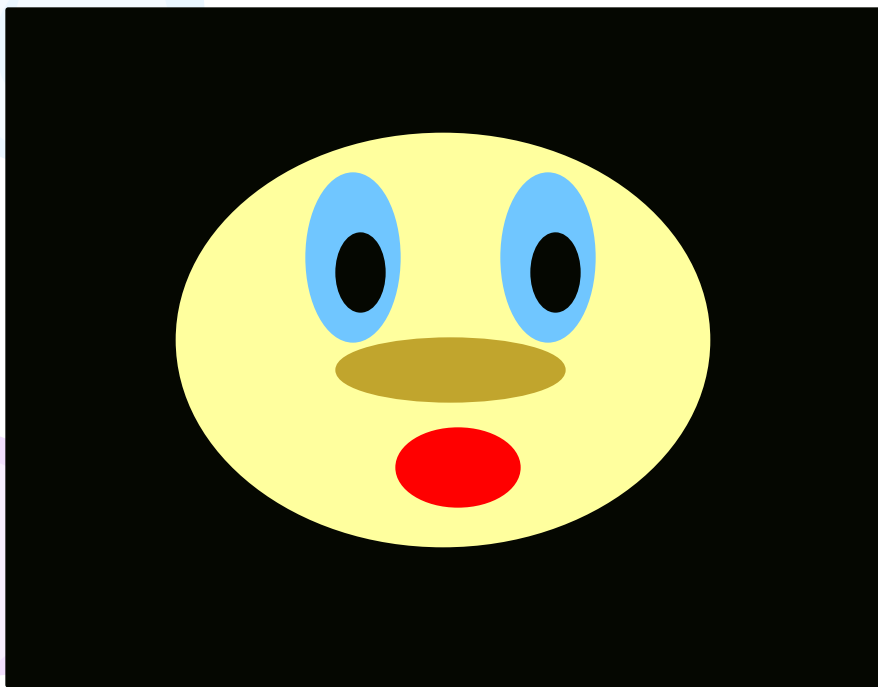
```
program snegovik;
uses crt, graph;
var gd, gm: integer;
begin clrscr;
gd:=0;  initgraph (gd, gm, 'c:\bp\bgi\');
setcolor(11); setfillstyle(1, 11); circle(260, 120, 40); floodfill(260, 120, 11);
setcolor(11); setfillstyle(1, 11); circle(260, 220, 60); floodfill(260, 220, 11);
setcolor(11); setfillstyle(1, 11); circle(260, 360, 80); floodfill(260, 360, 11);
setcolor(11); setfillstyle(1, 11); circle(200, 170, 20); floodfill(200, 170, 11);
setcolor(11); setfillstyle(1, 11); circle(320, 170, 20); floodfill(320, 170, 11);
setcolor(11); setfillstyle(1, 11); circle(200, 440, 20); floodfill(200, 440, 11);
setcolor(11); setfillstyle(1, 4); circle(260, 120, 10); floodfill(260, 120, 4);
setcolor(11); setfillstyle(1, 0); circle(245, 100, 10); floodfill(245, 100, 0);
setcolor(11); setfillstyle(1, 0); circle(275, 100, 10); floodfill(275, 100, 0);
readkey;
closegraph;
end.
```

1:1

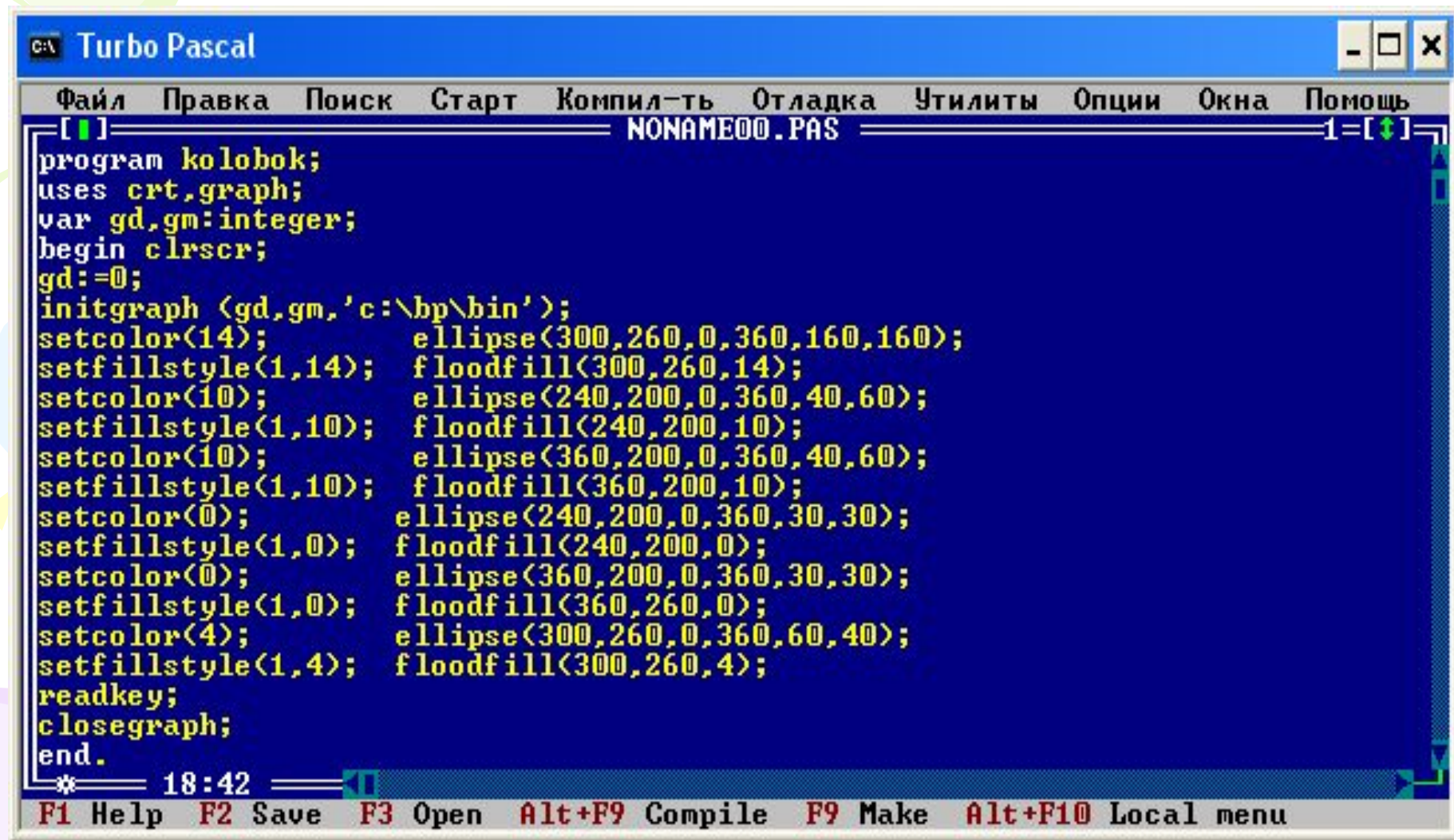
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu

Колобок

- Нарисовать колобка одними разноцветными эллипсами.



Текст программы



```
program kolobok;
uses crt,graph;
var gd,gm:integer;
begin clrscr;
gd:=0;
initgraph (gd,gm,'c:\bp\bin');
setcolor(14);      ellipse(300,260,0,360,160,160);
setfillstyle(1,14); floodfill(300,260,14);
setcolor(10);      ellipse(240,200,0,360,40,60);
setfillstyle(1,10); floodfill(240,200,10);
setcolor(10);      ellipse(360,200,0,360,40,60);
setfillstyle(1,10); floodfill(360,200,10);
setcolor(0);       ellipse(240,200,0,360,30,30);
setfillstyle(1,0);  floodfill(240,200,0);
setcolor(0);       ellipse(360,200,0,360,30,30);
setfillstyle(1,0);  floodfill(360,260,0);
setcolor(4);       ellipse(300,260,0,360,60,40);
setfillstyle(1,4);  floodfill(300,260,4);
readkey;
closegraph;
end.
```

* 18:42

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

Содержание

<u>1.Задание графического режима.....</u>	<u>3</u>
<u>2.Линия.....</u>	<u>8</u>
<u>3.Прямоугольник.....</u>	<u>10</u>
<u>4.Окружность.....</u>	<u>11</u>
<u>5.Эллипс.....</u>	<u>12</u>
<u>6. Дуга.....</u>	<u>13</u>
<u>7.Примеры решения задач.....</u>	<u>14</u>
• <u>Забавные квадраты.....</u>	<u>-</u>
• <u>Воздушные пузыри.....</u>	<u>16</u>
• <u>Звезда.....</u>	<u>18</u>
• <u>Цветок.....</u>	<u>20</u>
• <u>Робот.....</u>	<u>22</u>
• <u>Снеговик.....</u>	<u>24</u>
• <u>Колобок.....</u>	<u>26</u>

Тест №1

Тест №2