

به نام خدا

آموزش یازدهم

در طراحی برنامه های کاربردی زمان های بسیاری برای ما پیش خواهد آمد که نیاز داریم تا کاربر بخشی از اطلاعات ورودی را از طریق صفحه کلید وارد برنامه کند تا بر اساس اطلاعات ورودی، برنامه طراحی شده نتیجه مرتبط را به کاربر اعلام کند.

برای این منظور می بایست با کلاس Scanner در زبان برنامه نویسی جاوا آشنا شویم. در حقیقت این کلاس در API زبان برنامه نویسی جاوا طراحی شده است و ما صرفاً با Import کردن این کلاس و همچنین ساخت یک شیء از روی آن می توانیم از قابلیت های آن استفاده کنیم. برای روشن شدن این مطلب پروژه جدید در اکیپس تحت عنوان 11th Session به معنی جلسه یازدهم ایجاد می کنیم. سپس کلاسی تحت عنوان Project به معنی پروژه در آن ایجاد می کنیم. (کلیه نام ها اختیاری می باشند). در ابتدا برنامه ما می بایست به شکل زیر باشد:

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

در حقیقت از آنجا که نیاز داریم تا برنامه ما یک نقطه شروع داشته باشد، همانطور که در جلسات گذشته توضیح داده شد در حین ایجاد کلاس گزینه public static void main را تیک می زنیم. حال ما نیاز داریم تا این امکان را به کاربر خود بدهیم تا بتواند از طریق صفحه کلید اطلاعات را وارد برنامه کند، از اینرو کلاس Scanner را وارد برنامه خود می کنیم و نامی به دلخواه همچون myScanner برای آن در نظر می گیریم. به عکس زیر توجه کنید:

```
*Project.java X
1
2 class Project {
3
4     public static void main(String[] args) {
5         Scanner myScanner = new Scanner(System.in);
6
7     }
8 }
```

محیط برنامه نویسی اklipsis با قرار دادن نقطه چین دوره واژه های Scanner و همچنین قرار دادن یک علامت ستاره کنار نام فایل ما همانطور که در تصویر فوق مشاهده می شود به ما اخطار می دهد که مشکلی در برنامه وجود دارد. به منظور یافتن راه حل موس خود را روی علامت نقطه چین نگه می داریم و صفحه ای همچون تصویر زیر مشاهده خواهیم کرد:

```
*Project.java X
1
2 class Project {
3
4     public static void main(String[] args) {
5         Scanner myScanner = new Scanner(System.in);
6
7     }
8 }
9
```

Scanner cannot be resolved to a type
7 quick fixes available:
- Import 'Scanner' (java.util)
+ Create class 'Scanner'
+ Create interface 'Scanner'
+ Change to 'Signer' (java.security)
+ Create enum 'Scanner'
+ Add type parameter 'Scanner' to 'main(String[])'
+ Fix project setup...

در حقیقت محیط برنامه نویسی اklipsis چندین پیشنهاد برای رفع مشکل به ما می دهد که ما به عنوان برنامه نویس نیاز داریم تا تک تک موارد پیشنهادی را مورد بررسی قرار دهیم. حال از آنجا که می بایست مد نظر داشته باشیم که هر وقت یک کلاس تعبیه شده در جاوا را در برنامه خود مورد استفاده قرار می دهیم حتما بایستی آنرا Import یا وارد برنامه خود کنیم، به راحتی اولین گزینه که 'Scanner' Import به معنی کلاس Scanner را وارد برنامه کن است را کلیک می کنیم. حال کدی به شکل زیر خواهیم داشت:

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

```
import java.util.Scanner;
class Project {
    public static void main(String[] args) {
        Scanner myScanner = new Scanner(System.in);
    }
}
```

(لازم به ذکر است که به جای Import کردن توسط محیط برنامه نویسی اکتیپس، می توانیم خود کد های Import مورد نیاز را به صورت دستی وارد برنامه کرده و یا پس از استفاده از یکی از کلاس های API جاوا همچون کلاس Scanner کلیدهای Ctrl و Shift و حرف O را به طور هم زمان فشار دهیم که با این کار فرآیند Import انجام خواهد شد).

پس از Import کردن همانطور که در کد فوق مشاهده می شود یک خط کد import java.util.Scanner; به ابتدای کد ما اضافه می شود. از این پس بعد هر چند بار که نیاز داشته باشیم می توانیم بدون هیچ مشکلی از روی کلاس Scanner یک شیء جدید ساخته و آن را مورد استفاده قرار دهیم (نحوه ساخت اشیاء در زبان جاوا به طور مفصل در آموزش سی و یکم مورد بررسی قرار خواهد گرفت).

واژه Scanner نشانگر یک Class در جاوا است که از پیش نوشته شده است و ما صرفاً آن را وارد برنامه خود می کنیم. واژه myScanner یک نام است به معنی **کلاس اسکنر من** که ما به صورت دلخواه برای نام شیء ساخته شده از روی کلاس Scanner در نظر می گیریم. پس از علامت = با نوشتن واژه new ما از اکتیپس می خواهیم که یک شیء جدید از کلاس Scanner تحت عنوان myScanner برای ما ایجاد کند. حال پارامترهایی که داخل پرانتز قرار دارند عبارتند از کلید واژه System که حاکی از کلاسی است که این امکان را به جاوا می دهد که هرآنچه خارج از اختیار JVM بود وارد برنامه شود و کلید واژه in که موجب می گردد ورودی کامپیوتر یا لپ تاپ ما مورد استفاده این کلاس قرار گیرد که این ورودی همان صفحه کلید می باشد.

حال می توانیم برنامه خود را تست کنیم. برای این منظور کد خود را به شکل زیر تکمیل می کنیم:

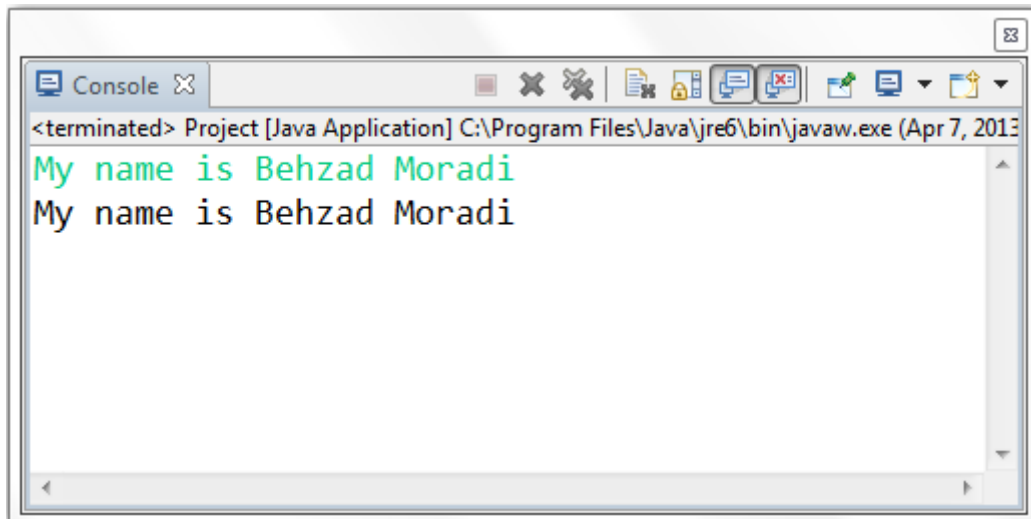
```
import java.util.Scanner;

class Project {

    public static void main(String[] args) {
        Scanner myScanner = new Scanner(System.in);
        System.out.println(myScanner.nextLine());
    }
}
```

در حقیقت با این Statement جدید که وارد برنامه خود کردیم، قصد داریم هر آنچه را که از طریق صفحه کلید وارد می کنیم برای ما مجدداً نمایش داده شود. داخل دستور `System.out.println()` ما نام شیئی که از روی کلاس `Scanner` ایجاد کردیم را نوشته سپس متدی تحت عنوان `nextLine()` ضمیمه آن می کنیم. به طور خلاصه کار این متد آن است که هر نوع ورودی از طریق صفحه کلید را وارد برنامه کند. حال اگر بخواهیم ورودی خاصی از صفحه کلید را مد نظر داشته باشیم بایستی از متد مخصوص همان داده ورودی استفاده کنیم.

متد `nextLine()` حاکی از آن است که هر آنچه را وارد صفحه کلید کنیم به شکل یک `string` یا متن روی صفحه مانیتور نمایش داده شود. برای مثال برنامه را اجرا کرده، سپس در بخش `Console` نشانگر موس خود را مشاهده می کنیم که چشمک می زند به معنی اینکه می توانیم در آن چیزی وارد نماییم. حال جمله ای به دلخواه مثل `My name is Behzad` `Moradi` به معنی **نام من بهزاد مرادی هست** را وارد می کنیم سپس دکمه `Enter` را می زنیم و مشاهده می کنیم که دقیقاً همان چیزی را که از طریق صفحه کلید وارد برنامه کردیم را نمایش خواهد داد:



پس از آشنایی با یکی از Class های مفید در زبان برنامه نویسی جاوا، در قسمت آتی خواهیم دید که چگونه می توان نوع ورودی داده را برای این کلاس از پیش تعیین کرد به طوری که کاربر فقط و فقط اجازه داشته باشد همان داده ای که از وی خواسته شده است را وارد برنامه کند.