

## به نام خدا

### دریافت ورودی از صفحه کلید

در پایان این مقاله خواهید توانست تا با استفاده از دو کلاس زیر، ورودی های برنامه هایتان را از صفحه کلید بخوانید:

- **BufferedReader**
- **JOptionPane**

✓ استفاده از کلاس **BufferedReader**:

با استفاده از این کلاس قادر خواهید بود تا از طریق **console** ورودی برنامه تان را از صفحه کلید بخوانید. به این منظور لازم است مراحلی که در ادامه به آن می پردازیم را طی کنید:

کلاس مذکور در پکیج **java.io** قابل دسترسی است. بنابراین لازم است تا در ابتدای برنامه ی خود خطوط زیر را بیفزائید:

```
import java.io.BufferedReader;  
import java.io.InputStreamReader;  
import java.io.IOException;
```

البته اگر به صورت زیر نیز این کار را انجام دهیم، کلاس های لازم، در دسترس خواهند بود:

```
import java.io.*;
```

در حقیقت در روش دوم تمام کلاس های موجود در پکیج **java.io** را در برنامه ی خود، در دسترس قرار می دهیم. به عنوان توضیحی در مورد پکیج ها باید

بگوئیم که در حقیقت پکیج ها شامل یکسری کلاس های از پیش تعریف شده هستند که اهدافی مرتبط بایکدیگر دارند.

این قطعه کد را در بلاک برنامه ی خود قرار دهید:

```
BufferedReader dataIn = new BufferedReader( new  
InputStreamReader(System.in) );
```

برای درک بهتر، لازم است که خودتان را با جزئیات مسئله درگیر نکنید و فقط به انجام آنچه گفته می شود، بسنده کنید. مطمئن باشید که در مقاله های بعدی به سایر مباحث پرداخته می شود. در این جا فقط کافایت بدانید که در حقیقت با عمل مذکور، یک شیء به نام `dataIn` ایجاد کرده اید.

برای اعلان یک متغیر رشته ای جهت گرفتن ورودی از صفحه کلید و نیز فراخوانی متد `readLn()`، باید قطعه کد زیر را عینا در برنامه ی خود وارد کنید: (لازم به ذکر است که مفهوم متد را می توان معادل مفهوم تابع در `C++` در نظر بگیریم)

```
try{  
    String temp = dataIn.readLine();  
}catch( IOException e ){  
    System.out.println("Error in getting input");  
}
```

حالا به مثال زیر توجه کنید که در آن یک برنامه ی نمونه آورده شده است:

```

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.IOException;

public class GetInputFromKeyboard {

    public static void main( String[] args ){
        BufferedReader dataIn = new BufferedReader(new
            InputStreamReader( System.in) );

        String name = "";
        System.out.print("Please Enter Your Name:");
        try{
            name = dataIn.readLine();
        }catch( IOException e ){
            System.out.println("Error!");
        }
        System.out.println("Hello " + name + "!");
    }
}

```



در مورد سه خط اول از برنامه ی فوق، باید گفت که در حقیقت با افزودن این سه دستور، اعلام می کنیم که می خواهیم سه کلاس `BufferedReader` و `InputStreamReader` و نیز `IOException` را از پکیج `java.io` در برنامه ی خود `import` کنیم. در حقیقت با این کار، سه کلاس مذکور را در دسترس خود قرار می دهیم تا بتوانیم در طول برنامه مان از آن ها استفاده کنیم. در اینجا مجددا یاد آوری می کنیم که می توانیم به جای سه خط مذکور، بنویسیم:

`import java.io.*;` که در حقیقت با این کار تمام کلاس های موجود در این پکیج را در برنامه ی خود، قابل دسترس کرده ایم.

در مورد مفهوم پکیج هم که در همین مقاله، توضیح داده شد.

عبارت `public class GetInputFromKeyboard{` به این معنا است که کلاسی را با نام `GetInputFromKeyboard` ایجاد کرده ایم. ( مجدداً به نحوه ی نامگذاری این کلاس و چگونگی استفاده از حروف بزرگ و کوچک در این نامگذاری دقت کنید و قوانین و قرار دادهایی را که در مورد نامگذاری شناسه ها ، کلاس ها و متدها در مقاله های قبلی عنوان کردیم را به یاد بیاورید.)

عبارت بعدی معرف متد اصلی برنامه است:

```
public static void main(String[] args){
```

عبارت زیر ، متغیری به نام `name` و از نوع رشته یا همان `String` معرفی کرده و به آن مقدار اولیه ی تهی می دهد (یک رشته ی خالی ایجاد می کند):

```
String name="";
```

عبارت زیر پیغام مناسبی را در خروجی برنامه چاپ می کند که طی آن از کاربر خواسته می شود که نامی را وارد کند(که این نام در حقیقت همان متغیر از نوع رشته است):

```
System.out.print("Please Enter Your Name:");
```

قسمت دیگری از برنامه ی فوق که احتیاج به توضیح دارد، بلوک زیر است که به بحث مدیریت استثنا ها در برنامه ها مربوط است و فعلاً به آن نمی پردازیم. فقط به این

مقدار بسنده می کنیم که با افزودن این بلاک به برنامه ی خود ، این مدیریت را به برنامه ی خود افزوده اید که اگر کاربر، مقدار غیر معتبری را وارد کرد، برنامه این مقدار را به عنوان ورودی قبول نکند. در حقیقت ویژگی این کار اینست که باعث می شود تا از پردازش های اشتباه و نیز ثبت خروجی های ناخواسته ، جلوگیری شود:

```
try{
    name = dataIn.readLine();
}catch( IOException e ){
    System.out.println("Error!");
}
```

آخرین قسمتی که در برنامه ی فوق احتیاج به توضیح دارد، متد `dataIn.ReadLine();` است که یک مقدار رشته ای را از کاربر گرفته و آن را در متغیری که با نام `name` تعریف کرده بودیم، ذخیره می کند.

### ✓ استفاده از کلاس **JOptionPane** :

این کلاس برای گرفتن ورودی از صفحه کلید، از یک واسط گرافیکی استفاده می کند. `JOptionPane` این امکان را به ما می دهد تا یک پنجره ی گرافیکی را به نمایش در بیاوریم و از این طریق این امکان را داشته باشیم که از کاربر بخواهیم تا یک ورودی برای برنامه وارد کند.

برای درک بهتر آنچه که گفته می شود، به نمونه برنامه ی زیر توجه کنید:

```

1 import javax.swing.JOptionPane;
2
3 public class GetInputFromKeyboard {
4
5     public static void main( String[] args ){
6         String name = "";
7         name=JOptionPane.showInputDialog("Please enter your name");
8         String msg = "Hello " + name + "!";
9         JOptionPane.showMessageDialog(null, msg);
10    }
11}

```

در خط اول این برنامه همانطوری که در مثال قبلی گفته شد، کلاس **JOptionPane**

از پکیج **javax.swing** را قابل دسترسی در برنامه قرار می دهیم.

حالا تنها قسمتی که لازم به توضیح دارد، عبارت هایی هستند که تاکنون مثل آن ها را در

مقاله های پیشین نیاورده بودیم. عبارت اولی که در خط شماره ی ۷ از تصویر بالا قابل

رویت است ، یک **dialog box** را در خروجی به کاربر نمایش می دهد و طی آن با

اعلام پیغام **Please enter your name** از کاربر می خواهد تا یک رشته را به

عنوان ورودی به برنامه اعلام کند. در عبارت دوم نیز که در خط شماره ی ۹ دیده می

شود، یک **dialog box** دیگر به کاربر نشان داده می شود که محتویات متغیر **msg** را

در خود جای می دهد. کلمه ی **null** که در همین خط آمده است ، باعث می شود که

**dialog box** ظاهر شده فقط دکمه ی **OK** داشته باشد.

برای درک بهتر از نحوه ی کارکرد برنامه ی مذکور و چگونگی طرز کار کلاس

**JOptionPane** ، فرض کنید که برنامه ی فوق را اجرا کرده ایم. ابتدا با تصویر زیر

روبه رو می شویم:



حالا فرض کنیم که عبارت **florence** را در نوار خالی وارد کنیم:



به عنوان آخرین **dialog box** ، نمای زیر را خواهیم داشت:



\*\*\*

تهیه شده در [www.java4every1.wordpress.com](http://www.java4every1.wordpress.com)

ایمیل تماس با مدیریت وبلاگ: [blogsofmine@gmail.com](mailto:blogsofmine@gmail.com)

مدیریت وبلاگ با کمال میل پذیرای انتقادات و پیشنهادات شما عزیزان برای هر چه بهتر برگزار شدن فرآیند آموزش در وبلاگ می باشد.