

## به نام خدا

### آموزش ششم

پس از آشنایی با انواع متغیرها از جنس Primitive Type حال نوبت به آشنایی با یکی از معروف ترین انواع Reference Type ها تحت عنوان String می رسد.

اگر به خاطر داشته باشید در اولین برنامه جاوا که طراحی کردیم عبارت In the Name of God را در یک String ذخیره ساخته سپس آن را نمایش دادیم. عناصر تشکیل دهنده String ها کاراکترها، اعداد و علائم می باشند که با کنار یکدیگر قرار گرفتن یک جمله را تشکیل می دهند. چنانچه بخواهیم جمله ای را در برنامه خود داشته باشیم، یکی از دو گزینه ای که این کار را برای ما انجام خواهد داد کلاسی از جنس String است (توجه داشته باشید که این متغیر بر خلاف دیگر متغیرها با حرف بزرگ شروع می شود). در حقیقت این Class در API زبان برنامه نویسی جاوا ایجاد شده است و ما فقط آن را فرا می خوانیم. از سوی دیگر کلاس String از جنس Reference Type است. به عبارت دیگر چنانچه ما متغیری داشته باشیم که جنس آن یک از هشت نوع Primitive Type نباشد، ما متغیری از جنس یک Class خواهیم داشت که نوع آن Reference Type می باشد. برای روشن تر شدن این مطلب می توانیم فرض کنیم که String یک Class است و نامی که ما برای آن در نظر می گیریم نام آن Class است. به مثال زیر توجه فرمایید:

```
String myString = "My Name Is Behzad";
```

واژه String به یک Class اشاره می کند که قبلاً توسط تیم طراحان زبان برنامه نویسی جاوا در Java API طراحی شده است. واژه myString به منزله یک Instance یا

نمونه ای از کلاس String است. در واقع myString یک Object یا شیء است(در صورت درک ناقص این مسئله، به مفاهیم شئی گرایی در آموزش های پیشین مراجعه فرمایید).

حال مقدار این String را جمله My Name Is Behzad در نظر می گیریم که داخل دو علامت " " قرار می گیرد(لازم به ذکر است که علامت های " " در حین اجرا نمایش داده نخواهند شد).

یکی از ابتدایی ترین راه کارهای نمایش دادن جمله ای در محیط جاوا، به کار گیری فرمان زیر است که قبلاً با عناصر تشکیل دهنده آن آشنا شده ایم:

```
System.out.println();
```

هم اکنون برای به نمایش در آوردن مقدار String خود تحت عنوان myString می بایست کدی به این شکل داشته باشیم:

```
public class Test {  
  
    public static void main(String[] args) {  
        String myString = "My Name Is Behzad";  
        System.out.println(myString);  
    }  
}
```

در این مرحله با اجرای فرمان Run جمله داخل " " در بخش Console نرم افزار اکلیپس به نمایش در خواهد آمد(چنانچه بخواهیم از طولانی شدن کد خود جلوگیری به عمل آوریم، به جای تعریف کردن یک Class از جنس String و اختصاص دادن یک Value به آن و سپس فرا خواندن String در System.out.println(); به راحتی می توانیم جمله مد نظر خود را داخل پرانتز مقابل println بنویسیم).  
به عبارت دیگر کد فوق را به شکل زیر هم می توانیم داشته باشیم:

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

```
public class Test {

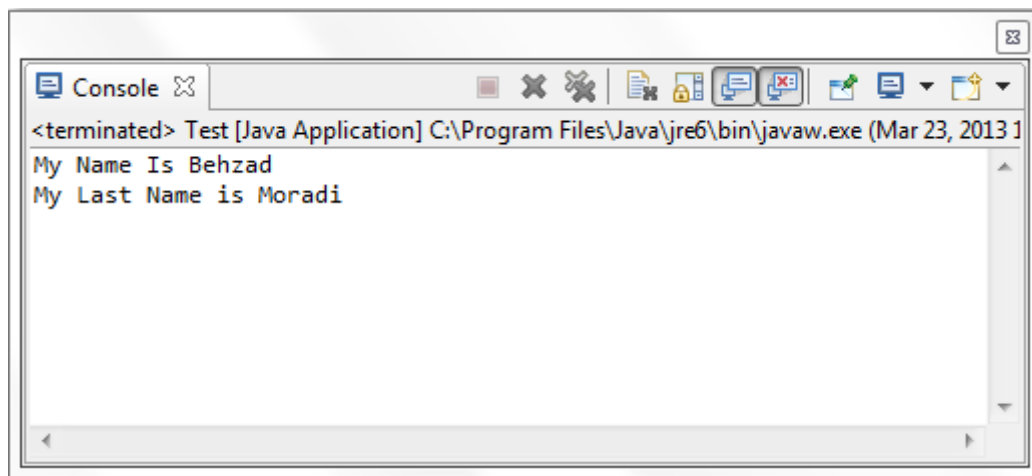
    public static void main(String[] args) {
        System.out.println("My Name Is Behzad");
    }
}
```

در حقیقت خروجی هر دو کد یک چیز خواهد بود (با استفاده از این دو روش ما می توانیم جمله ای را در زبان جاوا به نمایش در آوریم). دانستن نکته ای که در این جا حائز اهمیت است این است که ما می توانیم چندین Statement از این نوع را پشت سر هم داشته باشیم. برای روشن تر شدن این مطلب به مثال زیر توجه کنید:

```
public class Test {

    public static void main(String[] args) {
        System.out.println("My Name Is Behzad");
        System.out.println("My Last Name is Moradi");
    }
}
```

در این مثال ما هر دو جمله داخل " " را در بخش Console نرم افزار اکیپس در دو خط جداگانه خواهیم دید.



دوره آموزش جاوا

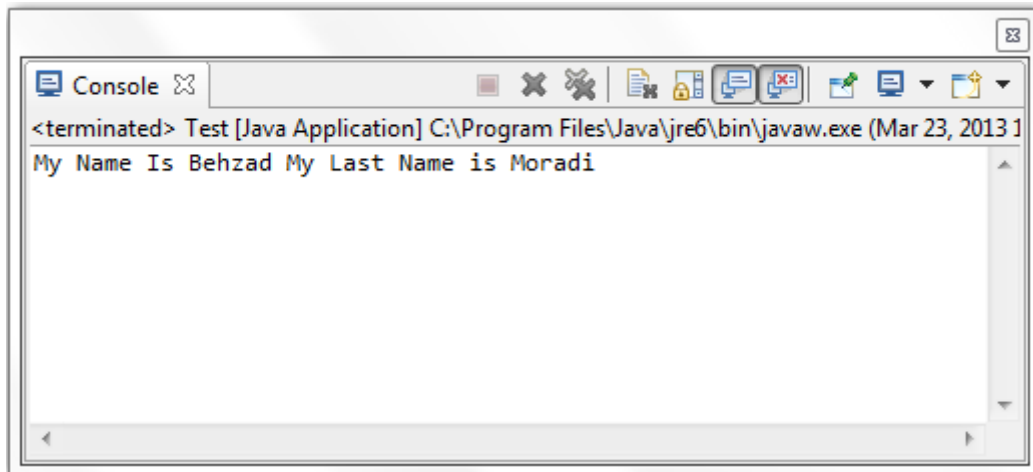
کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

در این مرحله از آموزش به راحتی خواهیم توانست تفاوت مابین دو دستور `println()` و `print()` را درک کنیم. در حقیقت زمانیکه ما بخواهیم یک خط به اتمام برسد و جمله بعدی در خط دوم به نمایش در آید، می بایست از دستور `println()` استفاده کنیم اما زمانیکه بخواهیم نوشته های ما در یک خط و پشت سر هم قرار گیرند، می بایست از دستور `print()` استفاده کنیم. برای روشن شدن این وجه تمایز به مثال زیر توجه فرمایید:

```
public class Test {  
  
    public static void main(String[] args) {  
        System.out.print("My Name Is Behzad ");  
        System.out.print("My Last Name is Moradi");  
    }  
}
```

خروجی این کد به شکل زیر خواهد بود:

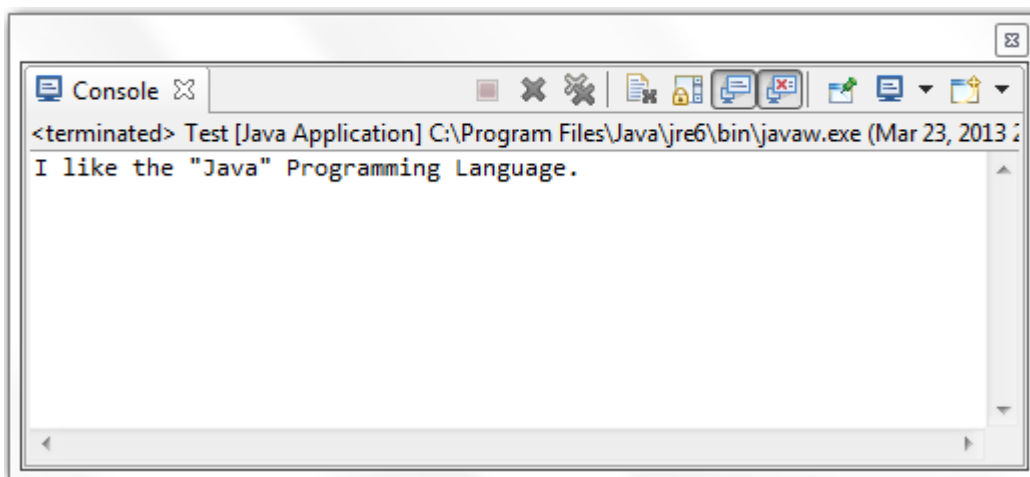


در حقیقت از آن جا که ما دستور صادر کرده ایم که هر دو Statement یکی پس از دیگری در یک خط قرار گیرند، Console یک خط بیشتر نمایش نخواهد داد.

همانطور که قبلاً اشاره شد علامت های " " در Console مشاهده نخواهند شد. اما زمان هایی برای ما پیش می آید که نیاز داریم کلمه ای را داخل " " قرار دهیم. برای این منظور می توانیم از ترکیب علامت های \" استفاده کنیم. به مثال زیر توجه کنید:

```
public class Test {  
  
    public static void main(String[] args) {  
        System.out.print("I like the \"Java\" Programming Language");  
    }  
}
```

خروجی این کد به شکل زیر خواهد بود:



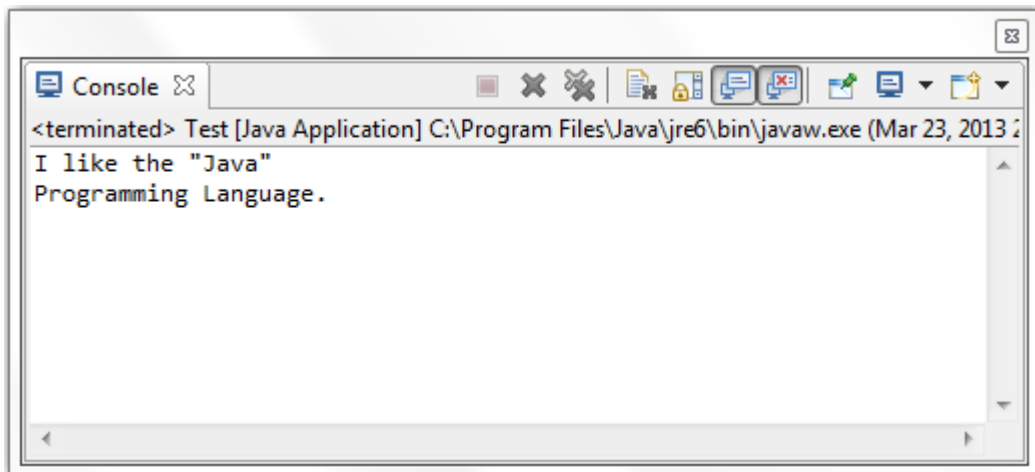
با این کار واژه Java داخل علامت " " قرار می گیرد. با استفاده از این تکنیک می توانیم برای قرار دادن یک Tab بعد از کلمه ای از \t و همچنین فرستادن بخشی از جمله به خط بعد از \n نیز استفاده کنیم. به عنوان مثال اگر بخواهیم دو کلمه Programming Language را به خط بعد ارسال کنیم، از دستور زیر استفاده خواهیم کرد:

```
public class Test {

    public static void main(String[] args) {
        System.out.print("I like the \"Java\" \nProgramming Language.");
    }

}
```

خروجی کد فوق به شکل زیر نمایش داده خواهد شد:



مفهوم Concatenation در زبان برنامه نویسی جاوا به زمانی اطلاق می گردد که ما دو چیز را به یکدیگر مرتبط می سازیم. به طور مثال زمان هایی در حین استفاده از دستورات فوق پیش می آید که ما نیاز داریم تا یک عبارت بسیار طولانی را در Console به نمایش در آوریم که در این صورت می بایست از علامت + که یک Operator است استفاده کنیم (در آموزش های آتی به طور مفصل پیرامون Operator ها صحبت خواهد شد).

برای روشن شدن این مطلب به مثال زیر توجه نمایید:

```
public class Test {

    public static void main(String[] args) {
        System.out.print("I like the \"Java\" Programming Language."
            + "This language is used to create android apps.");
    }
}
```

در مثال فوق مشاهده می شود که دو گروه از کلماتی که در کنار یکدیگر داخل " " قرار گرفته با استفاده از علامت + دنبال هم قرار گرفته اند (لازم به ذکر است علامت + در مورد متغیرهایی همچون int که دارای داده ای از جنس عددی هستند منجر به جمع شدن مقادیر آن متغیرهای می شود اما در مورد string ها به این شکل است که آن ها را در کنار یکدیگر قرار می دهد).

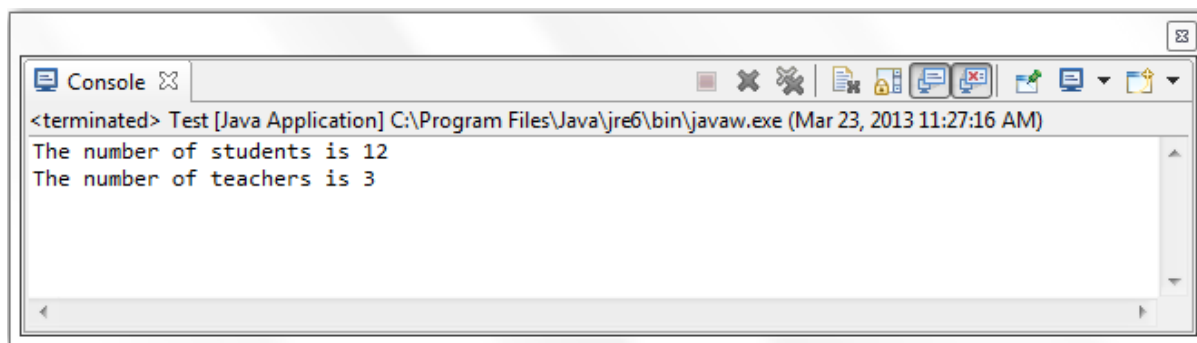
گاهی اوقات ما نیاز داریم تا متنی از جنس string را در کنار دیگر متغیرها مثلاً متغیری از جنس int یا double در محیط Console به نمایش در آوریم. به مثال زیر توجه نمایید:

```
public class Test {

    public static void main(String[] args) {
        int studentNumber = 12;
        int teacherNumber = 3;
        System.out.println("The number of students is " + studentNumber);
        System.out.println("The number of teachers is " + teacherNumber);
    }
}
```

در مثال فوق ما دو متغیر از جنس int تحت عناوین studentNumber و teacherNumber داریم که به ترتیب دارای مقدار ۱۲ و ۳ هستند. حال می خواهیم تا متغیری از جنس int را با یک string یا به عبارت دیگر جمله ای ترکیب کنیم. برای این منظور پس از علامت " یک علامت + قرار داده سپس نام متغیر خود را قرار می

دهیم. در حین Compile شدن، پس از به نمایش در آوردن string در محیط Console حال علامت + مقدار عددی متغیر int را به متن اضافه می کند. در واقع خروجی برنامه به این شکل خواهد بود:



```
<terminated> Test [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (Mar 23, 2013 11:27:16 AM)
The number of students is 12
The number of teachers is 3
```

در آموزش آینده با مباحث تکمیلی String ها همچون مقایسه دو String با یکدیگر، شمردن تعداد کاراکترهای یک String و همچنین تغییر حروف بزرگ به کوچک و بالعکس بحث خواهیم کرد.