

تقديم به همه هموطنان عزیزم

آموزش زبان برنامه نویسی جاوا

گرافیک در جاوا - پکیج Swing

جلسه بیست و ششم

نمایش تصویر در Swing

نویسنده: رحمان زارعی

جاوا را ساده، آسان و شیرین بنوشید!!!!



این جلسه آموزشی رایگان است، فروش و ویرایش آن ممنوع و حرام می باشد. اما این کتاب را می توانید همین جور که هست در سایت و شبکه اجتماعی خود به اشتراک بگذارید.

برای نمایش تصویر، ما می توانیم از متد `drawImage()` استفاده کنیم. متد `drawImage()` متعلق به کلاس `Graphics` می باشد.

سینتکس یا نحوه نوشتن متد `drawImage()`:

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

- از این متد برای رسم و نمایش تصویر در برنامه استفاده می شود.
- پارامترهای این متد بصورت زیر می باشد:
- پارامتر اول شی از نوع کلاس `Image` می باشد. این شی حاوی تصویر مورد نظر ما می باشد.
- `x,y` مختصات نقطه قرار گیری تصویر در برنامه می باشد.
- پارامتر `observer` اعمال بروزرسانی ها به یک تصویر که بصورت یکنواخت بارگذاری می شود اطلاع می دهد. در کل بیخیال توضیح! هر وقت خواستید از متد `drawImage` استفاده کنید اگر قصد دارید تصویر خودتون رو از طریق کلاس `BufferedImage` بخوانید مقدار پارامتر `observer` را `null` قرار دهید و اگر از طریق کلاس `Toolkit` تصویر خود را از کامپیوتر خواندید مقدار پارامتر این متد را `this` (اشاره به شی کلاسمون که `JPanel` را به ارث برده است دارد) قرار دهید و خودتون رو خلاص کنید 😊

- خب این متد برای رسم و نمایش تصویر در `panel` یا `frame` برنامه استفاده می شود. حالا چطور تصویر را از محل کامپیوتر بخوانیم و به عنوان پارامتر به این متد بدهیم که در برنامه نمایش داده شود؟ روش های مختلفی برای خواندن تصویر از کامپیوتر وجود دارد که در قالب مثال برخی از این روش ها رو بررسی می کنیم.

روش اول – خواندن تصویر از کامپیوتر با استفاده از کلاس `Toolkit` و کلاس `Image`:

```
package javapro.ir;

import java.awt.*;

import javax.swing.JFrame;
import javax.swing.JPanel;

public class MyPanel extends JPanel {

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

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

        Toolkit t = Toolkit.getDefaultToolkit();
        Image i = t.getImage("lemon.png");
        g2.drawImage(i, 80, 80, this);
    }

    public static void main(String[] args) {
        MyPanel m = new MyPanel();
        JFrame f = new JFrame();
        f.add(m);
        f.setTitle("www.javapro.ir");
        f.setSize(400, 400);
        f.setVisible(true);
    }
}
```

خروجی: تصویر (۱)



تصویر (۱)

توضیحات:

```
import java.awt.*;
import javax.swing.JFrame;
import javax.swing.JPanel;
```

- پکیج های مورد نیاز خود را import می کنیم.

```
public class MyPanel extends JPanel {
```

- اگه قصد داریم یک برنامه گرافیکی بنویسیم ، کلاس خود را extends به کلاس JPanel می کنیم. با این کار کلاس ما به ویژگی ها و رفتارهای کلاس JPanel دسترسی پیدا می کند.

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

- برای کشیدن و رسم کردن اشکال و تصاویر گرافیکی خود از متد paint استفاده می کنیم.
- این متد یک شی از کلاس Graphics را به عنوان پارامتر دارد.
- کلاس Graphics متدها و امکانات نوشتن یک برنامه گرافیکی را به ما می دهد.
- کلاس Graphics درون پکیج AWT جاوا قرار دارد:

```
import java.awt.*;
```

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

- متد paint کلاس پدر را صدا زده ایم.

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

- برای بالابردن کیفیت نمایش و کنترل لغزش گرافیک برنامه استفاده می شود.

کلا در متد paint دستورات زیر رو بصورت ثابت کنارهم استفاده کنید!

```
super.paint(g);  
Graphics2D g2 = (Graphics2D) g;  
  
g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,  
RenderingHints.VALUE_ANTIALIAS_ON);
```

```
Toolkit t = Toolkit.getDefaultToolkit();
```

- ما قصد داریم با استفاده از کلاس Toolkit تصویر خود را از کامپیوتر بخوانیم. پس از این کلاس به روش بالا شی ایجاد می کنیم.

```
Image i = t.getImage("lemon.png");
```

- متد getImage برای دریافت تصویر از کامپیوتر استفاده می شود. متعلق به کلاس Toolkit می باشد.
- با استفاده از شی t که از نوع کلاس Toolkit می باشد متد getImage را صدا می زنیم.
- آدرس تصویر خود در کامپیوتر را به عنوان پارامتر به متد getImage می دهیم.
- حالا تصویر دریافت شده از کامپیوتر را درون شی از نوع کلاس Image قرار می دهیم.

```
g2.drawImage(i, 80, 80, this);
```

- از این دستور برای کشیدن تصویر در برنامه استفاده می کنیم. همان طور که می بینید شی i را به عنوان تصویر به پارامتر این متد داده ایم.
- دو عدد ۸۰ هم مختصات نقطه قرار گیری تصویر در برنامه می باشد.
- پارامتر ImageObserver این متد را نیز this قرار دادیم.

روش دوم - خواندن تصویر از کامپیوتر با استفاده از کلاس ImageIO و کلاس Image :

```
package javapro.ir;
import java.awt.*;
import java.io.File;
import java.io.IOException;
import javax.imageio.ImageIO;
import javax.swing.JFrame;
import javax.swing.JPanel;

public class MyPanel extends JPanel {

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

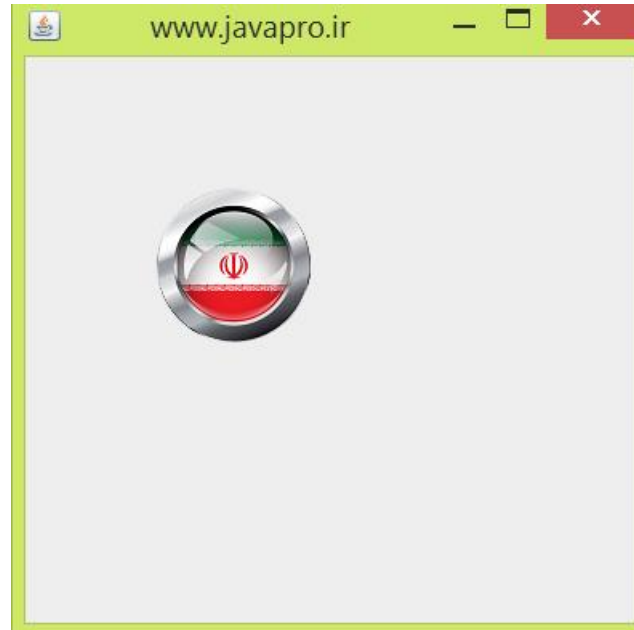
        g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
            RenderingHints.VALUE_ANTIALIAS_ON);

        Image i = null;
        try {
            i = ImageIO.read(new File("iran.png"));
        } catch (IOException e) {

            e.printStackTrace();
        }
        g2.drawImage(i, 80, 80, this);
    }

    public static void main(String[] args) {
        MyPanel m = new MyPanel();
        JFrame f = new JFrame();
        f.add(m);
        f.setTitle("www.javapro.ir");
        f.setSize(400, 400);
        f.setVisible(true);
    }
}
```

خروجی: تصویر (۲)



تصویر (۲)

توضیحات:

```
Image i = null;
```

- برای نگهداری تصویر خوانده از کامپیوتر نیاز به یک شی از نوع کلاس **Image** داریم.
- چون این شی را در درون یک متد تعریف کرده ایم ضروری بود مقداردهی اولیه کنیم پس ما مقدار **null** را بهش داده ایم.

```
try {  
    i = ImageIO.read(new File("iran.png"));  
} catch (IOException e) {  
    e.printStackTrace();  
}
```


- دستور زیر فایل تصویری ما را از کامپیوتر میخواند و درون شی `i` می ریزد:

```
ImageIO.read(new File("iran.png"))
```

- برای کنترل رخ دادن استثنای احتمالی بر اثر نبودن فایل در کامپیوتر از دستور `try-catch` استفاده کرده ایم.

```
g2.drawImage(i, 80, 80, null);
```

- در نهایت شی تصویر خود را درون متد `drawImage` برای نمایش در برنامه قرار می دهیم.
- نکته قابل توجه در متد بالا این است که پارامتر `ImageObserver` را `null` قرار داده ایم. مشکلی نیست `this` هم می توانید قرار دهید. خب در کلاس `Toolkit` الزامی بود که از تنها کلمه `this` استفاده کنیم.

روش سوم - خواندن تصویر از کامپیوتر با استفاده از کلاس `ImageIO` و کلاس `BufferedImage`:

```
package javapro.ir;

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

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

public class MyPanel extends JPanel {

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

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

        BufferedImage i = null;
        try {
            i = ImageIO.read(new File("quality.png"));
        }
    }
}
```

```
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
    g2.drawImage(i, 80, 80, null);  
}  
  
public static void main(String[] args) {  
    MyPanel m = new MyPanel();  
    JFrame f = new JFrame();  
    f.add(m);  
    f.setTitle("www.javapro.ir");  
    f.setSize(400, 400);  
    f.setVisible(true);  
}  
}
```

خروجی: تصویر (۳)



تصویر (۳)

```

BufferedImage i = null;
    try {
        i = ImageIO.read(new File("quality.png"));
    } catch (IOException e) {

        e.printStackTrace();
    }
    g2.drawImage(i, 80, 80, null);

```

- این مثال و کد تنها فرقی با مثال قبل استفاده از کلاس `BufferedImage` بجای کلاس `Image` می باشد.

```
g2.drawImage(i, 80, 80, null);
```

- ما می توانیم شی خود که از نوع کلاس `BufferedImage` هستش را جایگزین پارامتر متد `drawImage` که از نوع کلاس `Image` می باشد کنیم. چرا؟ چون کلاس `BufferedImage` فرزند کلاس `Image` می باشد.

برای درک بهتر سلسله مراتب این دو کلاس رو در شکل زیر تماشا کنید:

`java.lang.Object`

└ `java.awt.Image`

└ `java.awt.image.BufferedImage`

نکته: تمام این مثال هایی که کار کردیم یک تصویر با آدرس مشخص را از کامپیوتر میخواند و در برنامه نمایش می داد، پس هنگام تست کدهای بالا نیاز به تصویر دارید، در آدرس فولدر پروژه خود یک تصویر قرار دهید که نام و فرمت آن همنام با نام و فرمت تصویری باشد که در برنامه استفاده کرده ایم.

مثلا در مثال آخر نام و فرمت و آدرس تصویر ما بصورت زیر است:

"quality.png"

این فایل تصویری در فلدر پروژه ما قرار دارد. احتمالاً این سوال براتون پیش میاد که چرا آدرس خلاصه است؟!

پاسخ: در اصل آدرس فایل "quality.png" در فولدر پروژه ما بصورت زیر است:

"F:\\project java\\A\\quality.png"

همان طور که در آدرس بالا مشاهده می کنید، تصویر quality.png در فولدر A (فولدر A ، فولدر پروژه ما می باشد) و فولدر A در فولدر project java و فولدر project java در درایو F قرار دارد.

خب جاوا یک استثنای ویژه قائل شده گفته فایل های درون فولدر پروژه از خودمون اند! نیاز به این آدرس طولانی نیست! پس تنها نام و فرمت فایل موجود در فولدر پروژه را بکار ببرید، مثل آدرس زیر:

"quality.png"

سوال احتمالی بعدی!! خب شاید تصویر یا فایل ما در درایو دیگه ای از کامپیوتر قرار داشت و در فولدر پروژه نبود؟! جاوا این جا یکم شرمنده میشه و میگه بی زحمت در چنین شرایطی آدرس را بصورت کامل بکار ببرید. کد پستی هم فراموش نشه!!!! 😊 به مثال زیر که یک تصویر را خارج از فولدر پروژه و در مکان دیگری از کامپیوتر خوانده و نمایش داده ایم:

```
package javapro.ir;

import java.awt.*;
import java.io.File;
import java.io.IOException;

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

public class MyPanel extends JPanel {

    public void paint(Graphics g) {
        super.paint(g);
        Graphics2D g2 = (Graphics2D) g;
        setBackground(Color.white);
        g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
            RenderingHints.VALUE_ANTIALIAS_ON);

        Image i = null;
```

```
try {
    i = ImageIO.read(new File("D:\\files\\pic.png"));
} catch (IOException e) {

    e.printStackTrace();
}
g2.drawImage(i, 0, 0, null);
}

public static void main(String[] args) {
    MyPanel m = new MyPanel();
    JFrame f = new JFrame();
    f.add(m);
    f.setTitle("www.javapro.ir");
    f.setSize(400, 400);
    f.setVisible(true);
}
}
```

خروجی: تصویر (۴)



تصویر (۴)

توضیحات:

```
try {  
    i = ImageIO.read(new File("D:\\files\\pic.png"));  
} catch (IOException e) {  
    e.printStackTrace();  
}
```

- همان طور که در کد بالا مشاهده می کنید آدرس فایلی که خارج فولدر پروژه ما قرار دارد باید به صورت کامل و به شکل قالب بالا بکار برده شود.

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 از کانال

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.

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