гр. София, пл. Славейков 11 ет.5 тел. : +359 897 91 93 96

office@progressbg.net www.progressbg.net

Java Graphics

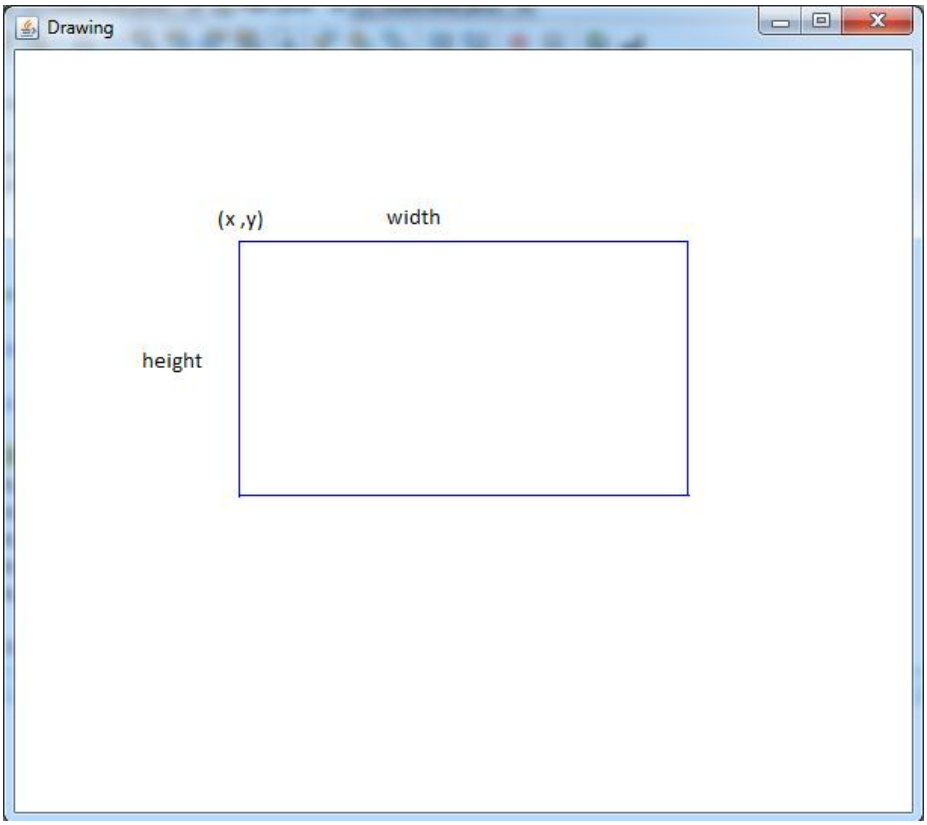
Запознаване с няколко нови метода на класа Graphics

- **drawRect**
- **drawArc**
- **drawPolygon**
- **drawstring**
- **fill методи**

g.drawRect (int *x* , int *y* , int *width* , int *height*)

Методът рисува правоъгълник като левият и десният ръб имат координати съответно *x* и *x* + *width*, а горният и долният ръб – *y* и *y* + *height*.

x , *y* – координатите на горният ляв връх на правоъгълника
width – широчината на правоъгълника
height– височината на правоъгълника

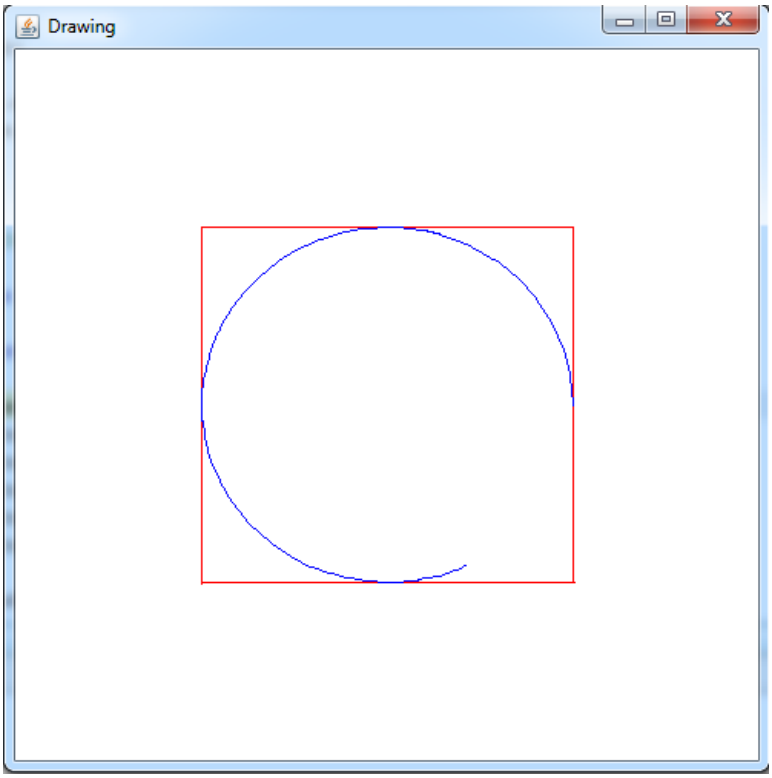



```
19      public void paintComponent(Graphics g) {
20          super.paintComponent(g);
21
22          g.drawRect (getWidth()/4 , getHeight()/4 , getWidth()/2 , getHeight()/2);
23          //           x           , y        ,width of the Rectangle , height of the Rectangle
24      }
    }
```

g.drawArc (int *x* , int *y* , int *width* , int *height* , int *startAngle* , int *arcAngle*)

Методът рисува дъга, която се допира до стените на правоъгълник, дефиниран с (*x* , *y* , *width* , *height*). Дъгата започва от *startAngle* градуса и се върти *arcAngle* градуса. Когато *startAngle* е 0 градуса това се интерпретира, като дъгата да започне от позицията на 3ч на часовника. Когато подадените градуси са положителни числа въртенето е по посока **ОБРАТНА** на часовниковата стрелка. Когато са отрицателни – по посока на часовниковата стрелка.

x , *y* – координатите на горният ляв връх на правоъгълника
width – широчината на правоъгълника
height– височината на правоъгълника
startAngle – ъгълът от който почва арката
arcAngle - ъгълът на дъгата

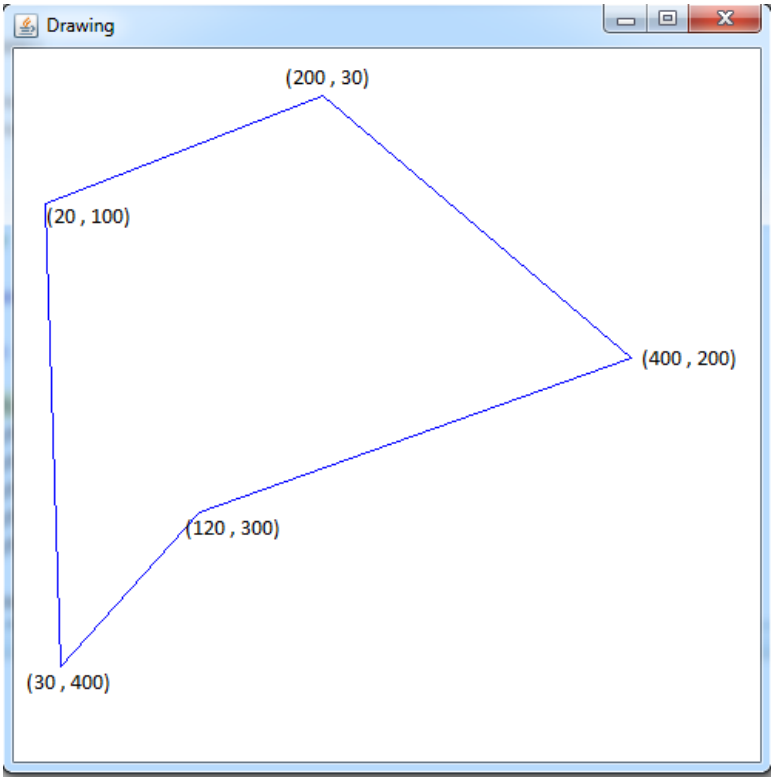


```
16  public class DrawPanel extends JPanel
17  {
18      public void paintComponent(Graphics g)
19      {
20          super.paintComponent(g);
21
22          g.setColor (Color.BLUE);
23          g.drawArc(getWidth()/4 , getHeight()/4 , getWidth()/2 , getHeight()/2 ,    0       ,    295);
24          //       |координатите на правоъгълника|        размертие му          |стартов ъгъл|ъгъл на дъгата
25      }
26  }
```

g.drawPolygon (int[] *xPoints* , int[] *yPoints* , int *nPoints*)

Методът рисува многоъгълник , на когото върховете му са зададени чрез масивите *xPoint* и *yPoint*. Всеки връх се дефинира с наредената двойка (*xPoint*[i] , *yPoint*[i]).

xPoints – масив с абцисните координати на точките
yPoints – масив с ординатните координати на точките
nPoints – броя на точките

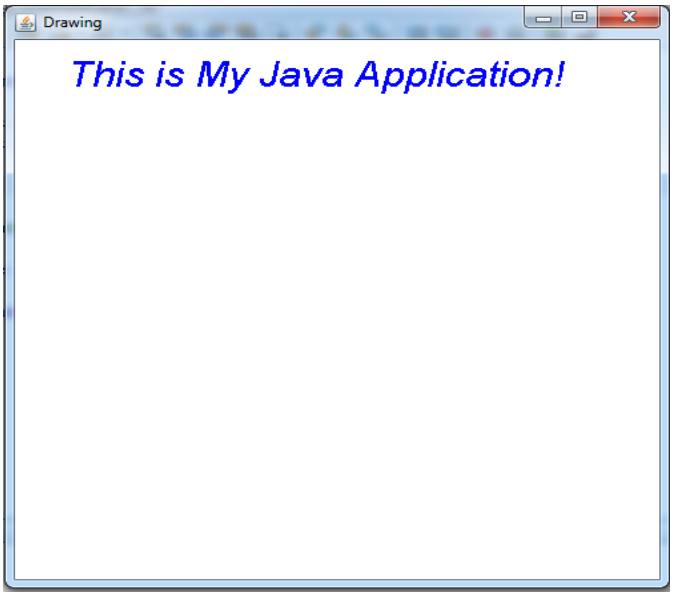


```
16  public class DrawPanel extends JPanel
17  {
19      public void paintComponent(Graphics g)
20      {
21          super.paintComponent(g);
22
23          g.setColor (Color.BLUE);
24          int[] xPoints = {20 , 30 ,120 , 400 , 200};        //Масив с абцисните координати
25          int[] yPoints = {100 , 400 , 300 , 200 , 30};      //Масив с ординатните координати
26          g.drawPolygon(xPoints , yPoints , xPoints.length); //Рисува многоъгълника. Забележка!: xPoints.length == yPoints.length
27
28      }
29  }
```

g.drawString (String *str* , int *x* , int *y*)

Методът рисува текста подаден от *str* като координатите на **първата буква** са *x* и *y.* Можем да манипулираме шрифта и големина като предварително ги зададем чрез метода g.setFont (Font f).

str – текста, който искаме да принтираме
x ,*y* – координатите на първата буква



```
17  public class DrawPanel extends JPanel
18  {
19      public void paintComponent(Graphics g)
20      {
21          super.paintComponent(g);
22
23          g.setColor (Color.BLUE);  //Избираме цвят
24
25          Font f = new Font("My Font" , Font.ITALIC , 30); //Създаваме си шрифт
26          g.setFont(f);                                    //Казваме, че искаме да пишем на този шрифт
27          g.drawString("This is My Java Application!", 40 , 40); //Рисуваме текста
28      }
29  }
```

Fill методи

Методите fill са абсолютно аналогични като draw методите с малката разлика, че освен чертаят фигурата , те я и оцветяват вътрешността в цвета, който е предварително зададен (напр. g.setColor(Color.RED)) Методите са съответно :

g.fillRect (int x , int y , int width , int height)

g.fillArc (int x , int y , int width , int height , int startAngle , int arcAngle)

g.fillPolygon(int[] xPoints , int[] yPoints , int nPoints)

Задача :

Да се напише програма, която рисува следната картинка

Примерна реализасия на метода:

```
20      public void paintComponent(Graphics g)
21      {
22          super.paintComponent(g);
23
24          //Draws the base of the house.
25          g.setColor(Color.YELLOW);  //Sets the color of the element to be YELLOW.
26          g.fillRect(getWidth()/4, getHeight()/2 , getWidth()/2, (getHeight()/2) - 5);
27
28          //Draws the roof of the house
29          g.setColor(Color.RED);
30          int[] xRoof = {getWidth()/4 - 15 , getWidth()/2    , (3 * getWidth() / 4) + 15};  //After making a few math calulations
31          int[] yRoof = {getHeight()/2     , getHeight()/10 , getHeight()/2};               //we create arrays that contain the
32          g.fillPolygon(xRoof, yRoof, xRoof.length);                                        //coordinates of the roof Polygon(Triangle)
```

```
33
34          //Here we paint the door
35          g.setColor(Color.BLACK);
36          //The arc represents the top part of the door, giving it a curve.
37          g.fillArc(2*getWidth()/5, 3 * getHeight() / 4, 2 * getWidth()/10, getHeight()/15, 0 , 180);
38          //The rectangle represents the base part of the door. It's coordiantes should be based in the arc's.
39          g.fillRect(2*getWidth()/5 , 3*getHeight()/4 + getHeight()/30 , 2*getWidth()/10 , getHeight()/4 - getHeight()/30 - 5);
40
41
42          //He we create a font with which we are going to write the string.
43          Font f = new Font ("myFont" , Font.ITALIC , 20);
44          g.setFont(f);
45          g.setColor (Color.BLUE);
46          g.drawString ("The House I DO NOT want to have!" , 5 , 21);
47      }//end of paintComponent
48  }//end of class
```