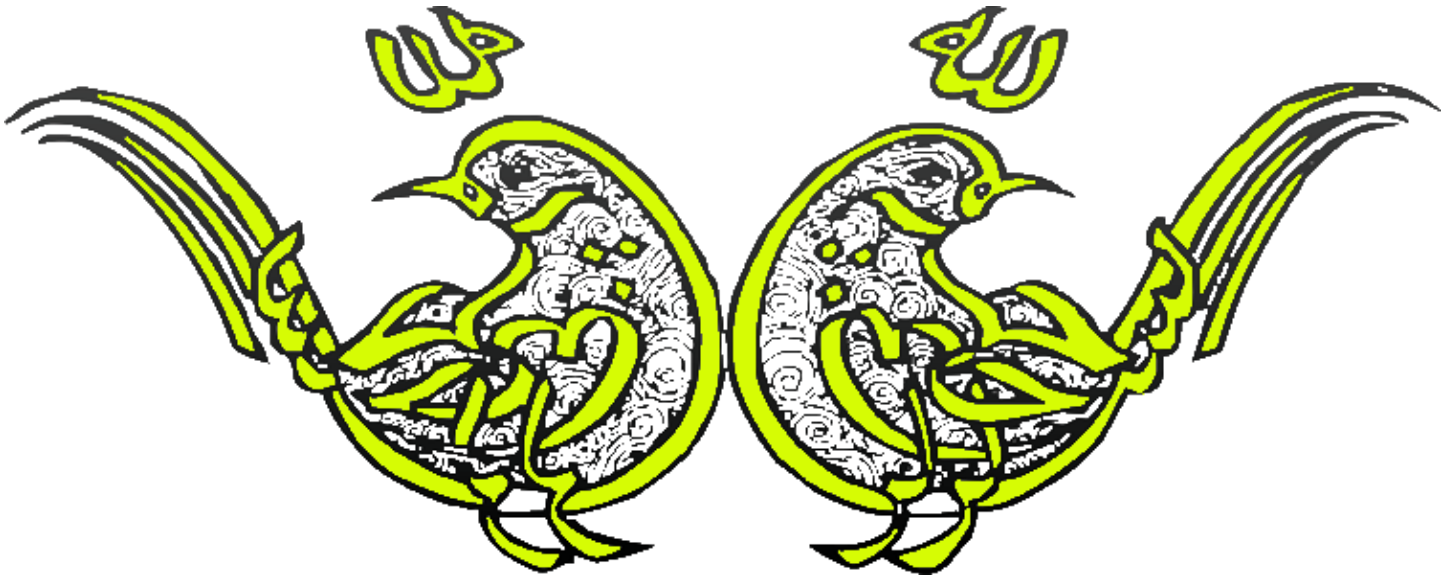


تقدیم به همه هموطنان عزیزم

آموزش زبان برنامه نویسی جاوا

گرافیک در جاوا – پکیج Swing

جلسه بیست و پنجم

کلاس Graphics

نویسنده : رحمان زارعی

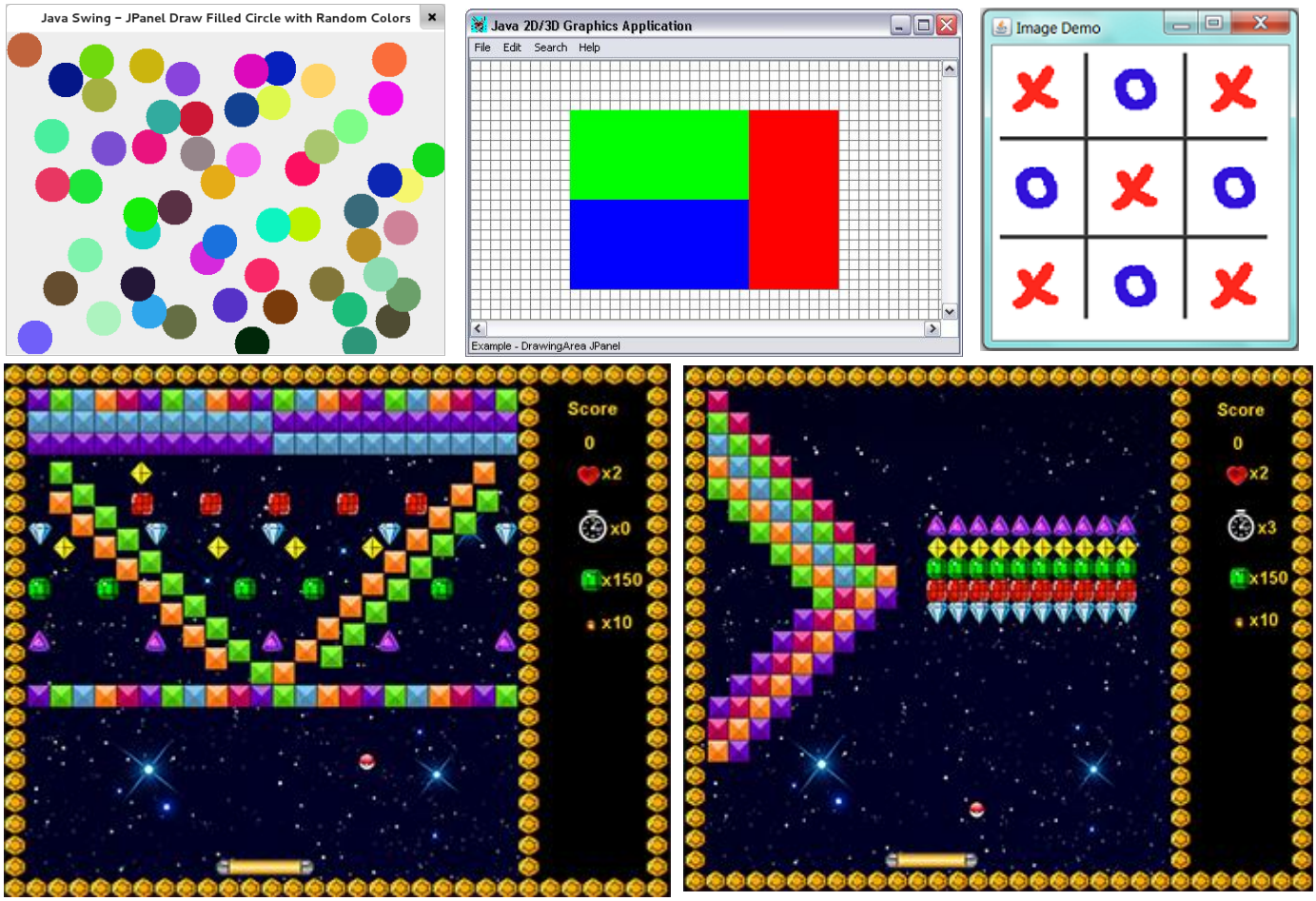
جاوا را ساده،آسان و شیرین بنوشید!!!!

نمایش گرافیکی در Swing :

کلاس Graphics متدهای فراوانی را برای ایجاد برنامه های گرافیکی فراهم کرده است.برای استفاده از این کلاس و دسرسی به ویژگی ها و رفتارهای آن باید پکیج زیر را در بالای کلاس خود import کنید:

```
import java.awt.Graphics;
```

قبل از این که بریم سراغ بررسی این کلاس برای درک تنها بخشی از امکانات این کلاس تصویر(۱) را مشاهده کنید:



تصویر(۱)

- در تصویر(۱) بخشی از امکاناتی که کلاس Graphics برای ایجاد برنامه های گرافیکی در جاوا در اختیار ما قرار می دهد را مشاهده می کنید.

برای ساخت بازی و برنامه های کاربردی گرافیکی به کلاس Graphics نیاز پیدا می کنیم.

پرکاربردترین متدهای کلاس Graphics در جاوا:

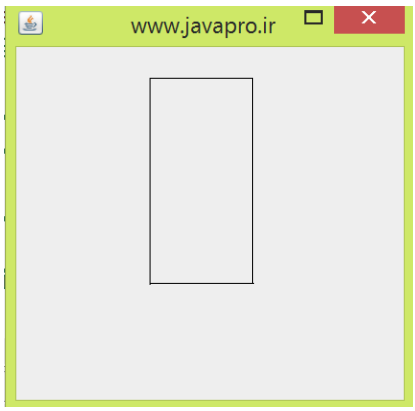
```
public abstract void drawString(String str, int x, int y)
```

- برای رسم رشته(String) مشخص استفاده می شود. تصویر(۲)

تصویر(۲)

```
public void drawRect(int x, int y, int width, int height)
```

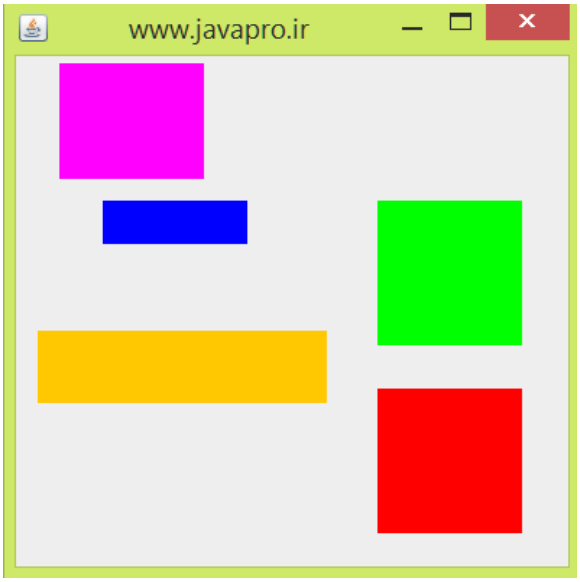
- برای رسم یک مستطیل یا مربع با عرض و ارتفاع مشخص استفاده می شود. تصویر(۳)



تصویر(۳)

```
public abstract void fillRect(int x, int y, int width, int height)
```

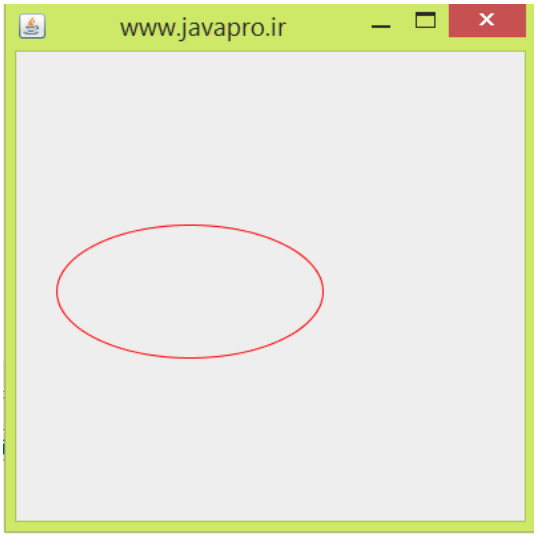
- برای ایجاد یک مستطیل یا مربع تو پر با رنگ پیشفرض(default) و عرض و ارتفاع مشخص استفاده می شود.تصویر(۴)



تصویر(۴)

```
public abstract void drawOval(int x, int y, int width, int height)
```

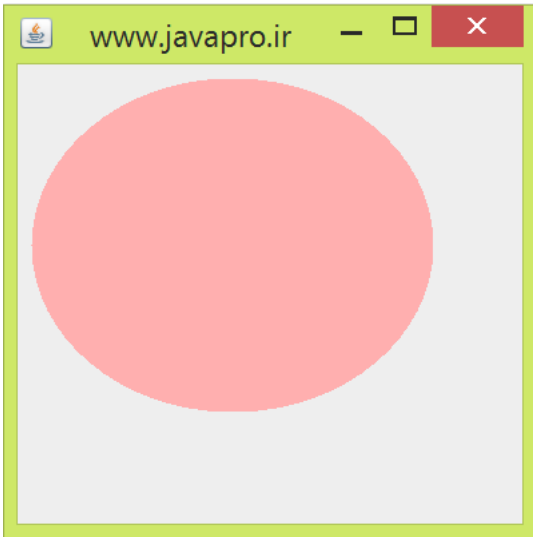
- برای رسم یک بیضی یا دایره با عر ض و ارتفاع مشخص استفاده می شود.تصویر(۵)



تصویر(۵)

```
public abstract void fillOval(int x, int y, int width, int height)
```

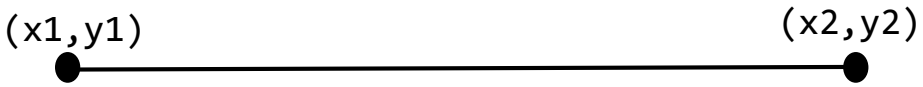
برای ایجاد یک بیضی یا دایره تو پر با رنگ پیشفرض(default) و عرض و ارتفاع مشخص استفاده می شود. تصویر(۶)

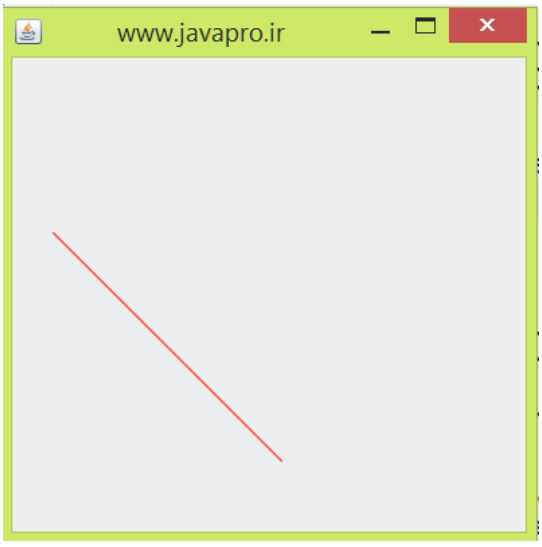


تصویر(۶)

```
public abstract void drawLine(int x1, int y1, int x2, int y2)
```

- برای رسم خط بین نقطه (x1,y1) و نقطه (x2,y2) در برنامه استفاده می شود.تصویر(۷)

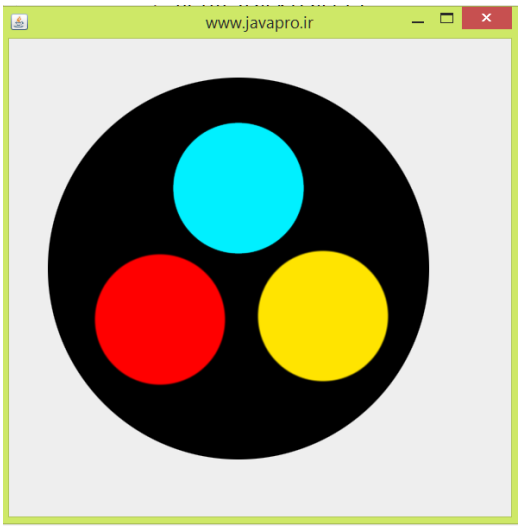




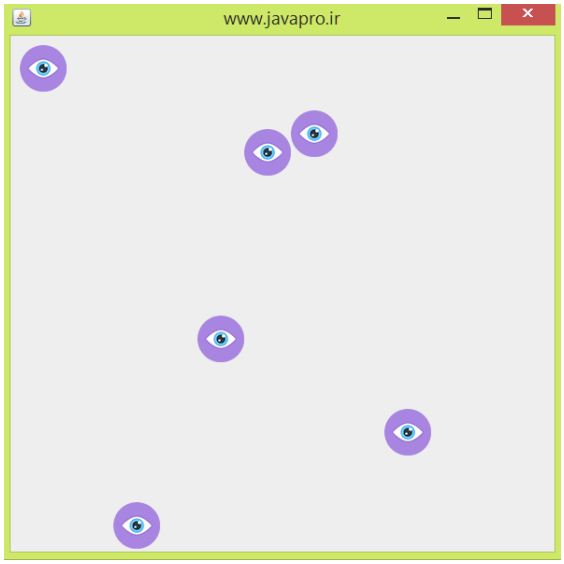
تصویر(۷)

```
public abstract boolean drawImage(Image img, int x, int y,
ImageObserver observer)
```

- برای رسم تصویر مشخص در برنامه استفاده می شود.تصاویر(۸) و (۹)



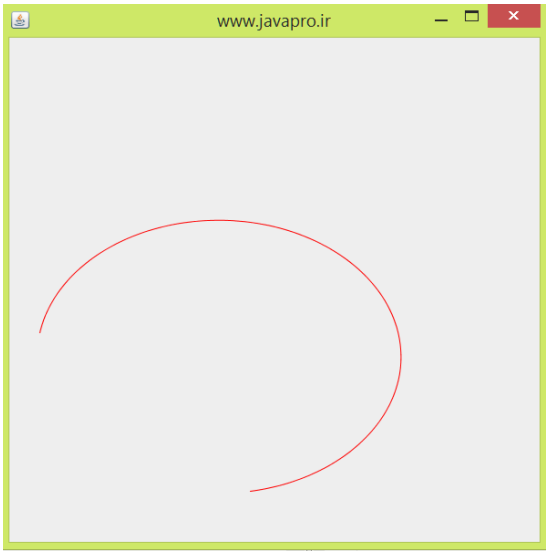
تصویر(۸)



تصویر(۹)

```
public abstract void drawArc(int x, int y, int width, int height,
int startAngle, int arcAngle)
```

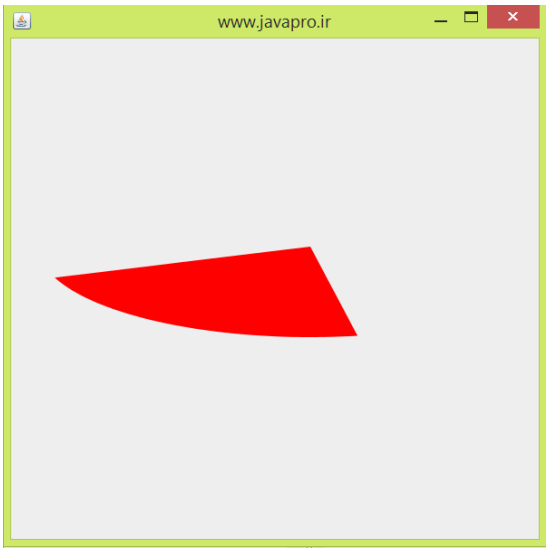
- برای رسم قوس دایره ای یا بیضوی استفاده می شود.تصویر(۱۰)



تصویر(۱۰)

```
public abstract void fillArc(int x, int y, int width, int height, int
startAngle, int arcAngle)
```

- برای رسم قوس دایره ای یا بیضوی **توپر** استفاده می شود.تصویر(۱۱)



تصویر(۱۱)

```
public abstract void setColor(Color c)
```

- برای تنظیم رنگ مورد نظر برای رنگ فعلی اشکال گرافیکی رسم شده استفاده می شود.

```
public abstract void setFont(Font font)
```

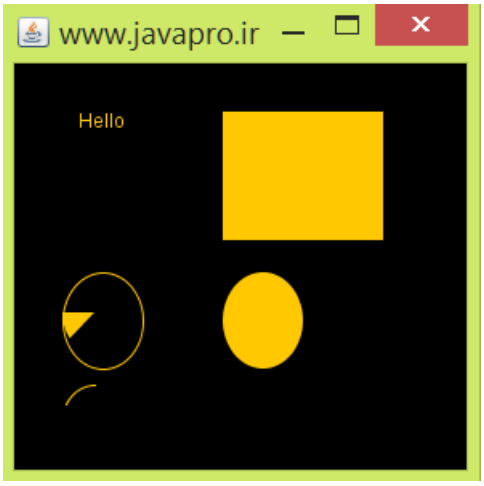
- برای **تنظیم فونت مورد نظر** برای **فونت فعلی** رشته گرافیکی رسم شده استفاده می شود.

مثال:

```
package javapro.ir;

import java.awt.*;
import java.awt.image.BufferedImage;
import java.io.BufferedWriter;
import java.io.File;
import java.io.IOException;

import javax.imageio.ImageIO;
import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JPanel;

public class DisplayGraphics extends JPanel {

	public DisplayGraphics() {

	}

	public void paint(Graphics g) {
		super.paint(g);
		Graphics2D g2 = (Graphics2D) g;

		g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
				RenderingHints.VALUE_ANTIALIAS_ON);

		setBackground(Color.black);
		setForeground(Color.orange);
		g2.drawString("Hello", 40, 40);

		g2.fillRect(130, 30, 100, 80);
		g2.drawOval(30, 130, 50, 60);
```

```
            g2.fillOval(130, 130, 50, 60);
            g2.drawArc(30, 200, 40, 50, 90, 60);
            g2.fillArc(30, 130, 40, 50, 180, 40);

      }

      public static void main(String[] args) {
            DisplayGraphics m = new DisplayGraphics();
            JFrame f = new JFrame();
            f.add(m);

            f.setSize(300, 300);
            f.setTitle("www.javapro.ir");
            f.setVisible(true);

      }

}
```

خروجی: تصویر(۱۲)



تصویر(۱۲)

توضیحات:

```
public class DisplayGraphics extends JPanel {
```

- در ابتدا جلسه متدهای پرکاربرد کلاس Graphics را بررسی کردیم و اغلب اوقات از کلمه کشیدن یا رسم کردن (Draw) صحبت کردیم. همان طور که در دنیای واقعی برای رسم کردن و کشیدن اشکال نیاز به یک پنل نقاشی داریم در جاوا نیز برای رسم کردن اشکال گرافیکی نیاز به یک پنل نقاشی داریم که از کلاس JPanel برای این کار استفاده می کنیم.
- در اینجا برای دسرسی به امکانات ، ویژگی ها و رفتارهای کلاس JPanel ، کلاس خود را extends به کلاس JPanel کرده ایم. با این کار کلاس ما زیر کلاس و فرزند کلاس JPanel می شود.

```
public void paint(Graphics g) {
```

- متد paint در کلاس JComponet وجود دارد.
- همان طور که از اسم متد paint پیداست ، این متد برای کشیدن و رسم کردن اشکال گرافیکی در برنامه ما استفاده می شود.
- **کلاس ما** کلاس JPanel را به ارث برده است از طرفی **کلاس JPanel** فرزند **کلاس JComponet** می باشد، پس کلاس ما نیز زیر کلاس و فرزند کلاس **JComponet** محسوب می شود و به همین خاطر می توانیم از متد paint در کلاسمون استفاده کرده و آن را باتوجه به نیاز overrid کنیم.
- یکی از نکات مهم متد paint این است که اگر در کلاسی که JPanel (JPanel فرزند JComponet هستش) را به ارث برده پیاده سازی شود ، بعد از شی سازی از کلاس به صورت اتوماتیک صدا زده و دستورات درون آن اجرا می شود و نیازی نیست بصورت مستقیم این متد را صدا بزنیم.
- در کل متد paint در کلاسی که به گونه ای فرزند JComponet باشد هنگام اجرای برنامه به صورت خودکار صدا زده می شود.
- پارامتر این متد Graphics g می باشد.
- از طریق شی از نوع کلاس Graphics متدهای گرافیکی موجود در کلاس Graphics را صدا می زنیم .

```
super.paint(g);
```

- وقتی که شما دستور بالا را پیاده سازی می کنید، متد paint **کلاس پدر** کلاس ما صدا زده می شود!
- خب کلاس ما اسمش DisplayGraphics هستش که فرزند JPanel می باشد.از طرفی JPanel نیز فرزند JComponet می باشد. پس کلاس ما یعنی DisplayGraphics فرزند کلاس JComponet می شود.حالا

در کلاس فرزند یعنی DisplayGraphics با کلمه کلیدی super متد paint کلاس پدر یعنی JComponet را صدا زده ایم. با این کار وقتی متد paint در کلاس DisplayGraphics صدا زده میشه هم زمان متد paint کلاس پدر یعنی JComponet نیز صدا زده شده و اجرا می شود.

- اگه متوجه نشدید چی شده! نگران نباشید از دستور بالا همیشه باید در برنامه گرافیکی به همین قالبی که هست استفاده کنید☺

```
Graphics2D g2 = (Graphics2D) g;
```

- در اینجا عمل casting یا تبدیل نوع انجام داده ایم.
- برای دسری به امکانات کلاس Graphics2D شی از نوع کلاس Graphics را به شی از نوع کلاس Graphics2D تبدیل کرده ایم.
- بهتره همیشه این تبدیل را انجام دهیم.البته برای یک برنامه گرافیکی روان تر و زیباتر حتما باید این تبدیل را انجام دهید.

```
g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);
```

- متد setRenderingHint وظیفه تنظیم کیفیت اشکال رسم شده ما را دارد.
- متد setRenderingHint دو پارامتر می گیرد.
- پارامتر های آن توسط کلید ها و مقادیری که مستقیم از طریق کلاس RenderingHints صدا زده می شوند، مقداردهی می شوند.
- پارامتر RenderingHints.KEY_ANTIALIASING یعنی اشکال گرافیکی که قراره رسم شوند ضد لغزش باشند.
- خب پارامتر بعدی متد هم از ما مقدار میخواد که ما RenderingHints.VALUE_ANTIALIAS_ON به ان داده ایم. در اینجا حالت ضد لغزش اشکال گرافیکی رو روشن (on) می کنیم. اگر از دستور RenderingHints.VALUE_ANTIALIAS_OFF استفاده کنید حالت ضد لغزش خاموش یا off می شود.
- در کل همیشه برای رسم اشکل گرافیکی در متد paint باید برای بالا بردن کیفیت برنامه گرافیکی خود از دستور زیر بصورت ثابت استفاده کنیم:

```
g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);
```

- اگه استفاده نکنیم برنامه ما دچار لرزش می شود.مخصوص اگر قصد داشته باشید انیمیشن و بازی و... طراحی کنید.

```
setBackground(Color.black);
```

- تعیین پس زمینه برای JPanel برنامه خود

```
setForeground(Color.orange);
```

- این دستور برای تعیین رنگ برای اشکال و متن های گرافیکی رسم شده می باشد.

```
1. g2.drawString("Hello", 40, 40);
2. g2.fillRect(130, 30, 100, 80);
3. g2.drawOval(30, 130, 50, 60);
4. g2.fillOval(130, 130, 50, 60);
5. g2.drawArc(30, 200, 40, 50, 90, 60);
6. g2.fillArc(30, 130, 40, 50, 180, 40);
```

۱. رسم یک رشته در پنل برنامه، پارامتر اول متن مورد نظر، پارامتر دوم و سوم تعیین مختصات نقطه قرار گیری در پنل برنامه می باشد. پارامتر دوم x (مقدار افقی) و پارامتر سوم y (مقدار عمودی) می باشد.

۲. کشیدن یک مستطیل توپر، (x=130,y=30)، عرض مستطیل=۱۰۰ ، ارتفاع مستطیل=۸۰

۳. کشیدن بیضی ، (x=30,y=130)، عرض بیضی=۵۰ ، ارتفاع بیضی=۶۰

۴. کشیدن بیضی توپر، (x=130,y=30)، عرض بیضی توپر=۵۰ ، ارتفاع بیضی توپر=۶۰

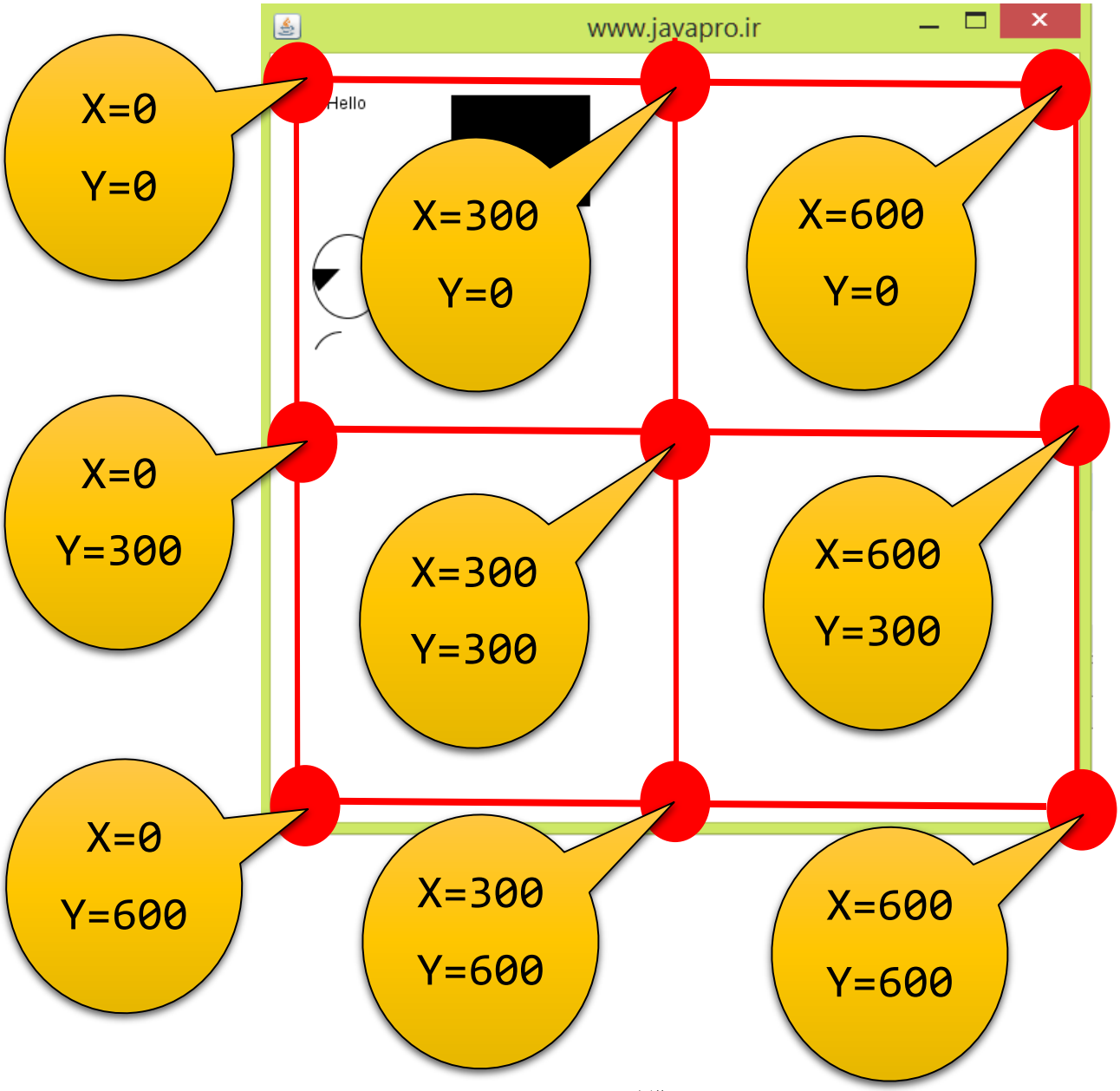
۵. کشیدن قوس بیضوی توخالی (x=30,y=200)، عرض =۴۰ ، ارتفاع =۵۰ – زاویه شروع=۹۰ ، زاویه اطراف قوس=۶۰

- قوس بخشی از یک بیضی است، در واقع می توانیم اینجوری فکر کنیم که با بهم پیوستن چند قوس بصورت منظم یک بیضی تشکیل می شود. متد drawArc با داشتن ۶ پارامتر یک قوس طراحی می کند چهار پارامتر اول شبیه به متد drawOval می باشد. دو پارامتر آخر یکی مربوط به زاویه شروع و دیگری مربوط به شماره درجه اطراف قوس است.

۶. متد fillArc برای کشیدن قوس توپر می باشد.پارامترهای این متد نیز شبیه متد drawArc است.

- برای درک مختصات درون یک فریم و پنل برنامه تصویر(۱۳) را مشاهده کنید:

- فرض کنید یک JFram داریم که آن اندازه ۶۰۰*۶۰۰ می باشد.مختصات نقاط مختلف این فریم بصورت زیر است:



تصویر(۱۳)

- بگذریم بریم سراغ ادامه توضیح مثال:

```
public static void main(String[] args) {
DisplayGraphics m = new DisplayGraphics();
JFrame f = new JFrame();
f.add(m);

f.setSize(300, 300);
f.setTitle("www.javapro.ir");
f.setVisible(true);

}
```

```
DisplayGraphics m = new DisplayGraphics();
```

- در متد main از کلاسمون شی ایجاد می کنیم.یادآوری می کنم که کلاس ما کلاس JPanel را به ارث برده است به نوعی یک component گرافیکی حساب می شود.

```
JFrame f = new JFrame();
```

- ما تا اینجا برا خودمون نقاشی کشیدیم ، شکل رسم کردیم، حالا نیاز به یک پایه برای نمایش نقاشی ها و پنل خودمون داریم که از کلاس JFrame شی ایجاد می کنیم.

```
f.add(m);
```

با متد add شی m که از نوع کلاس DisplayGraphics هستش را به فریم خود اضاف می کنیم.چطور تونستیم همچین کاری کنیم؟ از آنجایی که کلاس ما کلاس JPanel را به ارث برده است حق چنین کاری را داریم.

```
f.setSize(300, 300);
```

- تعیین اندازه فریم

```
f.setTitle("www.javapro.ir");
```

- تعیین عنوان برای فریم

```
f.setVisible(true);
```

- برای نمایش فریم و تمام اجزای گرافیکی برنامه

در آینده در قالب یک اموزش پروژه محور به صورت گسترده از این کلاس و سایر مفاهیم پایه و گرافیک جاوا استفاده خواهیم کرد.

پیروز و موفق باشید

سایت آموزش زبان جاوا به زبان ساده،آسان و شیرین!!!

www.JAVAPro.ir

آموزش جاوا SE را با تجربه شخصی و به زبان خودمونی یاد بگیرید!!!!!

بازدید ازکانال

بازدید از سایت

هر روز مفاهیم و مثال های جدید به سایت اضافه می شود برای اطلاع از مطالب جدید روی سایت عضو کانال شوید.

دخل و تصرف ، ویرایش و کپی زدن تمامی آموزش های جاوالایک به دور از اخلاق حرفه ای ست و حرام می باشد.