

به نام خدا

آموزش بیست و پنجم

در ادامه مبحث Loop ها در زبان برنامه نویسی جاوا بایستی گفت زمان هایی در طراحی پروژه ها برای برنامه نویسان پیش خواهد آمد که نیاز دارند تا در Loop خود بیش از یک متغیر را مورد سنجش قرار دهند. در واقع این مسئله به مثابه یکی از ویژگی های بسیار کاربردی در زبان برنامه نویسی جاوا است. برای همین منظور مبحث قرار دادن بیش از یک متغیر در Loop یی از جنس for را در قالب پروژه ای کاربردی توضیح خواهیم داد.

طراحی پروژه جدول ضرب

طبق روال گذشته ابتدا نیاز به یک سناریو داریم:

فرض کنیم که یک برادر یا خواهر کوچک تر از خود داریم که در مقطع ابتدایی تحصیل می کند. او در مبحث جدول ضرب کمی دچار مشکل است از این رو از آنجا که می داند برادر یا خواهر بزرگش یک برنامه نویس است، از ما می خواهد که برنامه ای برایش طراحی کنیم که در جدول ضرب به او کمک کند.

برای این منظور پروژه ای جدید تحت عنوان Helping My Sibling 25th Session به معنی "کمک به خواهر/برادرم جلسه بیست و پنجم" ایجاد می کنیم (علت اینکه از واژه انگلیسی Sibling استفاده می کنیم این است که هم مفهوم Sister به معنی "خواهر" را و هم مفهوم Brother به معنی "برادر" را شامل می شود). یک کلاس در پروژه خود تحت عنوان MyMultiplyProgram به معنی "برنامه ضرب من" ایجاد می کنیم. در این پروژه می خواهیم حاصلضرب عدد دو را در دیگر اعداد از یک تا پنجاه به خواهر یا برادرمان آموزش دهیم. به عبارت دیگر می خواهیم برنامه پاسخ هایی از جنس دو یکی دو تا، دو دو تا چهار تا و دو سه تا شش تا و الی آخر را به ما نشان دهد.

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

از آنجا که می خواهیم نحوه به کار گیری دو متغیر را در Loop یی از جنس for را آموزش دهیم، نیاز داریم تا دو متغیر تحت عناوین numberOne و numberTwo به معنی به ترتیب "عدد یک" و "عدد دو" تعریف کنیم. در این مرحله از برنامه، کد ما به شکل زیر خواهد بود:

```
public class MyMultiplyProgram {  
    public static void main(String[] args) {  
        int numberOne;  
        int numberTwo;  
    }  
}
```

در واقع متغیر numberOne قرار است عدد دو را در خود جای دهد که در دیگر اعدادی که در متغیر numberTwo ذخیره می شوند ضرب شود. حال Block مرتبط با for را به برنامه اضافه می کنیم و کد ما به شکل زیر تکمیل خواهد شد:

```
public class MyMultiplyProgram {  
    public static void main(String[] args) {  
        int numberOne;  
        int numberTwo;  
        for () {  
        }  
    }  
}
```

در این مرحله همانطور که قبلاً آموزش داده شده است، می بایست داخل پرانتز را با نقطه آغازین Loop و نقطه پایانی و در نهایت مقدار افزایشی تکمیل کنیم. برای این منظور کد فوق را به شکل زیر تکمیل می کنیم:

```
public class MyMultiplyProgram {
    public static void main(String[] args) {
        int numberOne;
        int numberTwo;
        for (numberOne = 2, numberTwo = 1;) {

        }
    }
}
```

نکته جدید که در این آموزش حتماً می بایست مد نظر قرار داده شود این است که هر موقع بخواهیم بیش از یک متغیر را برای Loop یی در نظر بگیریم، می بایست متغیر ها را با یک کاما از یکدیگر جدا ساخته و پس از آخرین متغیر یک علامت ; قرار دهیم. در این مثال Value اولیه برای متغیر numberOne را عدد دو در نظر گرفته ایم و Value اولیه برای متغیر numberTwo را عدد یک در نظر می گیریم چرا که قرار است برنامه از حاصلضرب دو یکی دو تا شروع به اجرا کند. از آنجا که قرار است حاصلضرب عدد دو را از یک تا پنجاه به خواهر و یا برادر خود آموزش دهیم از این رو در شرط نقطه پایانی برنامه قید می کنیم که این برنامه تا جایی ادامه یابد که حاصلضرب عدد دو در عدد دیگر کوچکتر از یا مساوی با عدد صد شود. حال کد خود را به شکل زیر تکمیل می کنیم:

```
public class MyMultiplyProgram {
    public static void main(String[] args) {
        int numberOne;
        int numberTwo;
        for (numberOne = 2, numberTwo = 1; numberTwo*numberOne <= 100;) {

        }
    }
}
```

حال پس از آنکه یک علامت ; پس از شرط نقطه پایانی Loop قرار دادیم، نوبت به تعریف کردن چگونگی اضافه شدن متغیر numberTwo می رسد. در واقع از آنجا که مقدار متغیر numberOne که معادل با دو است و همواره ثابت است، فقط نیاز است تا مقدار متغیر numberTwo یکی یکی افزایش یابد. به عبارت دیگر در این برنامه نیاز داریم تا در هر بار

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

Loop شدن یک واحد به متغیر numberTwo اضافه شود زیرا می خواهیم عدد دو را در دیگر اعداد ضرب کنیم که این اعداد از یک شروع شده و به عدد پنجاه ختم می شود. از این رو کد ما می بایست به شکل زیر تکمیل شود:

```
public class MyMultiplyProgram {  
    public static void main(String[] args) {  
        int numberOne;  
        int numberTwo;  
        for (numberOne = 2, numberTwo = 1;  
            numberTwo*numberOne<=100; numberTwo++) {  
        }  
    }  
}
```

حال که شرط Loop برنامه خود را تکمیل کردیم، نوبت به نوشتن کد برای خروجی برنامه می رسد. برای این منظور می بایست دستور خروجی برنامه را مابین دو علامت { } که مقابل for قرار گرفته اضافه نماییم. حال می خواهیم تا اولین خروجی برنامه به صورت The result of 1 is 2 به معنی "نتیجه دو در یک می شود دو" نمایش داده شود. از این رو می بایست دستور System.out.println(); را به شکلی تکمیل کنیم که چنین خروجی را در پنجره Console به نمایش در آورد. کد تکمیل شده برنامه به شکل زیر خواهد بود:

```
public class MyMultiplyProgram {  
    public static void main(String[] args) {  
        int numberOne;  
        int numberTwo;  
        for (numberOne = 2, numberTwo = 1; numberTwo * numberOne <=  
100; numberTwo++) {  
            System.out.println("The result of " + numberOne + " *  
" + numberTwo + " is " + (numberOne * numberTwo));  
        }  
    }  
}
```

حال به توضیح آنچه مابین دو پرانتز System.out.println(); قرار گرفته است خواهیم پرداخت. در ابتدا می خواهیم که عبارت The result of به معنی "نتیجه ی" ابتدای خط در

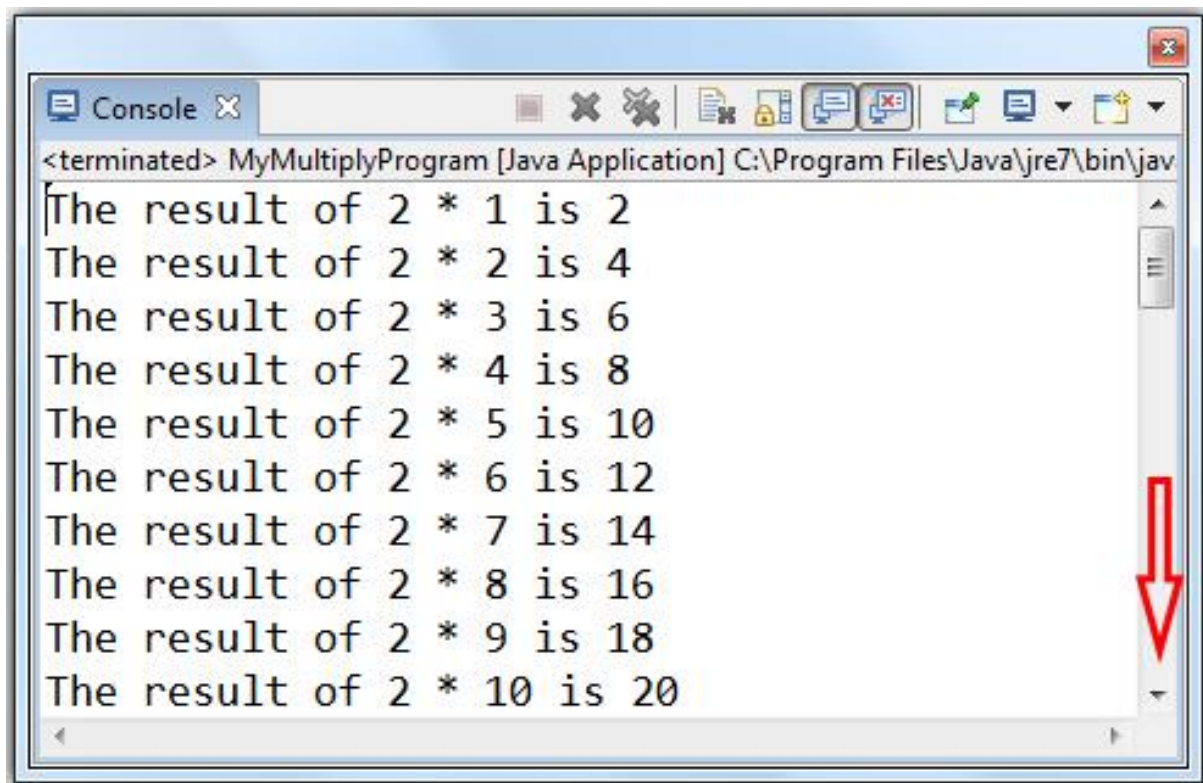
دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

پنجره Console به نمایش در آید. برای همین منظور عبارت فوق را مابین “ “ قرار می دهیم. سپس نیاز داریم تا بگوییم مثلاً دو ضرب در یک، از اینرو یک علامت + قرار داده و نام متغیر اول که numberOne است را می نویسیم سپس نیاز داریم تا علامت * روی صفحه مانیتور به نمایش در آید اما اگر خود علامت * را پس از متغیر numberOne بنویسیم، برنامه به شکل یک Operator به آن نگاه می کند پس باید پس از قرار دادن یک علامت + آن را داخل دو علامت “ “ قرار دهیم که در حین Compile شدن اکلیپس آن را به عنوان یک کاراکتر یا حتی یک متن تلقی کند. سپس یک علامت + دیگر قرار داده و نام متغیر دوم که numberTwo است را می نویسیم. تا اینجا کار اولین خروجی برنامه $2*1$ The result of خواهد بود. حال یک علامت + دیگر قرار داده و واژه is را به معنی "هست" داخل دو علامت “ “ می نویسیم. حال خروجی برنامه The result of $2*1$ is خواهد شد. همانطور که در آموزش های پیشین توضیح داده شد هر زمانیکه بخواهیم یک عمل ریاضیاتی را به طور مجزا از دیگر بخش های برنامه اجرا کنیم، می بایست تا آن را داخل دو پرانتز قرار دهیم و از آنجا که در این برنامه می خواهیم حاصلضرب مقادیر متغیرهای numberOne و numberTwo را بدست آوریم پس نیاز داریم تا عبارتی همچون $(numberOne*numberTwo)$ را نیز پس از نوشتن یک علامت + به آخر کد اضافه کنیم.

در این مرحله برنامه ما تکمیل شده است و می توانیم آن را اجرا کنیم:



```
<terminated> MyMultiplyProgram [Java Application] C:\Program Files\Java\jre7\bin\jav
The result of 2 * 1 is 2
The result of 2 * 2 is 4
The result of 2 * 3 is 6
The result of 2 * 4 is 8
The result of 2 * 5 is 10
The result of 2 * 6 is 12
The result of 2 * 7 is 14
The result of 2 * 8 is 16
The result of 2 * 9 is 18
The result of 2 * 10 is 20
```

همانطور که در تصویر بالا مشاهده می شود برنامه ما به درستی کار می کند. فلش قرمز رنگ به معنی آن است که این لیست ادامه داشته و تا نشان دادن حاصلضرب عدد دو ضرب در پنجاه ادامه خواهد یافت. در واقع از آنجا که حاصلضرب عدد دو در پنجاه مساوی با صد خواهد بود و در Loop خود تعریف کرده ایم که تا عدد صد این Loop ادامه یابد، از این رو برنامه از Loop پس از رسیدن به عدد صد خارج خواهد شد.

پیش از به پایان رساندن این بخش از آموزش نیاز است تا دو نکته یادآوری شود. نکته اول از این قرار است که ما می توانیم با Operator هایی همچون به علاوه و منها انواع متغیرها را نه تنها با یکدیگر بلکه حتی با string ها نیز ادغام کنیم که در این صورت خروجی برنامه یک string خواهد بود. نکته دوم اینکه برای خوانایی بیشتر خروجی برنامه می توان از Space قبل و بعد از علامت ضرب که داخل دو علامت “ “ قرار گرفته است استفاده کرد.

برای روشن شدن مطلب کد فوق را به شکل زیر تغییر می دهیم:

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

```

public class MyMultiplyProgram {
    public static void main(String[] args) {
        int numberOne;
        int numberTwo;
        for (numberOne = 2, numberTwo = 1; numberTwo * numberOne <=
100; numberTwo++) {
            System.out.println("The result of" + numberOne + "*" +
numberTwo + "is" + (numberOne * numberTwo));
        }
    }
}

```

حال چنانچه برنامه را اجرا کنیم، خروجی برنامه مشابه تصویر زیر خواهد بود:

```

<terminated> MyMultiplyProgram [Java Application] C:\Program Files\Java\jre7\bin\jav
The result of 2*1 is 2
The result of 2*2 is 4
The result of 2*3 is 6
The result of 2*4 is 8
The result of 2*5 is 10
The result of 2*6 is 12
The result of 2*7 is 14
The result of 2*8 is 16
The result of 2*9 is 18
The result of 2*10 is 20

```

در تصویر فوق مشاهده می شود که نحوه عملکرد برنامه هیچ تغییری نکرده است با این تفاوت که به خاطر حذف Space ها خوانایی جملات به مراتب کمتر شده است.

حال برنامه نویسان مبتدی خواهند توانست تا با تسلط به مباحث Loop نرم افزار هایی از این دست طراحی نمایند. به طور مثال با تلفیق کلاس Scanner و همچنین Loop ها می توان برنامه ای طراحی کرد تا به طور مثال خواهر یا برادر ما به سادگی بتواند مثلاً در مثال بالا numberOne را خود برای نرم افزار تعریف کند و خروجی برنامه را ببیند. به عبارت دیگر به

جای آن که عدد دو در دیگر اعداد ضرب شود، عدد دیگری همچون عدد هفت از طریق کیبورد وارد برنامه شده و حاصلضرب این عدد در دیگر اعداد نمایش داده خواهد شد.