

آموزش زبان برنامه نویسی جاوا

گرافیک در جاوا - پکیج Swing

جلسه سوم

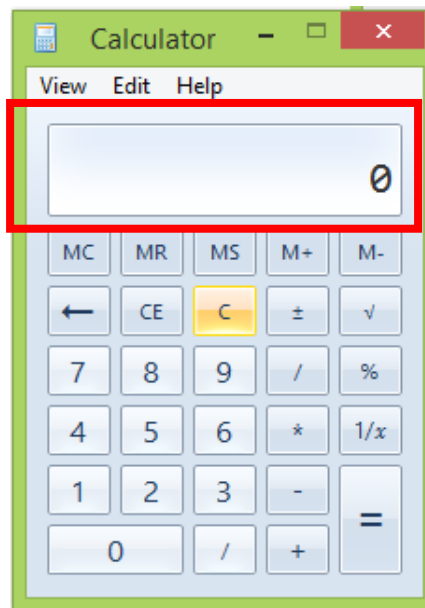
کلاس JLabel

نویسنده: رحمان زارعی

جاوا را ساده، آسان و شیرین بنوشید!!!!

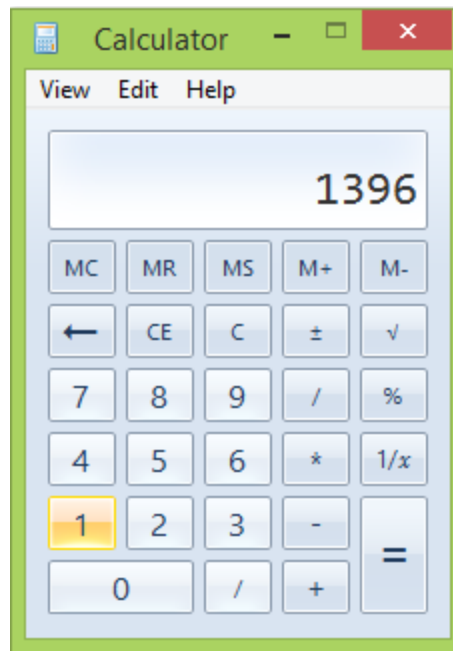


برای نمایش متن یا تصویر در برنامه گرافیکی خود از کلاس `JLabel` استفاده می کنیم. از ویژگی اجزای گرافیکی `JLabel` این است که متن نمایش داده شده در آن تنها با برنامه کاربردی نوشته شده قابل تغییر می باشد و کاربر توانایی ویرایش متن نمایش داده شده را ندارد. برای مثال در برنامه ماشین حساب ویندوز برای نمایش مقدار اعداد و نتیجه محاسبات از کلاس `JLabel` استفاده شده است کافی است همین الان ماشین حساب ویندوز رو باز کنید متوجه خواهید شد که صفحه نمایش ماشین حساب توسط کاربر غیر قابل ویرایش و دستکاری می باشد، اما از طریق دکمه های ماشین حساب (برنامه) قابل تغییر می باشد این همان خاصیت `label` می باشد تصویر(۱)



تصویر (۱)

- بخش قرمز مشخص شده در تصویر(۱) که همان صفحه نمایش ماشین حساب هستش از اجزای گرافیکی `label` استفاده شده است. همان طور که مشاهده می کنید این متن نمایش داده شده توسط کاربر غیر قابل تغییر می باشد اما از طریق دکمه های برنامه می توان مقدار `label` را تغییر داد. تصویر(۲)



تصویر (۲)

## سازنده های پر کاربرد کلاس JLabel:

### JLabel() ❖

برای ساخت یک نمونه label بدون متن و تصویر از این سازنده استفاده می شود.

```
JLabel a=new JLabel();
```

### JLabel(String s) ❖

برای ایجاد یک label با متن خاص از این سازنده استفاده می کنیم.

```
JLabel a=new JLabel(String s);
```

### JLabel(Icon i) ❖

برای ایجاد یک label با تصویر خاص از این سازنده استفاده می کنیم.

```
JLabel a=new JLabel(Icon i);
```

## متدهای پر کاربرد کلاس JLabel:

### String getText() ❖

این متد متنی که label نمایش داده است را بر می گرداند یا به عبارت دیگر متن label ما را به ما می دهد.

### void setText(String text) ❖

از این متد برای نمایش متن خود در label استفاده می کنیم.

### Icon getIcon() ❖

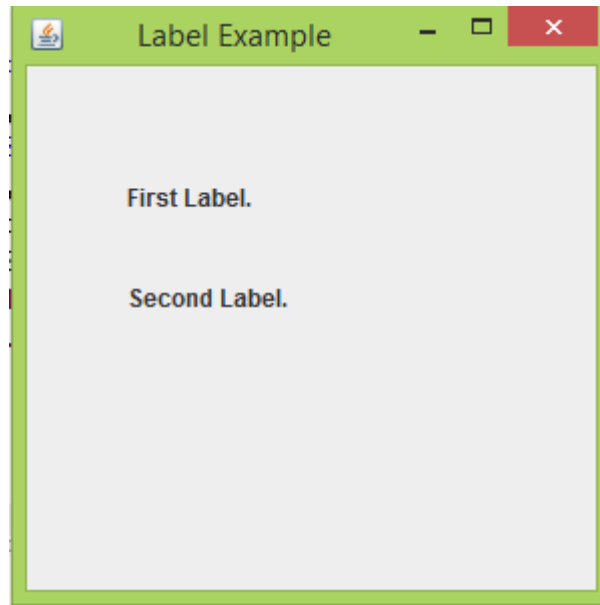
این متد تصویر گرافیکی نمایش داده شده در label را برای ما برمی گرداند.

- دوستان من برای یادگیری هر مبحثی از برنامه نویسی زیاد به تئوری ها اکتفا نکنید چرا که امکان داره حتی گمراه شوید! بیشتر به نمونه مثال ها و کدهای حل شده آنها توجه کنید.

## مثال ها (Examples):

```
package swing_javalike;
import javax.swing.*;
class LabelExample
{
public static void main(String args[])
{
    JFrame f= new JFrame("Label Example");
    JLabel l1,l2;
    l1=new JLabel("First Label.");
    l1.setBounds(50,50, 100,30);
    l2=new JLabel("Second Label.");
    l2.setBounds(50,100, 100,30);
    f.add(l1); f.add(l2);
    f.setSize(300,300);
    f.setLayout(null);
    f.setVisible(true);
}
}
```

خروجی: بعد از کامپایل و اجرای برنامه خروجی بصورت تصویر (۳) می باشد:



تصویر (۳)

- در تصویر (۳) دو label (لیبل) مشاهده می کنید که هر کدام متنی را در فریم ما نمایش داده اند.

```
JFrame f= new JFrame("Label Example");
```

- اجزای گرافیکی خود را روی یک اسکلت یا شاسی به نام فریم قرار می دهیم.
- در اینجا یک شی از کلاس JFrame ایجاد کرده ایم، و متنی را جایگزین پارامتر سازنده آن قرار داده ایم. که این متن جایگزین عنوان فریم ما می شود.

```
JLabel l1,l2;
```

- تعریف دو شی از نوع کلاس JLabel

```
l1=new JLabel("First Label.");
```

- با صدا زدن سازنده کلاس JLabel شی l1 ایجاد می شود. متن مورد نظر که قراره در لیبل ما نمایش داده شود را جایگزین پارامتر سازنده آن کرده ایم.

```
l1.setBounds(50,50, 100,30);
```

- با استفاده از متد `setBounds` مختصات و ابعاد لیبل l1 را در فریم خود مشخص می کنیم.

```
l2=new JLabel("Second Label.");
l2.setBounds(50,100, 100,30);
```

- کارایی که برای لیبل 1 انجام دادیم را برای لیبل 2 نیز انجام می‌دهیم.
- صدا زدن سازنده، مقدار دهی متنی که قراره در فریم نمایش داده شود و در نهایت تنظیم مختصات و ابعاد لیبل 2 در فریم

```
f.add(l1); f.add(l2);
```

- از طریق شی f که از نوع کلاس JFrame هستش ، متد add را صدا زده و اجزای گرافیکی لیبل 1 و 2 را به فریم خود اضافه می‌کنید.

```
f.setSize(300,300);
f.setLayout(null);
f.setVisible(true);
```

- تعیین سایز و اندازه فریم خود
- چون فعلا قصد نداریم از طرح بندی خاصی برای اجزای گرافیکی خود استفاده کنیم مقدار پارامتر متد setLayout را null قرار داده ایم.
- Layout برای طرح بندی و نحوه قرار گیری خاص اجزای گرافیکی استفاده می‌شود. خودش یک مبحث مفصل هستش که در آینده بهش خواهیم پرداخت.
- برای نمایش فریم و تمامی اجزای گرافیکی مقدار پارامتر متد setVisible را true می‌کنیم.

مثال: در برنامه زیر ما یک لیبل داریم و سه دکمه (button) و هر دکمه حاوی یک مقدار عددی می‌باشد، با کلیک کردن روی هر دکمه (button) مقدار عددی آن در label نمایش داده می‌شود.

```
package swing_javalike;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.*;

class LabelExample {
    public static void main(String args[]) {
        JFrame f = new JFrame("Label Example");
        JLabel l1;
        l1 = new JLabel("....");
        l1.setBounds(100, 50, 100, 30);
```

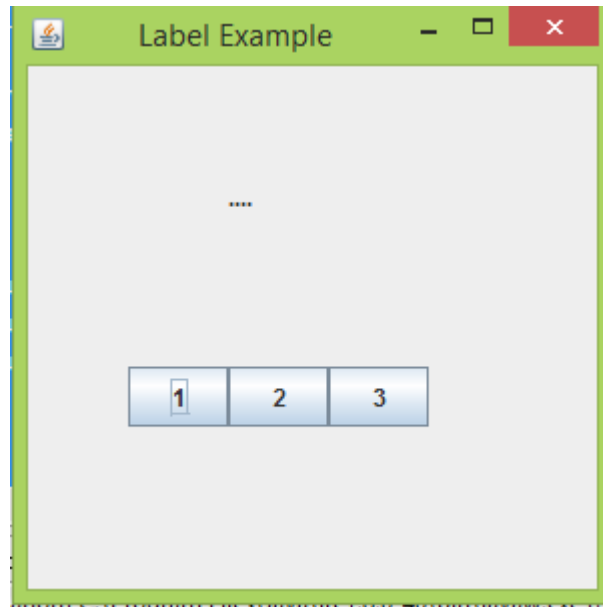
```
JButton b1 = new JButton("1");
JButton b2 = new JButton("2");
JButton b3 = new JButton("3");
b1.setBounds(50, 150, 50, 30);
b2.setBounds(100, 150, 50, 30);
b3.setBounds(150, 150, 50, 30);
b1.addActionListener(new ActionListener() {

    public void actionPerformed(ActionEvent arg0) {
        l1.setText("1");
    }
});
b2.addActionListener(new ActionListener() {

    public void actionPerformed(ActionEvent arg0) {
        l1.setText("2");
    }
});
b3.addActionListener(new ActionListener() {

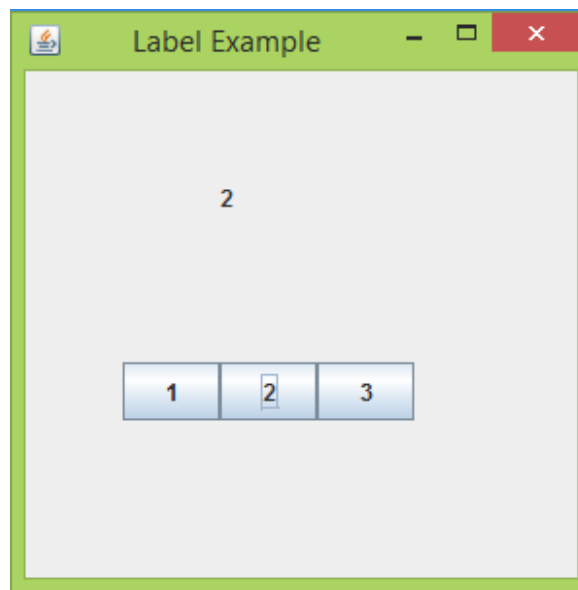
    public void actionPerformed(ActionEvent arg0) {
        l1.setText("3");
    }
});
f.add(l1);
f.add(b1);
f.add(b2);
f.add(b3);
f.setSize(300, 300);
f.setLayout(null);
f.setVisible(true);
}
}
```

خروجی: ابتدای برنامه که اجرا می شود بصورت تصویر (۴) می باشد:



تصویر (۴)

با زدن دکمه شماره ۲ متن **label** ما بصورت تصویر (۵) تغییر می کند:



تصویر (۵)

- همچنین با زدن هر دکمه شماره مورد نظر در لیبل نمایش داده می شود. همان طور که مشاهده می کنید این متن نمایش داده شده در فریم تنها از طریق برنامه قابل تغییر می باشد و ما نمی توانیم مستقیم متن لیبل را ویرایش کنیم.



نکته بسیار مهم: دوستان من میدونم الان چه حسی در مورد این نوع روش طراحی رابط گرافیکی برای برنامه کاربردی خود دارید!!! خیلی خسته کننده! طاقت فرسا! کد زدن های اضافی! شاید بگید که ما به جای اصل موضوع برنامه ای که میخوایم بنویسیم باید گرفتار کد زدن برای طراحی رابط گرافیکی (منظور همان دکمه ها، لیبل ها، فریم، منو و...) شویم! قطعاً با نظر شما موافق هستم! همان طور که در جلسات گذشته گفتیم در ویرایشگر Netbeans با drag کردن اجزای گرافیکی میشه سریع دیزاین یا همون اجزای گرافیکی برنامه خود را سریع طراحی کنید! پس چرا اینقدر خودمون رو معطل می کنیم و سراغ Netbeans نمی رویم؟! خب برنامه ما به این صورت است که ابتدا با این روش طراحی اجزای گرافیکی ( کد زدن برای ایجاد اجزای گرافیکی) آشنا بشیم بعدش سراغ Netbeans خواهیم رفت 😊

مثال: نمایش تصویر در یک label:

```
package swing_javalike;

import java.awt.event.ActionListener;

import javax.swing.AbstractButton;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;

public class Test {

    public static void main(String[] args) {

        JFrame f = new JFrame("Label Example");
        JLabel l1;
        l1 = new JLabel(new ImageIcon("javalikeTutorial.png"));
        l1.setBounds(0, 0, 300, 248);
        f.add(l1);
        f.setSize(300, 300);
        f.setLayout(null);
        f.setVisible(true);

    }

}
```

خروجی: بعد از اجرای برنامه خروجی بصورت تصویر (۶) می باشد:



تصویر (۶)

- ما در این برنامه یک تصویر در پوشه پروژمون رو در `label` موجود در برنامه نمایش داده ایم.
- شما می توانید یک تصویر به دلخواه خود در پوشه پروژه و همانم با `"javalikeTutorial.png"` برای تست قرار دهید.
- می توانیم ابعاد و مختصات تصویر و لیبل را به دلخواه تنظیم کنیم.

یکی از درخواست های شما عزیزان این است که چرا در جلسات آموزشی مثال کم میزنید؟! حق با شماست یک مبحث برنامه نویسی به مثال های فراوان نیاز دارد! خب از جهتی مثال فراوان باعث طولانی شدن و خسته کننده شدن جلسات آموزشی می شویم! خب راه کار چیست؟! ما در برنامه ای جداگانه قصد داریم انفجاری از مثال های جاوا با تنوع فراوان برای شما تولید کنیم. پس آموزش های ما را دنبال کنید.

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.

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