

آموزش زبان برنامه نویسی جاوا

گرافیک در جاوا - پکیج Swing

جلسه شانزدهم

کلاس JTree

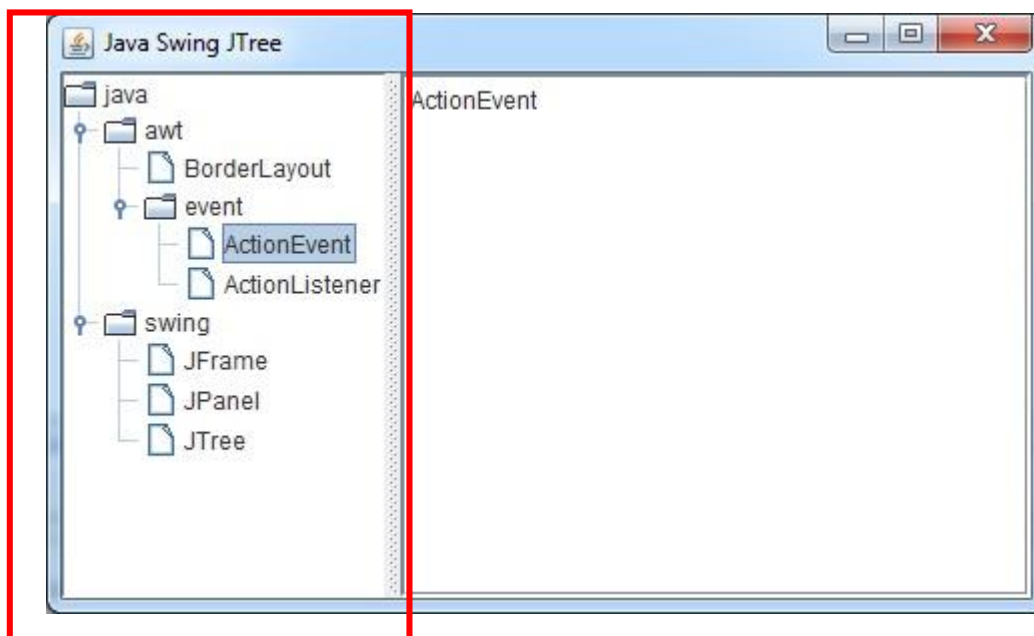
نویسنده: رحمان زارعی

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.

خب در ادامه مبحث گرافیک در جاوا در خدمت شما عزیزان هستیم. در این جلسه قصد داریم کلاس `JTree` را بررسی کنیم. همان طور که از اسم این کلاس پیداست برای نمایش داده ها به صورت ساختار درختی یا بصورت سلسله مراتبی از این اجزای گرافیکی در جاوا استفاده می کنیم. `JTree` یک `component` یا اجزای گرافیکی پیچیده است. همان طور که گفتیم داده ها در اجزای گرافیکی `JTree` به صورت سلسله مراتبی یا درختی می باشد به این صورت که یک گره ریشه داریم که پدر تمام گره های درخت می باشد. برای درک بهتر کلاس `JTree` تصویر (۱) را مشاهده کنید:



تصویر (۱)

هان طور که در تصویر (۱) مشاهده می کنید شاهده یک ساختار درختی یا سلسله مراتبی می باشیم. به این صورت که گره ای که بالای سایر گره ها قرار گرفته به عنوان ریشه یا پدر تمام گره های درخت در نظر می گیریم. می دونم الان سوالات زیادی براتون پیش اومده! الان یکی یکی بهشون می پردازیم! اولین سوال گره چیه؟! به عضوی از یک درخت گره می گویند. حالا به عضوی که

بالای سر همه گره ها قرار گرفته است ریشه درخت می گویند و از طرفی چون ریشه بالای سایر گره های درخت قرار گرفته به آن پدر همه گره های درخت هم می گویند. در تصویر(۱) عضوی که نام آن java هستش بالای سر سایر اعضا (گره های) درخت قرار گرفته است پس java ریشه و پدر سایر گره ها در این ساختار درخت می باشد.

## • سازنده های پدر کاربرد کلاس JTree:

سازنده	کاربرد
JTree()	ایجاد یک JTree با یک مدل نمونه.
JTree(Object[] value)	ایجاد یک JTree با هر عنصر از آرایه مشخص به عنوان فرزند یک گره ریشه جدید
JTree(TreeNode root)	ایجاد یک JTree که پارامتر root که از نوع TreeNode می باشد به عنوان ریشه درخت نمایش داده می شود.

خب توضیحات رو بی خیال ، مثال بچسبیم!!!

مثال:

```
package javalike;

import javax.swing.*;
import javax.swing.tree.DefaultMutableTreeNode;

public class TreeExample {
    JFrame f;

    TreeExample() {
        f = new JFrame();
        DefaultMutableTreeNode style = new DefaultMutableTreeNode("Style");
        DefaultMutableTreeNode color = new DefaultMutableTreeNode("color");
        DefaultMutableTreeNode font = new DefaultMutableTreeNode("font");
        style.add(color);
        style.add(font);
        DefaultMutableTreeNode red = new DefaultMutableTreeNode("red");
        DefaultMutableTreeNode blue = new DefaultMutableTreeNode("blue");
        DefaultMutableTreeNode black = new DefaultMutableTreeNode("black");
        DefaultMutableTreeNode green = new DefaultMutableTreeNode("green");
        color.add(red);
        color.add(blue);
    }
}
```

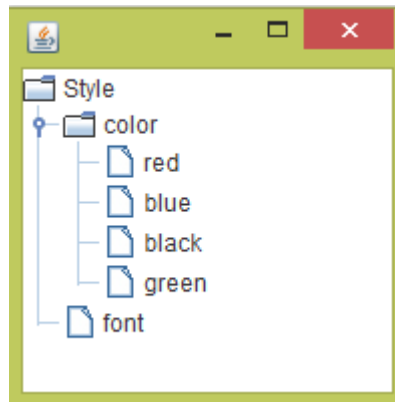
```

        color.add(black);
        color.add(green);
        JTree jt = new JTree(style);
        f.add(jt);
        f.setSize(200, 200);
        f.setVisible(true);
    }

    public static void main(String[] args) {
        new TreeExample();
    }
}

```

خروجی: تصویر (۲)



تصویر (۲)

## توضیحات:

```

DefaultMutableTreeNode style = new DefaultMutableTreeNode("Style");
DefaultMutableTreeNode color = new DefaultMutableTreeNode("color");
DefaultMutableTreeNode font = new DefaultMutableTreeNode("font");

```

- کلاس DefaultMutableTreeNode برای ایجاد گره (node) در یک درخت (JTree) استفاده می شود.
- خب در کدهای بالا گره هایی با شی ساختن از کلاس DefaultMutableTreeNode ایجاد کرده ایم.
- شی های font.color.style گره های درخت ما را تشکیل می دهند.

```

style.add(color);
style.add(font);

```

- در کد بالا گره **style** گره **color** را به خودش اضافه می کند با این کار گره **color** زیر گره **style** قرار می گیرد به عبارتی گره **style** پدر گره **color** می شود. همچنین گره **style** گره **font** را به خودش اضافه میکند با این کار گره **style** پدر گره **font** خواهد شد.
- تا اینجا گره **style** ریشه درخت و پدر دو گره **color** و **font** می باشد.

```
DefaultMutableTreeNode red = new DefaultMutableTreeNode("red");
DefaultMutableTreeNode blue = new DefaultMutableTreeNode("blue");
DefaultMutableTreeNode black = new DefaultMutableTreeNode("black");
DefaultMutableTreeNode green = new DefaultMutableTreeNode("green");
```

- خب چهار گره دیگر برای درخت ایجاد می کنیم.

```
color.add(red);
color.add(blue);
color.add(black);
color.add(green);
```

- چهار گره ایجاد شده را به گره **color** اضافه می کنیم.
- خب گره **color** پدر و ریشه چهار گره **red** ، **blue** ، **black** و **green** قرار میگیرد. چرا که تمام گره های مذکور پایین گره **color** قرار می گیرند.
- از طرفی چون گره **style** تا اینجا پدر گره **color** هستش و همچنین ریشه درخت ، پس می توان گفت گره **style** پدر گره های **red** ، **blue** ، **black** و **green** نیز می باشد.

```
JTree jt = new JTree(style);
```

- خب برای ایجاد ساختار درختی خود از کلاس **JTree** شی ایجاد می کنیم.
- گره ای که جایگزین پارامتر سازنده کلاس **JTree** می شود در جایگاه ریشه و پدر سایر گره های درخت قرار میگیرد.
- خب **style** هم طوری ایجادش کردیم که شایسته قرار گیری در جایگاه ریشه درخت می باشد.

```
1. f.add(jt);
2. f.setSize(200, 200);
3. f.setVisible(true);
```

۱. افزودن شی **jt** که از نوع کلاس **JTree** هستش ، به فریم برنامه.
۲. تنظیم سایز فریم
۳. صدا زدن این متد برای نمایش فریم و سایر اجزای گرافیکی الزامی می باشد.

- تا الان بیش از ۸۰۰ صفحه آموزش فارسی رایگان جاوا تهیه کرده ایم. برای رشد محتوای با کیفیت آموزشی ما را حمایت کنید.

پیروز و موفق باشید

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

[www.JAVAPRO.ir](http://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.