آموزش زبان برنامه نویسی جاوا

گرافیک در جاوا – پکیج Swing

جلسه یازدهم

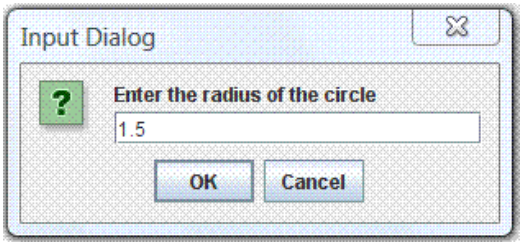
کلاس JOptionPane

نویسنده : رحمان زارعی

جاوا را ساده،آسان و شیرین بنوشید!!!!

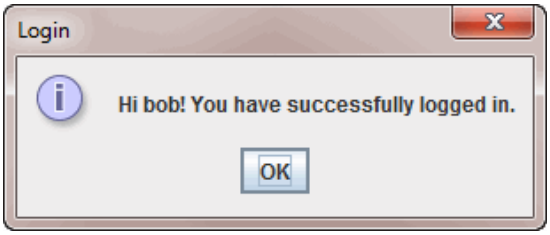
کلاس JOptionPane یک جعبه محاوره ای استاندارد را فراهم کرده است. JOptionPane برای نمایش اطلاعات،دریافت ورودی از کاربر،نمایش پیام تایید استفاده می شود.

در کل هنگام استفاده از JOptionPane یک پنجره کوچک ظاهر می شود که اطلاعاتی را نمایش یا در خواست ورودی و تایید از کاربر می کند. از همه این توضیحات بگذریم خودم میدونم با این توضیحات مفهومی روشن نمی شود!!! پس برای درک بهتر ازکاربرد این اجزای گرافیکی تصویر(۱) ، تصویر(۲) و تصویر(۳) را مشاهده کنید:



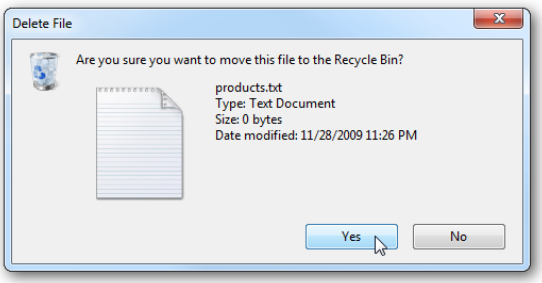
تصویر(۱)

- در تصویر(۱) از JOptionPane برای دریافت ورودی از کاربر استفاده کرده ایم.



تصویر(۲)

- در تصویر(۲) از JOptionPane برای نمایش پیام به کاربر استفاده کرده ایم.



تصویر(۳)

- در تصویر(۳) از JOptionPane برای نمایش تایید یا رد کردن عملی خاص توسط کاربر استفاده کرده ایم که به اصطلاح به آن confirm کردن می گویند.

• سازنده های پرکاربرد کلاس JOptionPane :

کاربرد	سازنده
برای ایجاد یک JOptionPane استفاده می شود.	`JOptionPane()`
ایجاد یک JOptionPane برای نمایش پیام مشخص	`JOptionPane(Object message)`
ایجاد یک JOptionPane برای نمایش پیام همراه با مشخص کردن نوع پیامی که قراره به کاربر نمایش داده شود.	`JOptionPane(Object message, int messageType`

• متدهای پرکاربرد کلاس JOptionPane :

متد	`static void showMessageDialog(Component parentComponent, Object message)`
کاربرد	این متد برای نمایش پیام حاوی اطلاعات استفاده می شود. چیزی شبیه تصویر(۲)

متد	`static void showMessageDialog(Component parentComponent, Object message, String title, int messageType)`
کاربرد	این متد برای ایجاد یک جعبه پیام همراه با مشخص کردن عنوان و نوع پیامی که قراره نمایش داده شود.

متد	`static int showConfirmDialog(Component parentComponent, Object message)`
کاربرد	این متد برای ایجاد یک جعبه پیام همراه با گزینه های Yes، No و Cancel همچنین مشخص کردن عنوان و انتخاب یک گزینه چیزی شبیه تصویر(۳)

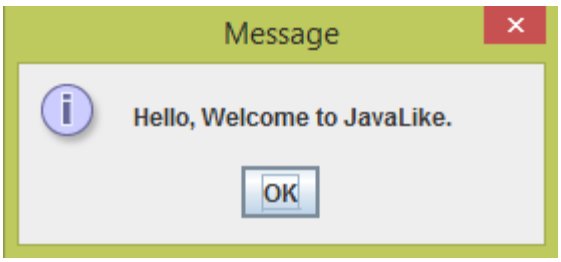
متد	`static String showInputDialog(Component parentComponent, Object message)`
کاربرد	این متد برای نمایش یک پیام سوالی و درخواست ورودی از کاربر می باشد. تصویر(۱)

اصلا نگران نباشید! هیچ کس از با خوندن توضیحات و تئوری برنامه نویس نشده است!!! تنها با مثال و تمرین عملی است که می توان خودمون رو به چالش بکشیم و یادبگیریم!! پس به مثال های زیر توجه کنید:

مثال JOptionPane در جاوا:

```
package javalike;

import javax.swing.*;

public class OptionPaneExample {
	JFrame f;

	OptionPaneExample() {
		f = new JFrame();
		JOptionPane.showMessageDialog(f, "Hello, Welcome to JavaLike.");
	}

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

خروجی: تصویر(۴)

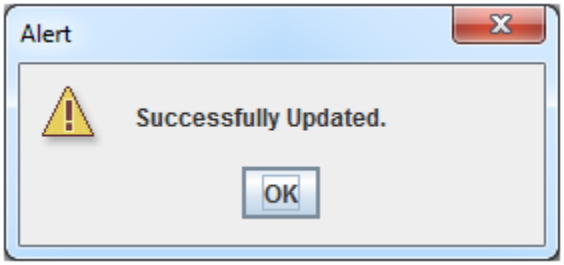


تصویر(۴)

مثال:

```
package javalike;
import javax.swing.*;
public class OptionPaneExample2 {
JFrame f;
OptionPaneExample2(){
    f=new JFrame();
    JOptionPane.showMessageDialog(f,"Successfully
Updated.","Alert",JOptionPane.WARNING_MESSAGE);
}
public static void main(String[] args) {
    new OptionPaneExample2();
}
}
```

خروجی: تصویر(۵)



تصویر(۵)

```
JOptionPane.showMessageDialog(f,"Successfully
Updated.","Alert",JOptionPane.WARNING_MESSAGE);
```

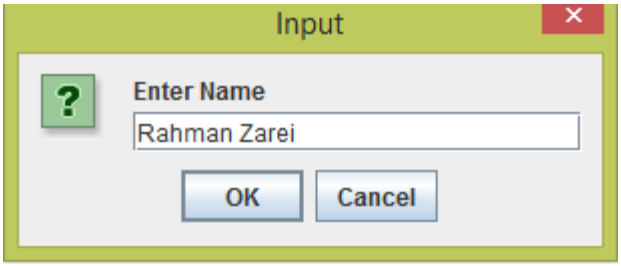
- متد بالا همان متد زیر است که مقدار دهی شده است:

```
showMessageDialog(Component arg0, Object arg1, String arg2,int arg3)
```

- که بجای **arg0** شی از نوع کلاس **JFrame** که فرزند کلاس **Component** هستش قرار داده ایم.
- بجای **arg1** یک رشته **String** گذاشته! چرا ؟ چون کلاس **Object** پدربزرگ همه کلاس هاست! پس مجاز هستیم از کلاس **String** استفاده کنیم.
- بجای **arg2** هم یک متن رشته قرار داده ایم.
- بجای **arg3** نوع پیامی که قراره برای کاربر نمایش داده شود را تعیین کردیم.در اینجا نوع **JOptionPane.WARNING_MESSAGE** را انتخاب کردیم.این نوع برای پیام هایی که جنبه هشدار دارند استفاده می شود.

```
package javalike;
import javax.swing.*;
public class OptionPaneExample3 {
JFrame f;
OptionPaneExample3(){
    f=new JFrame();
    String name=JOptionPane.showInputDialog(f,"Enter Name");
}
public static void main(String[] args) {
    new OptionPaneExample3();
}
}
```

خروجی: تصویر(۶)



تصویر(۶)

JOptionPane.*showInputDialog*

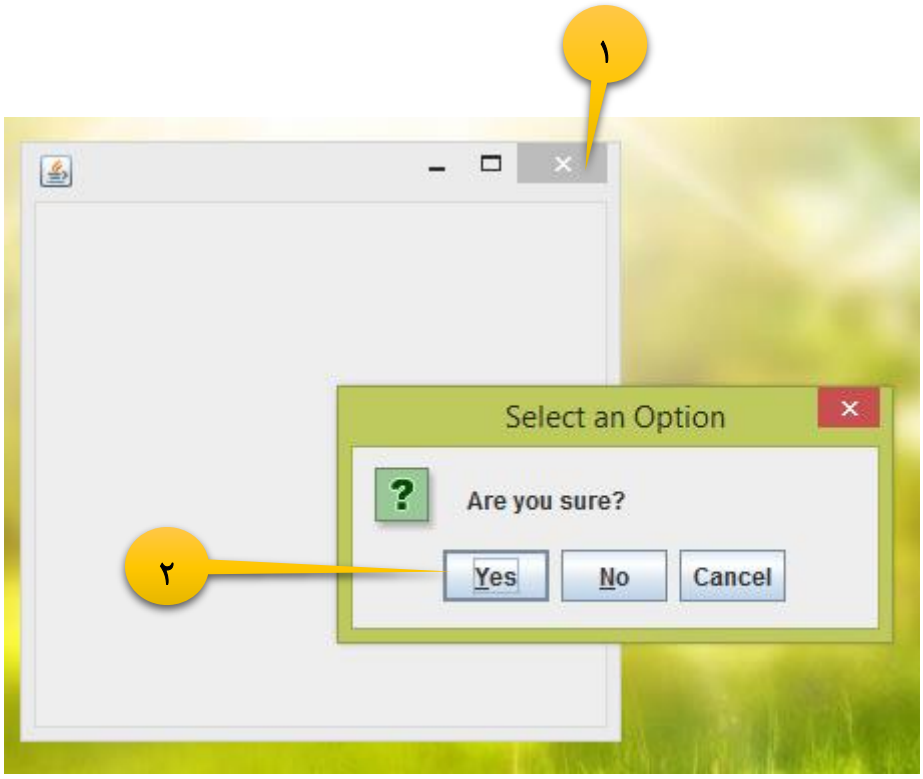
- همان طور که ابتدای جلسه هم توضیح دادیم این متد برای دریافت ورودی از کاربر استفاده می شود.

مثال:

```
package javalike;

import javax.swing.*;
import java.awt.event.*;

public class OptionPaneExample extends WindowAdapter {
     JFrame f;

     OptionPaneExample() {
          f = new JFrame();
          f.addWindowListener(this);
```

```
            f.setSize(300, 300);
            f.setLayout(null);
            f.setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
            f.setVisible(true);
      }

      public void windowClosing(WindowEvent e) {
            int a = JOptionPane.showConfirmDialog(f, "Are you sure?");
            if (a == JOptionPane.YES_OPTION) {
                  f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            }
      }

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

خروجی: تصویر(۷)



تصویر(۷)

- طبق تصویر(۷) اگر ما دکمه بستن پنجره فریم یا همون Close که به شکل یک ضربدر هستش را انتخاب کنیم، پیامی برای ما نمایش داده می شود که نیاز به تایید کردن برای بستن پنجره برنامه دارد.در صورت فشردن دکمه Yes برنامه بسته می شود.

```
f.setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
```

- متد setDefaultCloseOperation برای تعیین حالت پیشفرض دکمه close یا همان ضربدر قرمز گوشه برنامه می باشد.اگر پارامتر این متد را JFrame.DO_NOTHING_ON_CLOSE قرار دادیم، با زدن دکمه close یا همون ضربدر قرمز گوشه برنامه دیگر برنامه بسته نمی شود.
- چون که قراره پیام تایید یا رد بسته شدن پنجره برای برنامه قرار دهیم از این دستور استفاده کرده ایم.

```
public void windowClosing(WindowEvent e) {
          int a = JOptionPane.showConfirmDialog(f, "Are you sure?");
          if (a == JOptionPane.YES_OPTION) {
               f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
          }
     }
```

- متد windowClosing برای بستن برنامه کاربرد دارد.این متد رویدادهای مربوط به ویندوز را دریافت میکند.
- متد windowClosing در کلاس WindowAdapter قرار دارد.
- کلاس WindowAdapter یک کلاس انتزاعی (abstract) می باشد که برای دریافت رویدادهای (events) مربوط به ویندوز استفاده می شود.
- خب ما برای استفاده از متد windowClosing در برنامه خود باید کلاس WindowAdapter را به ارث ببریم :

```
public class OptionPaneExample extends WindowAdapter {
```

- خب بعد از به ارث بردن کلاس WindowAdapter ، متد windowClosing را در بدنه کلاس خود override یا بازنویسی می کنیم.

```
int a = JOptionPane.showConfirmDialog(f, "Are you sure?");
```

- این متد پیام تاییدکردن برای کاربر را نمایش می دهد. مقدار نوع دکمه ای که کاربر انتخاب کرده است (این مقدار بصورت یک عدد صحیح می باشد) را درون متغیر a می ریزیم.

```
if (a == JOptionPane.YES_OPTION) {
```

- اگر مقدار a (نوع دکمه ای که کاربر انتخاب کرده است) برابر گزینه Yes بود، دستور زیر را اجرا کن:

```
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

- این دستور حالت پیشفرض دکمه close یا همون ضربدر قرمز گوشه فریم را به JFrame.EXIT_ON_CLOSE تغییر می دهد. یعنی پنجره یا همون فریم بسته می شود.

مطمئنا از کمبود منابع فارسی در زمینه جاوا شکایت دارد!! خب شما هم با حمایت کردن خودتون سهمی در رشد منابع آموزشی فارسی داشته باشید.

پیروز و موفق باشید

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