

به نام خدا

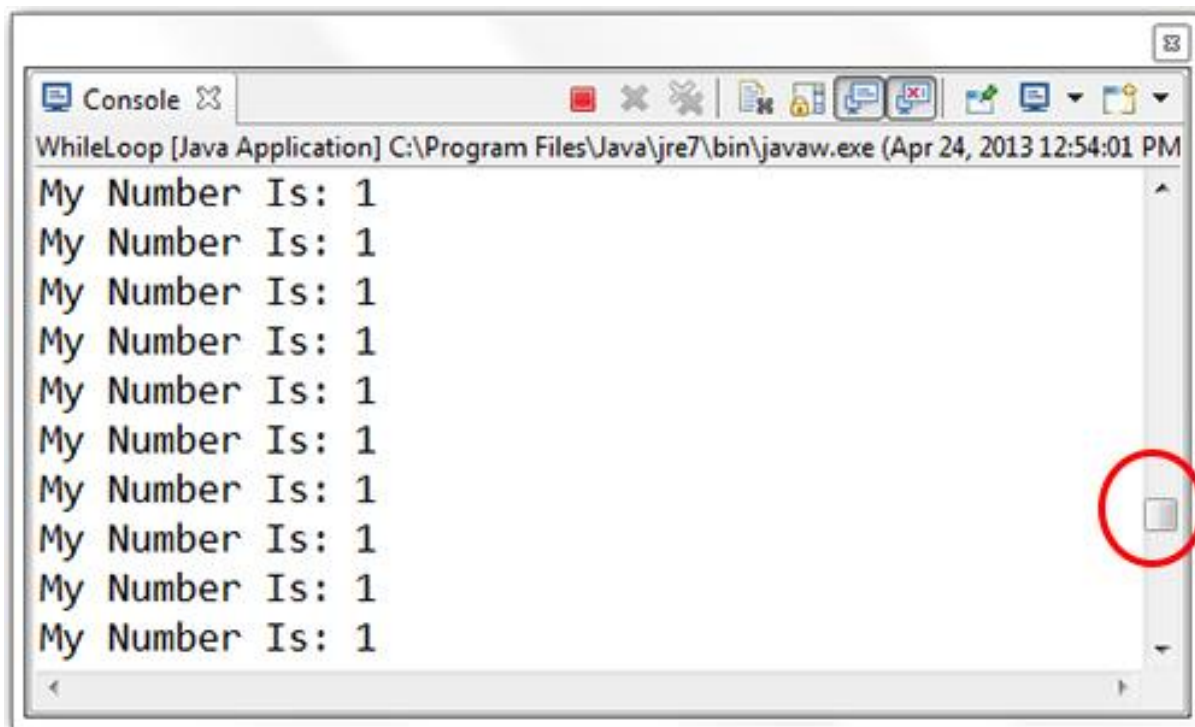
آموزش بیست و سوم

پس از آشنایی با اولین و پیچیده ترین نوع Loop ها در آموزش بیست و دوم، حال نوبت به بررسی نوع دیگری از انواع Loop ها تحت عنوان while می رسد. در واقع بیشترین وجه تمایز for و while تفاوت ساختاری است. در Loop یی از جنس while بر خلاف for ما ابتدا به ساکن می بایست متغیر خود را تعریف نماییم سپس در while() شرط خود و یا همان نقطه پایانی Loop را در نظر بگیریم و در داخل دو علامت {} که مقابل while() قرار می گیرد مقدار افزایش متغیر را تعریف کنیم. برای روشن تر شدن مطالب فوق، پروژه ای تحت عنوان 23rd Session به معنی **جلسه بیست و سوم** ایجاد کرده سپس یک کلاس تحت عنوان WhileLoop در آن ایجاد می کنیم و مطالب فوق را در آن به صورت زیر اعمال می کنیم:

```
public class WhileLoop {  
    public static void main(String[] args) {  
        int number = 1;  
        while (number <= 10) {  
            System.out.println("My Number Is: " + number);  
        }  
    }  
}
```

همانطور که مشاهده می کنیم، متغیری از جنس int خارج از Loop تعریف کرده ایم و Value اولیه آن را معادل با یک قرار داده ایم. سپس ساختار اولیه Loop یی از جنس while را به شکل while(){ } نوشته ایم و به این صورت آن را تکمیل کرده ایم که شرط خود یا همان نقطه پایانی را داخل پرانتز به صورت number <= 10 نوشته ایم. به عبارت دیگر مادامیکه مقدار متغیر ما تحت عنوان number کوچکتر یا مساوی با ده باشد، این Loop ادامه خواهد یافت. سپس در داخل دو علامت {} خروجی برنامه را می نویسیم و از آنجاییکه می خواهیم خروجی برنامه ما در پنجره Console به نمایش در آید از دستوری System.out.println(); استفاده می کنیم و در آن پارامتری تحت عنوان My Number Is: به معنی **"عدد من هست:"** به علاوه نام

متغیر خود که همان number است را می نویسیم. حال بایستی انتظار داشته باشیم پس از اجرای برنامه مادامیکه مقدار متغیر ما کوچکتر یا مساوی با ده است این Loop اجرا شود. پس از اجرای برنامه با موقعیتی مواجه می شویم که در تصویر زیر نشان داده شده است:



در حقیقت از آنجا که در Loop خود تعریف نکرده ایم که مقدار متغیر ما به چه میزان در هر بار Loop شدن افزایش یابد، این Loop تا بی نهایت ادامه خواهد یافت (اگر به دایره قرمز رنگ در عکس توجه کنیم می بینیم که نشانگر لیستی طولانی از Loop شدن متغیر است که مسلماً بیش از ۱۰ بار می باشد).

برای رفع این مشکل نیاز داریم تا مقدار افزایشی متغیر خود در هر بار Loop شدن را به برنامه اضافه کنیم. نکته ای که در اینجا می بایست مد نظر قرار دهیم این است که در while بر خلاف for این مقدار افزایشی را می بایست در داخل دو علامت { } قرار دهیم و این در حالی است که جای مقدار افزایشی در نحوه اجرای برنامه خیلی تعیین کننده است. برای روشن شدن این مطلب کد خود را به شکل زیر تکمیل می کنیم:

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

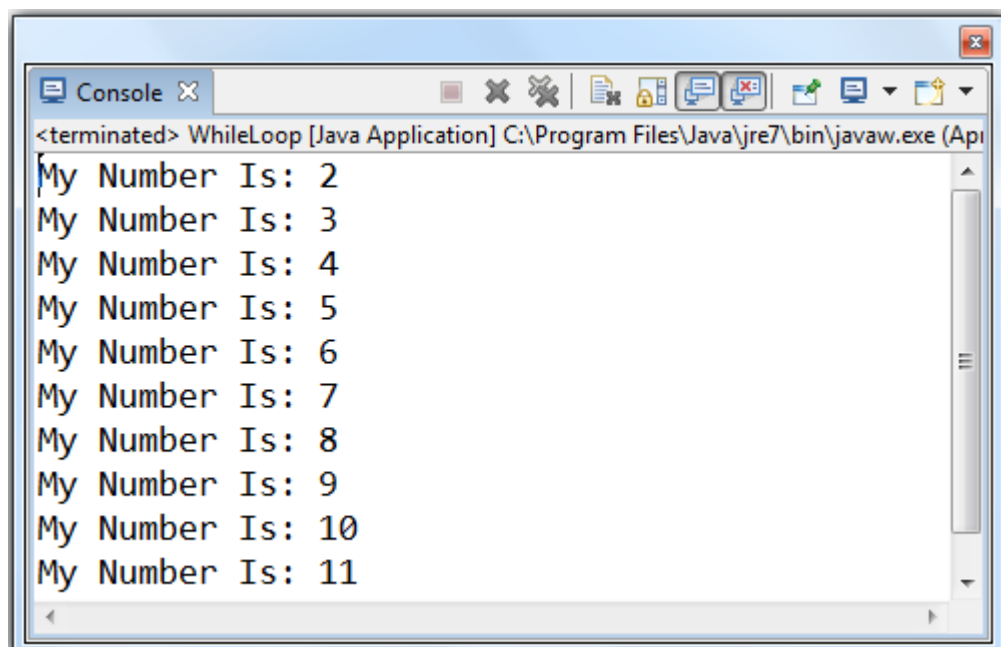
مدرس: بهزاد مرادی

```

public class WhileLoop {
    public static void main(String[] args) {
        int number = 1;
        while (number <= 10) {
            number++;
            System.out.println("My Number Is: " + number);
        }
    }
}

```

در این کد پیش از اجرای دستور `System.out.println("My Number Is: " + number);` برنامه یک واحد به مقدار متغیر `number` اضافه کرده سپس دستور بعدی که به نمایش در آوردن `My number is: + number` در پنجره Console است را اجرا می کند که در این صورت همانطور که ملاحظه می شود مقدار اولیه متغیر ما مساوی با یک است، حال `while` به برنامه دستور می دهد که یک واحد به مقدار اولیه متغیر اضافه شود و در نهایت مقدار نهایی را در پنجره Console به نمایش در آورد که این مقدار نهایی معادل با عدد دو است چرا که مقدار افزایشی را `number++` تعریف کرده ایم. خروجی برنامه در این مرحله به شکل زیر خواهد بود:



```

<terminated> WhileLoop [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (Api
My Number Is: 2
My Number Is: 3
My Number Is: 4
My Number Is: 5
My Number Is: 6
My Number Is: 7
My Number Is: 8
My Number Is: 9
My Number Is: 10
My Number Is: 11

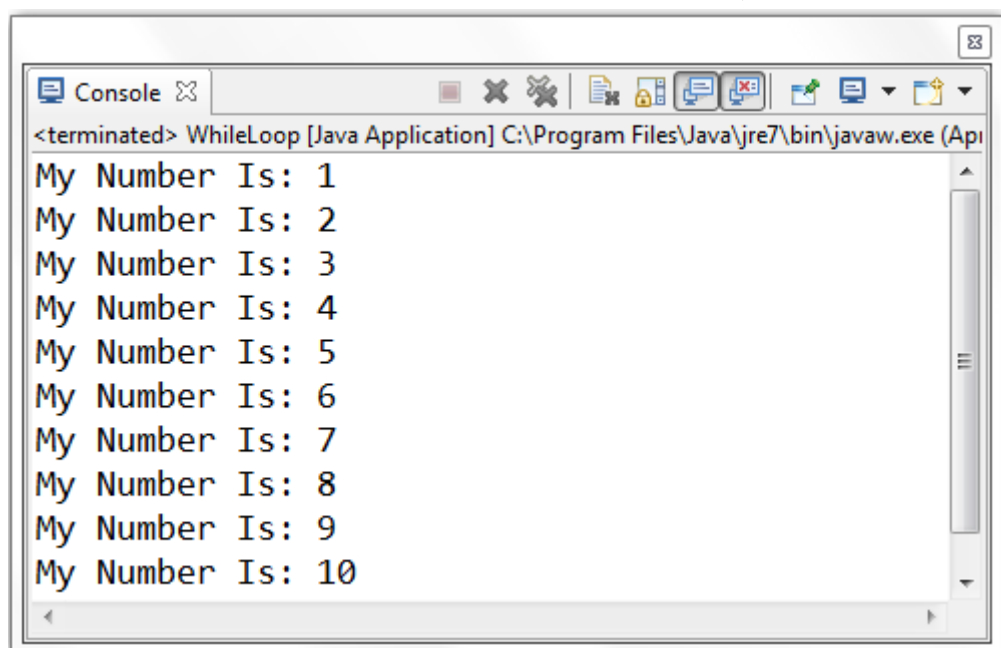
```

همانطور که انتظار می رفت اولین خروجی برنامه عدد دو است و این کار ادامه می یابد تا Loop طراحی شده توسط ما به عدد ده می رسد. از آنجا که ما خود به برنامه دستور داده ایم مادامیکه مقدار متغیر کوچکتر یا مساوی با ده است این کار ادامه یابد پس در فاز یکی مانده به آخر که My Number Is: 10 می باشد جواب شرط قرار گرفته در داخل پرانتز true است پس این Loop یک بار دیگر اجرا می شود و عدد یازدهم به نمایش در می آید اما در این مرحله از برنامه از آنجا که جواب شرط داخل پرانتز به عدد یازده معادل با false می باشد برنامه ما از Loop خارج خواهد شد.

حال ببینیم اگر جای مقدار افزایشی را از ماقبل دستور نمایش خروجی برنامه به بعد از آن تغییر دهیم چه اتفاقی خواهد افتاد. برنامه خود را به شکل زیر بازنویسی خواهیم کرد:

```
public class WhileLoop {  
    public static void main(String[] args) {  
        int number = 1;  
        while (number <= 10) {  
            System.out.println("My Number Is: " + number);  
            number++;  
        }  
    }  
}
```

برنامه را اجرا می کنیم:



```
<terminated> WhileLoop [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (Api  
My Number Is: 1  
My Number Is: 2  
My Number Is: 3  
My Number Is: 4  
My Number Is: 5  
My Number Is: 6  
My Number Is: 7  
My Number Is: 8  
My Number Is: 9  
My Number Is: 10
```

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

همانطور که ملاحظه می شود خروجی برنامه ما با برنامه پیشین کاملاً متفاوت است. برای روشن شدن این وجه تمایز به بررسی خط به خط کد خود می پردازیم. پس از اجرای برنامه، دستور `while` می بیند که مقدار اولیه تعیین شده برای متغیر `number` معادل با عدد یک است و پاسخ به شرط داخل پرانتز که می گوید این عدد می بایست کوچکتر از عدد ده باشد و یا حداکثر مساوی با عدد ده باشد، نیز `true` است پس بلافاصله دستور `System.out.println()` را اجرا می کند و در پنجره Console عبارت `My Number Is: 1` به نمایش در می آید. پس از به نمایش در آمدن عبارت `My Number Is: 1` حال نوبت به اجرای دستور `number++` می رسد که عبارت از آن است که یک واحد به مقدار متغیر `number` اضافه کند. در واقع پس از به نمایش در آمدن اولین عبارت یک واحد به مقدار متغیر `number` اضافه می شود. این کار ادامه می یابد تا به عدد نه می رسیم. از آنجا که جواب به شرط `number <= 10` معادل با `true` است این `Loop` یک بار دیگر اتفاق می افتد و `Value` اختصاص داده شده به متغیر `number` معادل با عدد ده می شود. حال برنامه می خواهد تا مجدداً `Loop` شود از اینرو اجرای برنامه یک بار دیگر از اول شروع می شود اما از آنجا که در شرط داخل پرانتز قید کرده ایم که مقدار متغیر می بایست کوچکتر و یا حداکثر مساوی با عدد ده باشد و در این مرحله مقدار متغیر برابر با ده است، برنامه از `Loop` خارج خواهد شد و بر خلاف برنامه پیشین عدد یازده به نمایش در نخواهد آمد و برنامه ما به پایان می رسد.

در قسمت آموزشی آینده با `do-while Loop` ها آشنا می شویم.