

## به نام خدا

### آموزش هشتم

پس از آن که در دو جلسه گذشته با string ها در زبان برنامه نویسی جاوا آشنا شدیم، اکنون می توانیم با استفاده از Operator ها یا همان اعمال ریاضیاتی مثل جمع و تفریق و غیره مقادیر متغیر های متفاوتی را در تعامل با یکدیگر قرار دهیم. فرض کنیم که ما یک متغیر از جنس int تحت عنوان studentNumber داریم. مقدار اختصاص داده شده به متغیر ما ۱۲۱ می باشد:

```
int studentNumber = 121;
```

حال اگر بخواهیم عددی مثل ۱۰ را به مقدار متغیر studentNumber که معادل با ۱۲۱ است اضافه کنیم به این صورت عمل می کنیم:

```
int studentNumber = 121;  
studentNumber = studentNumber + 10;
```

روش دیگری که از آن طریق می توان ۱۰ واحد به ۱۲۱ اضافه کرد به صورت زیر است:

```
int studentNumber = 121;  
studentNumber +=10;
```

در این صورت عدد ۱۰ به مقدار متغیر که ۱۲۱ است اضافه شده، حال از این پس مقدار متغیر studentNumber عدد ۱۳۱ خواهد بود. چنان چه بخواهیم عدد ۱۰ را از مقدار متغیرمان کم کنیم به صورت زیر عمل می کنیم:

```
int studentNumber = 121;  
studentNumber = studentNumber - 10;
```

روش دیگری که از آن طریق می توان ۱۰ واحد از ۱۲۱ کم کرد به صورت زیر است:

```
int studentNumber = 121;  
studentNumber -=10;
```

حال مقدار نهایی متغیر studentNumber عدد ۱۱۱ خواهد بود. چنانچه بخواهیم مقدار متغیر studentNumber را در عددی ضرب کنیم به صورت زیر عمل می کنیم:

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

```
int studentNumber = 121;  
studentNumber = studentNumber * 10;
```

روش دیگری که از آن طریق می توان ۱۰ را در ۱۲۱ ضرب کرد به صورت زیر است:

```
int studentNumber = 121;  
studentNumber *=10;
```

در این صورت نتیجه نهایی ۱۲۱۰ خواهد بود. به عبارت دیگر از این پس در هر کجای برنامه که ما متغیر studentNumber را مورد استفاده قرار دهیم عدد ۱۲۱۰ مبنای محاسبه خواهد بود.

چنانچه بخواهیم مقدار متغیرمان را بر عددی تقسیم کنیم به شکل زیر عمل خواهیم کرد:

```
int studentNumber = 121;  
studentNumber = studentNumber / 10;
```

لازم به ذکر است که علامت / نمی بایست با علامت \ اشتباه گرفته شود. روش دیگری که از آن طریق می توان ۱۲۱ را بر ۱۰ تقسیم کرد به صورت زیر است:

```
int studentNumber = 121;  
studentNumber /=10;
```

که در این صورت مقدار متغیر ما که ۱۲۱ است بر عدد ۱۰ تقسیم خواهد شد و عدد 12.1 به دست خواهد آمد. (توجه داشته باشیم که در اینجا جاوا عدد ۱۲ را به ما نشان خواهد داد چرا که به صورت خودکار عدد حاصله را رند می کند).

حال مواقعی برای ما پیش می آید که تمایل داریم مقدار متغیر خود را به میزان یک عدد بیشتر افزایش یا کاهش دهیم که در این صورت بایستی اصطلاحاتی همچون Incrementing و Decrementing را مورد استفاده قرار دهیم. معنای لغوی این واژگان به ترتیب افزایش دادن و کاهش دادن خواهد بود. به طور مثال چنانچه بخواهیم مقدار عددی متغیر studentNumber به میزان یک عدد افزایش دهیم به صورت زیر عمل می کنیم:

```
int studentNumber = 121;  
studentNumber++;
```

حال مقدار عددی متغیر ما عدد ۱۲۲ خواهد بود. به همین منوال می توانیم یک عدد نیز از مقدار متغیر خود به صورت زیر کم نماییم:

```
int studentNumber = 121;  
studentNumber--;
```

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

حال مقدار نهایی متغیر studentNumber عدد ۱۲۰ خواهد بود.

در مورد Incrementing و Decrementing نکته مهمی را همواره می بایست مد نظر قرار دهیم و آن هم این که چنانچه علامت های + و - قبل از نام متغیر قرار بگیرند (مثلاً ++studentNumber) اصطلاحاً Pre-incrementing نامیده می شوند و چنانچه پس از نام متغیر قرار بگیرند (مثلاً studentNumber++) Post-incrementing نامیده می شوند (در مورد علامت - به ترتیب Pre-decrementing و Post-decrementing نامیده می شوند). تفاوتی که مابین Pre و Post وجود دارد از آنجا ناشی می شود که در مورد Pre یک واحد به مقدار متغیر قبل از نمایش دادن آن اضافه می شود. به عبارت دیگر:

```
class test {  
  
    public static void main(String[] args) {  
        int studentNumber = 121;  
  
        System.out.println(++studentNumber);  
    }  
}
```

در واقع قبل از آنکه مقدار متغیر ما در بخش Console نمایش داده شود یک واحد به آن اضافه می شود و عدد ۱۲۲ نمایش داده خواهد شد. حال اگر علامت ++ را بعد از نام متغیر قرار دهیم، مقدار یک واحد به مقدار متغیر ما اضافه خواهد شد اما پس از آنکه برنامه یک بار اجرا شد. در واقع کد زیر

```
class test {  
  
    public static void main(String[] args) {  
        int studentNumber = 121;  
  
        System.out.println(studentNumber++);  
    }  
}
```

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

یک واحد به متغیر studentNumber پس از آنکه برنامه اجرا شد اضافه می کند که در این صورت ما در بخش Console همان عدد ۱۲۱ را مشاهده خواهیم کرد چرا که برنامه اجرا می شود، مقدار ۱۲۱ نمایش داده می شود، سپس یک واحد به آن اضافه می شود و تبدیل به ۱۲۲ می شود. هم اکنون اگر بخواهیم برنامه خود را تست کنیم که ببینیم آیا یک واحد اضافه شده است یا خیر یک Statement دیگر به کد خود به صورت زیر اضافه می کنیم:

```
class test {  
  
    public static void main(String[] args) {  
        int studentNumber = 121;  
  
        System.out.println(studentNumber++);  
        System.out.println(studentNumber);  
    }  
}
```

حال پس از آنکه یک واحد در Statement اول به مقدار متغیر ما اضافه شد، پس از فراخوانی مجدد متغیر مقدار جدید آن که ۱۲۲ است در بخش Console نمایش داده خواهد شد. زمان هایی در برنامه نویسی برای ما اتفاق می افتد که نیاز داریم بیش از یک عمل ریاضیاتی را در کد خود اعمال کنیم که در این صورت دانستن این نکته که جاوا اول کدام عمل اصلی را محاسبه خواهد کرد از اهمیت بسزایی برخوردار خواهد بود. فرض کنیم که دو متغیر از جنس int به نام های X و Y داریم که متغیر X دارای هیچ داده ای نمی باشد اما متغیر Y دارای مقدار ۱۰ است:

```
int x;  
int y = 10;
```

حال در نظر بگیریم که مقدار متغیر X قرار است حاصلضرب مقدار متغیر Y در عدد ۲ و سپس به علاوه عدد ۶ باشد. به عبارت دیگر کد فوق به صورت زیر تغییر خواهد یافت:

```
int x;  
int y = 10;  
x = y * 2 + 6;
```

نتیجه متغیر X هم می تواند عدد ۲۶ و هم ۸۰ باشد بسته به اینکه اول عمل ضرب انجام شود یا عمل جمع. در زبان برنامه نویسی جاوا اعمال Incrementing و Decrementing اول رخ می دهند،

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

سپس اعمال ضرب و تقسیم صورت گرفته و در نهایت اعمال جمع و تفریق اعمال می شوند. حال با دانستن این قوانین می توان گفت که نتیجه متغیر X در مثال فوق ۲۶ خواهد بود.

گاهی اوقات ما به عنوان برنامه نویس مجبور هستیم در قواعد زبان برنامه نویسی جاوا دخل و تصرف کنیم. به طور مثال گاهی اوقات ما نیاز داریم تا عمل جمع قبل از عمل ضرب صورت گیرد. به عبارت دیگر در مثال فوق ما نیاز داریم تا اول عددهای ۲ و ۶ با یکدیگر جمع شده سپس حاصل جمع آن ها در مقدار متغیر Y که برابر با ۱۰ است ضرب شود. در چنین مواقعی به راحتی می توان دو عددی که می خواهیم با یکدیگر جمع شوند و یا از یکدیگر کم شوند را داخل علامت های ( ) قرار دهیم:

```
int x;  
int y = 10;  
x = y * (2 + 6);
```

حال نتیجه ای که به دست خواهد آمد عدد ۸۰ خواهد بود. نکته ای که در اینجا می بایست حتماً به خاطر بسپاریم این است که نمی توانیم این اعمال را (ضرب، تقسیم، جمع و تفریق) مابین یک متغیر از جنس int و یک متغیر از جنس double اجرا کنیم که در این صورت Compiler در حین اجرای برنامه Error خواهد داد. پس از آشنایی با انواع متغیرها در آموزش پیشین و همچنین آشنایی با Operator ها در این جلسه، خواهیم توانست مهارت کدنویسی خود را یک پله ارتقاء داده و برنامه هایی کمی پیچیده تر بنویسیم. در قسمت آموزشی آتی خواهیم دید که چگونه با استفاده از همین اصول ابتدایی Operator ها می توان پروژه ای طراحی کرد که وزن ما را روی دیگر سیارات منظومه شمسی مثل مشتری و زحل محاسبه کند.