

آموزش زبان برنامه نویسی جاوا

گرافیک در جاوا - پکیج Swing

جلسه پانزدهم

کلاس JProgressBar

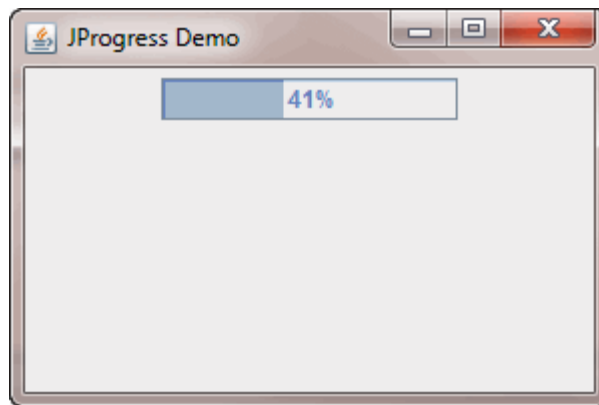
نویسنده: رحمان زارعی

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.

کلاس JProgressBar برای نمایش پیشرفت کار در برنامه استفاده می شود. برای درک بهتر از این اجزای گرافیکی در جاوا تصویر(۱) را مشاهده کنید:



تصویر(۱)

- مطمئنا با دیدن تصویر(۱) JProgressBar برای ما آشنا خواهد بود، چرا که ان اجزای گرافیکی را در جاهای مختلف از کامپیوتر خود دیده ایم مثلا هنگام دانلود ، فرآیند پیشرفت دانلود را با این ابزار گرافیکی نمایش می دهند.

• سازنده های پر کاربرد کلاس JProgressBar :

سازنده	کاربرد
<code>JProgressBar()</code>	ایجاد یک <code>ProgressBar</code> افقی بدون داشتن متن
<code>JProgressBar(int min, int max)</code>	ایجاد یک <code>ProgressBar</code> افقی با مقدار <code>minimum</code> و <code>maximum</code> مشخص
<code>JProgressBar(int orient)</code>	ایجاد یک <code>ProgressBar</code> با جهت یابی مشخص . منظور از جهت یابی حالت افقی یا عمودی بودن <code>ProgressBar</code> ما در برنامه می باشد. برای حالت افقی دستور ثابت <code>SwingConstants.HORIZONTAL</code> را جایگزین پارامتر <code>orient</code> می کنیم. برای حالت عمودی دستور ثابت <code>SwingConstants.VERTICAL</code> را جایگزین پارامتر <code>orient</code> می کنیم.
<code>JProgressBar(int orient, int min, int max)</code>	پارامترهای این سازنده ترکیبی از دو سازنده قبل می باشد. یعنی ایجاد یک <code>ProgressBar</code> با جهت یابی و با مقدار <code>minimum</code> و <code>maximum</code> مشخص

• متدهای پر کاربرد کلاس JProgressBar :

متد	کاربرد
<code>void setStringPainted(boolean b)</code>	این متد برای تعیین نمایش یا عدم نمایش یک <code>String</code> در <code>ProgressBar</code> ما استفاده می شود.
<code>void setString(String s)</code>	این متد برای مقداردهی رشته (متن) <code>ProgressBar</code> ما استفاده می شود.
<code>void setOrientation(int orientation)</code>	این متد برای تعیین جهت قرار گیری <code>ProgressBar</code> ما در برنامه استفاده می شود. این جهت قرارگیری ممکن است افقی یا عمودی باشد. برای حالت افقی دستور ثابت <code>SwingConstants.HORIZONTAL</code> را جایگزین پارامتر <code>orientation</code> می کنیم. برای حالت عمودی دستور ثابت <code>SwingConstants.VERTICAL</code>

	را جایگزین پارامتر orientation می کنیم.
<code>void setValue(int value)</code>	برای مقادری که در ProgressBar استفاده می شود.

مثال:

```
package javalike;

import javax.swing.*;

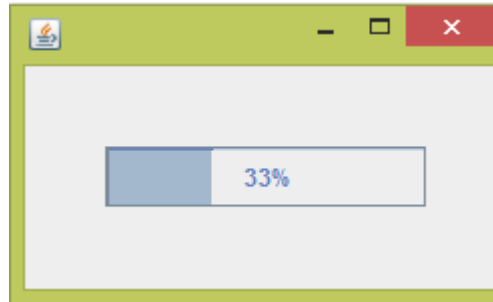
public class ProgressBarExample extends JFrame {
    JProgressBar jb;
    int i = 0;

    ProgressBarExample() {
        jb = new JProgressBar(0, 2000);
        jb.setBounds(40, 40, 160, 30);
        jb.setValue(0);
        jb.setStringPainted(true);
        add(jb);
        setSize(250, 150);
        setLayout(null);
    }

    public void iterate() {
        while (i <= 2000) {
            jb.setValue(i);
            i = i + 20;
            try {
                Thread.sleep(150);
            } catch (Exception e) {
            }
        }
    }

    public static void main(String[] args) {
        ProgressBarExample m = new ProgressBarExample();
        m.setVisible(true);
        m.iterate();
    }
}
```

خروجی: بعد از اجرای برنامه خروجی بصورت تصویر (۲) خواهد بود:



تصویر (۲)

در تصویر (۲) ProgressBar از صفر شروع می شود و تا صد که پایان روند پیشرفت کار می باشد متوقف می شود.

توضیحات:

```
JProgressBar jb;
```

- یک شی از نوع JProgressBar تعریف می کنیم.

```
int i = 0
```

- متغیر **i** نقش شمارنده ما را بازی می کند.

```
jb = new JProgressBar(0, 2000);
```

- ایجاد یک شی از کلاس JProgressBar با مقدار minimum عدد صفر و مقدار maximum عدد دو هزار
- این تعیین مقدار یعنی ProgressBar ما تنها بین ۰ تا ۲۰۰۰ می تواند مقدار بگیرد.

```
jb.setBounds(40, 40, 160, 30);
```

- تعیین مختصات و ابعاد ProgressBar ما در فریم.

```
1. add(jb);
2. setSize(250, 150);
3. setLayout(null);
```

۱. اضافه کردن شی ProgressBar خود به فریم برنامه

۲. تعیین اندازه فریم برنامه

۳. چون از طرح بندی خاصی در برنامه استفاده نکردیم مقدار این متد را `null` قرار می دهیم.

```
public class ProgressBarExample extends JFrame {
```

- چون کلاس ما کلاس `JFrame` را به ارث برده است دیگر نیازی به شی سازی از کلاس `JFrame` در برنامه خود نداریم به عبارتی کلاس ما خود یک `JFrame` حساب می شود چون فرزند کلاس `JFrame` هستش و تمام ویژگی ها و متدهای کلاس `JFrame` را به ارث برده و به آنها دسترسی دارد.

```
public void iterate() {
    while (i <= 2000) {
        jb.setValue(i);
        i = i + 20;
        try {
            Thread.sleep(150);
        } catch (Exception e) {
        }
    }
}
```

- این متد را برای تعیین روند پیشرفت کار در `ProgressBar` ایجاد کرده ایم.
- همان طور که ابتدا گفتیم متد `setValue` جریان پیشرفت کار در `ProgressBar` را مقداردهی می کند.
- در اینجا یک شمارنده `i` داریم که از صفر شروع می شود و ۲۰ تا ۲۰ به مقدار آن اضافه می شود تا این که مقدار آن به ۲۰۰۰ برسد.

```
i=0,20,40,60,80,100,120,.....,2000
```

```
jb.setValue(i);
```

- هر بار که حلقه تکرار می شود مقدار شمارنده `i` وارد متد `setValue` می کنیم.
- ابتدای کار `minimum` مقدار `ProgressBar` خود را `0` و `maximum` مقدار `ProgressBar` خود را `2000` قرار دادیم. این یعنی بازه مقدارهایی که می توانیم به `ProgressBar` خود دهیم بین `0` تا `2000` می باشد. حالا اگر شمارنده خود را طوری تنظیم کنیم که مقدار بین `0` تا `2000` را بشمارد و وارد متد `setValue` کند ، جریان پیشرفت کار بین `0%` تا `100%` خواهد بود. چرا که از صفر تا `2000` را شمارنده ما پیموده و وارد متد `setValue` کرده است. حال

اگر شمارنده ما صفر تا ۱۰۰۰ را بییماید و وارد متد `setValue` کند جریان پیشرفت کار `ProgressBar` ما بین ۰ تا ۵۰ درصد خواهد بود چرا که تنها نصف بازه تعیین شده را شمارش و پیموده است.

- پس تا اینجا فهمیدیم درصد پیشرفت جریان کار یک `ProgressBar` به شمارش شمارنده ما در بازه بین `minimum` تا `maximum` مقدار تعیین شده برای `ProgressBar` ما بستگی دارد، یعنی اگر مقدار شمارنده ما از `minimum` شروع شود و به مقدار آن در نهایت به `maximum` برسد یک فرآیند ۰ تا ۱۰۰ درصدی در `ProgressBar` خود شاهد خواهیم بود.

```
int i = 0;
```

- مقدار شمارنده `i` وقتی وارد حلقه می شویم در ابتدا صفر است. از طرفی مقدار `minimum` ما نیز صفر می باشد. این یعنی جریان روند `ProgressBar` ما از صفر درصد شروع می شود. چرا که ابتدا مقدار شمارنده و مقدار `minimum` ما برابر بوده است.

```
while (i <= 2000) {
```

- از طرفی دیگر شرط حلقه ما این است که تا هنگامی که مقدار شمارنده `i` برابر ۲۰۰۰ نشده است دستورات درون بدنه حلقه را تکرار کن. این یعنی مقدار پیشرفت جریان `ProgressBar` تا ۱۰۰ درصد خواهد بود چرا که مقدار شمارنده `i` در نهایت به ۲۰۰۰ که برابر مقدار `maximum` ما می باشد طبق شرط حلقه خواهد رسید. اگر شرط به جای ۲۰۰۰ عدد ۱۰۰۰ بود مقدار پیشرفت جریان `ProgressBar` ما ۵۰ درصد می شد چرا که ما نصف کل بازه ۰ تا ۲۰۰۰ که ابتدا برای `ProgressBar` خود تعیین کرده ایم را پیموده ایم

```
i = i + 20;
```

- مقدار شمارنده `i` به صورت ۲۰ تا ۲۰ تا افزایش می یابد.
 - تا اینجا از روند کلی که `ProgressBar` ما پیشرفت می کند آشنا شدیم. یعنی دوستیم که اگر شمارنده بازه `min` تا `max` مقدار `ProgressBar` را بییماید یک جریان صفر تا ۱۰۰ درصد خواهیم داشت. خب سوالی که پیش میاد اینه که هنگام نمایش `ProgressBar` ما اعدادی بین صفر تا صد نمایش داده می شود، که گاهی این اعداد یکی یکی و گاهی دو تا دو تا و یا ۱۰ تا ۱۰ تا تغییر میکنند تا این که به عدد ۱۰۰ برسند، میخوایم بدونیم چطور این اتفاق می افتد؟
 - برای این که بدانیم جریان پیشرفت کار که از صفر درصد شروع می شود و به صد در صد خاتمه پیدا میکند، چندتا چندتا مقدار آن تغییر می کند از فرمول زیر استفاده می کنیم:
- `maximum` بیشترین مقداری که می توانیم به `ProgressBar` خود دهیم.

VC مقداری که هر بار به شمارنده اضافه می شود مثلا در مثال بالا این مقدار ۲۰ است.

$$\frac{\text{maximum}}{vc} = S$$

$$\frac{100}{s} = C$$

C تا C تا مقدار جریان پیشرفت ProgressBar ما تغییر خواهد کرد.

مثلا با توجه به مثال بالا:

```
maximum=2000;
```

```
vc=20;
```

$$\frac{\text{maximum}}{vc} = S;$$

$$\frac{2000}{20} = 100$$

```
s=100;
```

$$\frac{100}{s} = C;$$

$$\frac{100}{100} = 1$$

```
c=1;
```

- پس یکی یکی به مقدار جریان پیشرفت کار اضافه می شود تا این که به صد برسیم.
- ابتدا مقدار جریان پیشرفت کار بین ۰ تا ۱۰۰ درصد بود چرا که شمارنده از صفر تا ۲۰۰۰ را می پیمود. با محاسبه بالا می فهمیم که نمایش جریان پیشرفت بصورت زیر خواهد بود:

```
0% 1% 2% 3% 4% 5% 6% .....30% 31% 32% .....98% 99% 100%
```

همان طور که می بینید یکی یکی به مقدار جریان پیشرفت ProgressBar ما افزوده می شود تا به ۱۰۰ درصد برسد.

دوستان این فرمول الکی ساختم!!! تنها برای درک بهتر برای تعیین مقدار پیشرفت کار یک ProgressBar اگه گیج تون میکنه کلا بیخیال توضیحات بشید و مثال رو چندبار ببینید و برای خودتون مقادیر رو تغییر بدید راحت روش کار دستتون میاد.

حال اگر شمارنده ما به جای دستور زیر

```
i = i + 20;
```

بصورت زیر تغییر کند :

```
i = i + 40;
```

```
maximum=2000;
```

```
vc=40;
```

```
maximum  
vc = s;
```

```
2000  
40 = 50;
```

```
s=50;
```

```
100  
s = c;
```

```
100  
50 = 2;
```

```
c=2;
```

با محاسبه بالا می فهمیم که نمایش جریان پیشرفت بصورت زیر خواهد بود:

```
0% 2% 4% 6% 8% 10% 12% .....30% 32% 34% 36% .....94% 96% 98% 100%
```

همان طور که می بینید **دو تا دو تا** به مقدار جریان پیشرفت ProgressBar ما افزوده می شود تا به ۱۰۰ درصد برسد.

بازم تکرار میکنم اگر توضیحات بالا گیج کننده بود اصلا نگران نباشید! توضیحات رو بی خیال بشید و سراغ مثال و تغییر مقادیر

مثال بروید یواش یواش دستتون میاد!

```
try {  
    Thread.sleep(150);  
} catch (Exception e) {  
}
```

- با هر بار تغییر شمارنده یک مکث ۱۵۰ میلی ثانیه انجام می دهیم، چرا؟! تا نمایش جریان پیشرفت ProgressBar را بتونیم ببینیم بدون مکث کردن آنقدر سریع انجام میشه که دیگه تا برنامه رو اجرا کردید می بینید که پیشرفت کار ۱۰۰ درصد شده است!!

```
1. ProgressBarExample m = new ProgressBarExample();
2. m.setVisible(true);
3. m.iterate();
```

۱. از کلاس خود شی ایجاد کرده ایم.
۲. با شی ایجاد شده از کلاس خود این متد را صدا زده ایم. این متد باعث میشه تمام اجزای گرافیکی ما نمایش داده شود.
۳. خب برای اجرای روند پیشرفت کار این متد را صدا زده ایم.

- خب میدونم شاید گیج شده اید!!! اون توضیحات رو برای روشن شدن مفاهیم مثال ها گفتم اگر اوضاع رو بهتر نکرده فقط مثال ، مثال و مثال ببینید و کار 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 جاوالایک به دور از اخلاق حرفه ای ست و حرام می باشد.