

به نام خدا

آموزش دهم

مطلب جدیدی که در این آموزش قرار است فرا بگیریم نحوه صحیح نامگذاری متغیر ها است. علاوه بر این تک تک اجرای تشکیل دهنده یک برنامه را مورد بررسی قرار خواهیم داد. از این رو اینکار را در قالب پروژه ای که قرار است بنویسیم انجام خواهیم داد.

در جلسه آموزشی نهم پروژه ای به صورت زیر تعریف کردیم:

فرض کنیم که در یک برج مسکونی زندگی می کنیم و علیرغم اینکه مدیر ساختمان از سکنه درخواست کرده است که بیش از ۱۰ نفر باهم سوار آسانسور نشوند باز هم مشاهده می کند که سکنه رعایت نمی کنند. با توجه به اینکه ایشان می دادند که ما یک برنامه نویس هستیم از ما درخواست می کند که یک برنامه برای ظرفیت مجاز آسانسور بنویسیم. ما هم از درخواست ایشان استقبال کرده و اقدام به نوشتن برنامه می کنیم!

برای نوشتن چنین برنامه ای اول یک پروژه جدید در اکیلیس ایجاد می کنیم و نام آن را 10th Session به معنی جلسه دهم می گذاریم. حال یک Class تحت عنوان ElevatorCapacity به معنی ظرفیت آسانسور ایجاد می کنیم. به خاطر داشته باشیم که در حین ایجاد Class گزینه public static void main(String[] args) را تیک بزنیم. حال می بایست کدی به شکل زیر داشته باشیم:

```
public class ElevatorCapacity {  
  
    /**  
     * @param args  
     */  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
    }  
}
```

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

حال اگر بخواهیم می توانیم Comment ها را حذف کنیم. در این برنامه ما یک Method از جنس main داریم که به منزله بخشی از برنامه ما است که برنامه از آن جا شروع می شود. اکنون به نظر می رسد که اگر بخواهیم که یک جمله روی صفحه مانیتور نمایش دهیم که عنوان برنامه ما را نمایش دهد با توجه به تمرین های پیشین هیچ مشکلی نداشته باشیم. برای این منظور کد فوق را به صورت زیر تکمیل می کنیم:

```
public class ElevatorCapacity {  
  
    public static void main(String[] args) {  
        // عنوان برنامه ظرفیت آسانسور  
        System.out.println("How many people should get on?");  
    }  
}
```

به خاطر داشته باشید که برای وارد کردن Comment به زبان فارسی پس از Save کردن برنامه پیغامی روی صفحه ظاهری می شود که در این مرحله می بایست گزینه Save as UTF-8 را انتخاب نماییم.

حال به بررسی تک تک کلید واژه هایی از زبان برنامه نویسی جاوا می پردازیم که بعد Comment در کد آمده اند. در حقیقت واژه System یکی از کلاس ها یا آجرهای سازنده API در زبان برنامه نویسی جاوا است. این برنامه داخلی جاوا به بخش هایی از کامپیوتر شما دسترسی پیدا خواهد کرد که خارج از کنترل مستقیم ماشین مجازی جاوا یا JVM است (این توضیحات فقط جهت آگاهی بیشتر آمده است و از برنامه نویسان مبتدی درخواست می شد خیلی ذهن خود را درگیر مسائلی از این دست نکنند). واژه out نشانگر جایی در برنامه اکلیپس است که متن را در آن به نمایش در می آورد. در واقع در محیط برنامه نویسی اکلیپس این مکان همان Console است که نمونه اجرا شده برنامه خود را می توانیم در آن مشاهده کنیم. متد print این دستور را به کامپیوتر می دهد که یک خط از کد را روی صفحه به نمایش در آورد. اهمیت بسیار زیاد آن در زبان برنامه نویسی جاوا، علت توضیح مجدد موضوعی که می خواهم هم اکنون عرض کنم. زبان برنامه نویسی جاوا به حروف بزرگ و کوچک حساس است از اینرو اگر واژه System را با حرف کوچک به

صورت system بنویسیم برنامه ما اجرا نخواهد شد. این مسئله در موارد مابقی کلمات Statement فوق نیز صدق می کند.

حال که با معنای تک تک کلید واژه های فوق آشنا شدیم بایستی عرض کنم که هر آنچه که در داخل پرانتز در مقابل متد println قرار گیرد روی صفحه مانیتور ما در حین اجرای برنامه چاپ خواهد شد. نکته ای که در اینجا می بایست مد نظر قرار دهیم این است که اگر بخواهیم یک جمله را روی صفحه مانیتور چاپ نماییم، در داخل پرانتز می بایست دو علامت “ “ قرار داده و متن خود را داخل آن تایپ کنیم اما در صورتی که بخواهیم فقط به طور مثال مقدار یک متغیر را روی صفحه به نمایش در آوریم می بایست صرفاً نام آن متغیر را درون پرانتز مقابل println بنویسیم که در این صورت هر آنچه مقداری که برای آن متغیر از پیش تعیین کرده باشیم روی صفحه نمایش داده خواهد شد.

حال پس از اجرای برنامه جمله How many people should get on? به معنای **چند نفر می بایست سوار شوند؟** روی صفحه مانیتور ظاهر شود. هم اکنون می بایست یک متغیر از جنس int برای ذخیره سازی وزن هر شخص که به طور متوسط ۷۵ کیلو است را تعریف نماییم. برای اینکار به صورت زیر عمل می کنیم:

```
int weightOfEachPerson = 75;
```

در واقع نام این متغیر را weightOfEachPerson به معنی **وزن هر فرد** می گذاریم و Value آن را ۷۵ کیلو در نظر می گیریم (توجه داشته باشیم که واحد کیلو فقط در ذهن ما به عنوان برنامه نویس است و هیچ جایی در int آن را تعریف نمی کنیم. از سوی دیگر فرض را در این برنامه بر آن می گذاریم که وزن افراد عدد اعشاری نخواهد داشت. اگر بخواهیم چنین امکانی را فراهم کنیم صرفاً نیاز است تا از متغیری از جنس double استفاده کنیم).

نحوه نام گذاری این متغیر هم به این شکل است که اولین کلمه که weight است با یک حرف کوچک شروع شده و حرف اول کلمات بعدی با حرف بزرگ شروع شده اند (در حقیقت شما هر نامی به هر شکلی را می توانید برای خود در نظر بگیرید اما نکته این جا است که این نوع نامگذاری یک روش رایج مابین برنامه نویسان حرفه ای است و به امید خدا چنانچه روزی شما هم به یک برنامه نویس حرفه ای جاوا و اندروید مبدل شدید و خواستید عضوی از یک تیم برنامه

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

نویسی شوید دیگر مشکلی در رابطه با نحوه نامگذاری نخواهید داشت. از سوی دیگر این روش نام گذاری که اصطلاحاً camelCase نامیده می شود منجر به راحت خواندن اسامی در محیط برنامه نویسی جاوا می شود زیرا اگر به طور مثال نام خود را به صورت عادی و به شکل weightofeachperson بنویسیم به دشواری معنای آن را متوجه خواهیم شد. در زبان انگلیسی واژه Camel به معنای **شتر** است و این سبک نوشتن به این خاطر به camelCase معرف شده است که نامی که ما در نظر می گیریم یک شمایی از کوهان شتر را خواهد داشت).

در مرحله بعد یک دیگر از جنس int برای ظرفیت آسانسور به شکل زیر تعریف می کنیم:

```
int elevatorCapacity = 750;
```

نام این را elevatorCapacity به معنی **ظرفیت آسانسور** با Value به میزان ۷۵۰ کیلو تعریف می کنیم. حال بایستی یک متغیر دیگر از جنس int ایجاد کنیم که در آن مقدار elevatorCapacity را بر مقدار weightOfEachPerson تقسیم کنیم. نام این متغیر آخر را numberOfPeople به معنی **تعداد افراد** می گذاریم و به صورت زیر آن را می نویسیم:

```
int numberOfPeople = elevatorCapacity / weightOfEachPerson;
```

در واقع در کد بالا میزان Value این متغیر را حاصل تقسیم ظرفیت آسانسور بر وزن هر شخص تعیین می کنیم. در مرحله آخر از نوشتن برنامه خود می بایست یک متغیر از جنس boolean تعریف کنیم که از طریق آن اجازه بسته شدن درب آسانسور را صادر خواهیم کرد. نام این متغیر را doorClosingOrder یا **فرمان بسته شدن در** می گذاریم و به شکل زیر آن را می نویسیم:

```
boolean doorClosingOrder = numberOfPeople > 10;
```

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

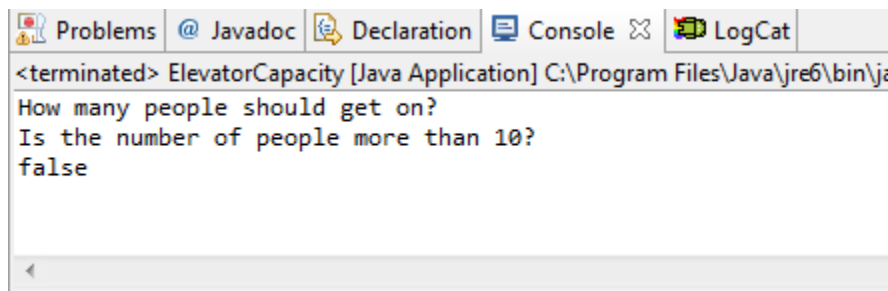
مدرس: بهزاد مرادی

به طور خلاصه ما به این متغیر دستور می دهیم که اگر مقدار numberOfPeople تقسیم مقدار elevatorCapacity بر weightOfEachPerson می باشد بیش از ۱۰ شود بایستی مقدار boolean که دو حالت بیشتر ندارد true شود یعنی تعداد افراد سوار شده در آسانسور بیش از ۱۰ نفر است و در غیر این صورت مقدار false در نظر گرفته شود به این معنی که تعداد افراد سوار شده در آسانسور کمتر از ۱۰ نفر است. حال کد خود را به شکل زیر تکمیل می کنیم:

```
public class ElevatorCapacity {

    public static void main(String[] args) {
        // عنوان برنامه ظرفیت آسانسور
        System.out.println("How many people should get on?");
        // وزن هر شخص
        int weightOfEachPerson = 75;
        // ظرفیت آسانسور
        int elevatorCapacity = 750;
        // تقسیم کردن ظرفیت آسانسور بر وزن هر شخص و اختصاص آن به یک متغیر
        int numberOfPeople = elevatorCapacity / weightOfEachPerson;
        // تعیین اینکه آیا ظرفیت بیش از 10 نفر است یا خیر؟
        boolean doorClosingOrder = numberOfPeople > 10;
        // نمایش جمله ای به معنای اینکه آیا تعداد افراد بیش از ده نفر است؟
        System.out.println("Is the number of people more than 10?");
        // نمایش نتیجه متغیر بولین
        System.out.println(doorClosingOrder);
    }
}
```

در نهایت جمله Is the number of people more than 10? را از طریق دستور System.out.println() روی صفحه نمایش می دهیم و پس از آن نتیجه متغیر boolean را در خط آخر کد خود روی صفحه نمایش می دهیم. حال برنامه خود را اجرا کرده و خروجی برنامه را روی صفحه مانیتور مشاهده می کنیم. همانطور که در تصویر زیر مشاهده می شود پس از اجرای برنامه در بخش Console کلمه false نمایش داده خواهد شد.



```
<terminated> ElevatorCapacity [Java Application] C:\Program Files\Java\jre6\bin\java
How many people should get on?
Is the number of people more than 10?
false
```

در واقع برای روشن تر شدن مطلب کد خود را تفسیر می کنیم. در ابتدا وزن هر فرد را به طور متوسط ۷۵ کیلو در نظر می گیریم و ظرفیت آسانسور را ۷۵۰ کیلو. سپس ظرفیت آسانسور را بر ۷۵ تقسیم می کنیم که عدد ۱۰ به دست می آید. حال در `boolean` محاسبه می کنیم که اگر مقدار `numberOfPeople` بیش از ۱۰ باشد مقدار این متغیر `true` شود و اگر کمتر از ۱۰ باشد مقدار این `false` شود. سپس جمله `Is the number of people more than 10?` به معنی **آیا تعداد افراد بیش از ۱۰ نفر است؟** را روی صفحه مانیتور نمایش می دهیم و بعد از آن نتیجه `boolean` خود را روی صفحه نمایش نشان می دهیم. در واقع با محاسباتی که انجام می دهیم در پاسخ به سوال فوق، برنامه نتیجه `false` به معنای نادرست یا خیر را به ما می دهد. پس حال که پاسخ به سوال فوق خیر است این اجازه به درب آسانسور داده می شود تا بسته شود. هم اکنون می توانیم برنامه نهایی را در اختیار مدیر ساختمان قرار داده تا با همکاری شرکت آسانسور سازی، آسانسور برج مسکونی را به آن مجهز کند (البته کل این سناریو ساختگی بود و صرفاً جهت درک بهتر مطالب آموزشی مورد استفاده قرار گرفت).

در آموزش های آتی این امکان را خواهیم داشت تا بتوانیم بخشی از اطلاعات را به صورت پویا و از طریق صفحه کلید وارد برنامه کنیم تا برنامه طراحی شده بر اساس اطلاعات ورودی نتیجه نهایی را به ما اعلام کند.