به نام خدا

آموزش چهل و دوم

اهداف آموزشی این قسمت عبارتند از:

۱. معرفی دستور super در زبان جاوا
۲. استفاده از کلید واژه super در متغیرها
۳. استفاده از کلید واژه super در متدها

یکی از مباحثی که تسلط به آن در رابطه با به کار گیری وارثت در زبان برنامه نویسی جاوا ضروری است نحوه به کارگیری کلید واژه super است. چنانچه بخواهیم به طور خلاصه کاربرد کلید واژه super را بیان کنیم، بایستی گفت که هر موقع که بخواهیم در کلاسهای Subclass یک متد، متغیر و یا Constructor را در Superclass فرا بخوانیم می بایست از کلید واژه super استفاده کنیم. برای روشن شدن اهمیت این کلید واژه به توضیحات بیشتر در قالب یک پروژه خواهیم پرداخت. پروژه جدیدی در اکلیپس تحت عنوان 42nd Session به معنی **"جلسه چهل و دوم"** ایجاد کرده و دو کلاس به اسم های SuperClass و SubClass به معنی به ترتیب **"کلاس اصلی"** و **"کلاس زیرشاخه"** در آن می سازیم. برای کلاسی که قرار است به منزله نقطه شروع برنامه باشد نام ActionClass به معنی **"کلاس عملیاتی"** را در نظر گرفته و گزینه public static void را برای آن تیک دار می کنیم.

در این مرحله نیاز است تا کلاس SuperClass را به شکل زیر تکمیل نماییم:

```
public class SuperClass {
	int speed = 280;
	public void showLexusSpeed() {
		System.out.println("Lexus speed is " + speed);
	}
}
```

در کد فوق در کلاس SuperClass یک Instance Variable از جنس int تحت عنوان speed به معنی **"سرعت"** ایجاد کرده ایم و مقدار اولیه آن را معادل با ۲۸۰ قرار داده ایم(منظور ۲۸۰ کیلومتر در ساعت است) سپس یک متد از جنس void تحت عنوان showLexusSpeed

به معنی "**سرعت لکسوس را نمایش بده**" ایجاد کرده ایم. در نهایت با استفاده از دستور System.out.println(); می خواهیم مقدار متغیر speed را روی صفحه مانیتور نمایش دهیم. در این دستور علاوه بر نشان دادن مقدار متغیر speed می خواهیم جمله Lexus speed is به معنی "**سرع لکسوس برابر است با ...**" هم نشان داده شود. برای همین منظور این جمله را داخل دو علامت " " نوشته و یک علامت به علاوه قرار داده سپس نام متغیر خود را می نویسیم. اکنون نوبت به تکمیل کد کلاس SubClass می رسد که می بایست به شکل زیر عمل کنیم:

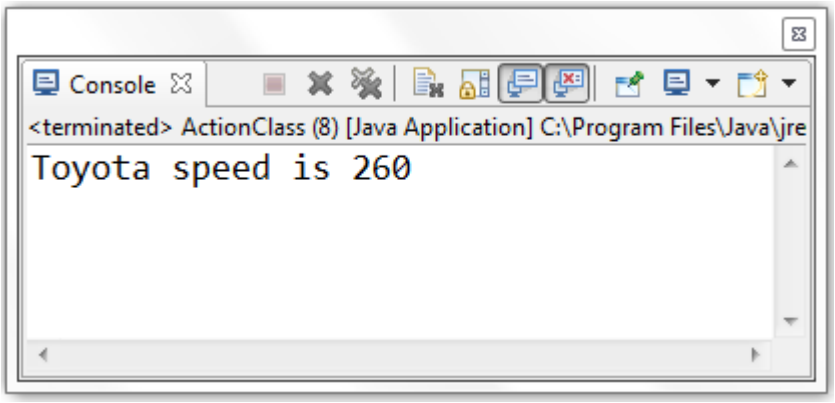
```
public class SubClass extends SuperClass {
	int speed = 260;
	public void showToyoTaSpeed() {
		System.out.println("Toyota speed is "+speed);
	}
}
```

همانطور که در کد فوق مشاهده می شود، کلاس SubClass از کلاس SuperClass ارث بری خواهد کرد(این کار با استفاده از کلید واژه extends به علاوه نام کل SuperClass صورت می گیرد). مجدد یک متغیر از جنس int تحت عنوان speed ایجاد کرده ولی این بار مقدار اولیه ۲۶۰ را برای آن در نظر می گیریم. متد ساخته شده در این کلاس showToyotaSpeed به معنی "**سرعت تویوتا را نشان بده**" می باشد. در نهایت همانند کلاس قبلی از دستور System.out.println(); استفاده می کنیم تا مقدار متغیر speed را به علاوه جمله Toyota speed is به معنی "**سرعت تویوتا برابر است با ...**" روی صفحه مانیتور نمایش دهیم. اکنون می خواهیم از روی این کلاس یک شیئ ساخته و آن را در متد main در کلاس ActionClass نمایش دهیم. برای همین منظور کلاس ActionClass را به شکل زیر تکمیل می کنیم:

```
public class ActionClass {

	public static void main(String[] args) {
		SubClass myObject = new SubClass();
		myObject.showToyoTaSpeed();
	}
}
```

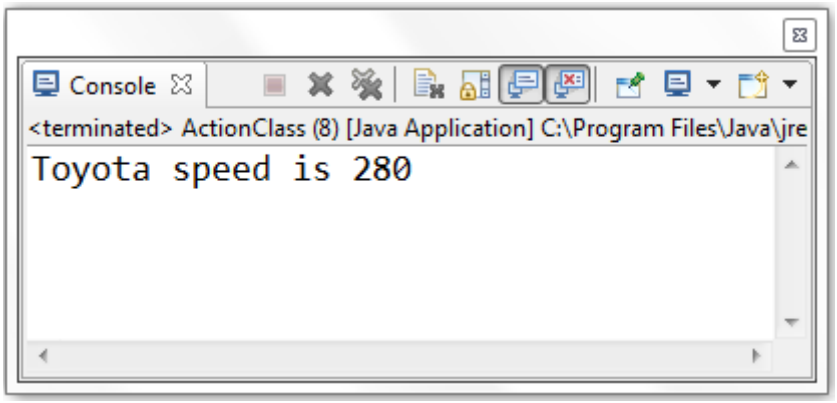
همانطور که در کد فوق مشاهده می شود یک شیئ جدید از روی کلاس SubClass به نام myObject به معنی **"شیئ من"** ساخته، سپس با استفاده از این شیئ جدید متد موجود در داخل کلاس SubClass را فرا می خوانیم. حال برنامه را اجرا می کنیم:



همانطور که در تصویر فوق مشاهده می شود برنامه متد showToyotaSpeed را فراخوانده و خروجی آن را در پنجره Console نمایش می دهد. نکته ای که در اینجا می بایست مد نظر قرار داده شود این است هم در کلاس SuperClass و هم در کلاس SubClass ما متغیر speed را نوشته ایم. در اجرای فوق علیرغم اینکه کلاس SubClass از کلاس SuperClass ارث بری می کند، اما این در حالی است که شیئ ساخته شده در آن از متغیر موجود داخل SubClass استفاده می کند و نتیجه این می شود که سرعت 260 کیلومتر در ساعت برای خودروی تویوتا در نظر گرفته می شود. حال فرض کنیم می خواهیم از عمد کاری کنیم که مقدار متغیر speed موجود در کلاس SuperClass در متد فراخوانده شود. برای این منظور کد خود در داخل متد SubClass را به شکل زیر ویرایش می کنیم:

```
public class SubClass extends SuperClass {
	int speed = 260;
	public void showToyoTaSpeed() {
		System.out.println("Toyota speed is " + super.speed);
	}
}
```

همانطور که در کد فوق ملاحظه می شود، قبل از speed از کلید واژه super به علاوه یک نقطه استفاده کرده ایم. حال مجدداً برنامه را اجرا می کنیم:



می بینیم که کلید واژه super این دستور را به برنامه می دهد که بایستی برود از متغیر موجود در کلاس اصلی استفاده کند. بنابراین مقدار ۲۸۰ کیلومتر در ساعت برای خودروی تویوتا در نظر گرفته می شود.

اکنون ببینیم که به چه نحوی می توان از کلید واژه super برای متدها استفاده کرد. برای روشن شدن مطلب کد کلاس SuperClass را به شکل زیر بازنویسی می کنیم:

```
public class SuperClass {

	public void showSuperClass() {
		System.out.println("This is the SuperClass method");
	}
}
```

در کد فوق یک متد تحت عنوان showSuperClass به معنی **"کلاس اصلی را نشان بده"** ساخته و وظیفه این متد آن است که عبارت This is the SuperClass method به معنی **"این متد کلاس اصلی است"** را نمایش دهد. سپس به صورت زیر اقدام با بازنویسی کلاس SubClass می کنیم:

```
public class SubClass extends SuperClass {

	public void showSubClass() {
		System.out.println("This is the SubClass method");
	}
}
```

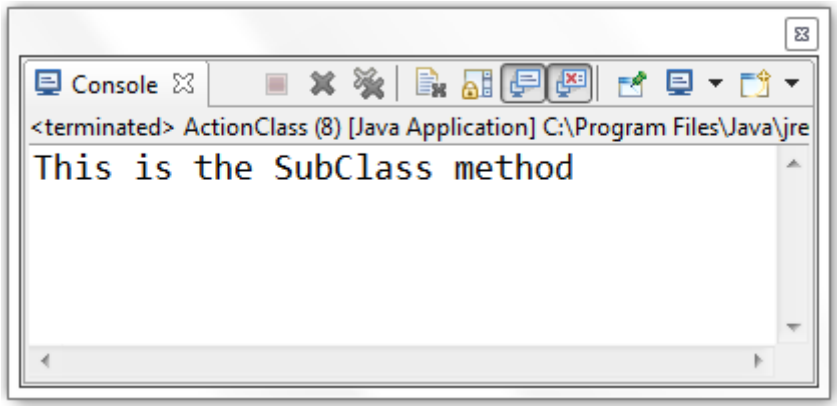
در کد فوق یک متد تحت عنوان showSubClass به معنی **"کلاس زیرمجموعه را نشان بده"** ساخته و به آن دستور داده ایم تا عبارت This is the SubClass method را معنی

"این متد کلاس زیر مجموعه است" را نمایش دهد. در نهایت کد کلاس ActionClass را به شکل زیر بازنویسی می کنیم:

```
public class ActionClass {

	public static void main(String[] args) {
		SubClass myObject = new SubClass();
		myObject.showSubClass();
	}
}
```

حال قصد داریم تا به برنامه خود دستور دهیم تا شیئ ساخته شده ما تحت عنوان myObject متد موجود در کلاس زیرمجموعه را تحت عنوان showSubClass را روی صفحه مانیتور نمایش دهد. حال برنامه را اجرا کرده و می بایست خروجی زیر را مشاهده نماییم:

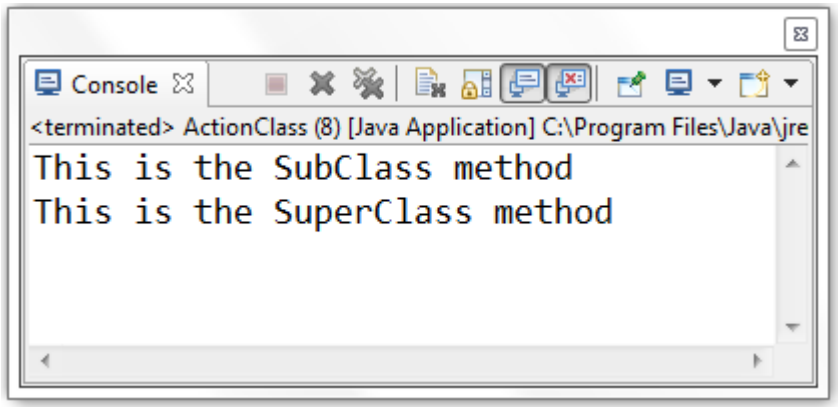


همانطور که انتظار می رفت هر دستوری که داخل متد showSubClass می باشد اجرا شده است. حال اگر بخواهیم با استفاده از کلید واژه super متد خود را مجبور کنیم تا متد موجود در کلاس اصلی را نیز نمایش دهد، می بایست کد کلاس زیر مجموعه را به شکل زیر تکمیل نماییم:

```
public class SubClass extends SuperClass {

	public void showSubClass() {
		System.out.println("This is the SubClass method");
		super.showSuperClass();
	}
}
```

در حقیقت با نوشتن کلید واژه super و قرار دادن یک نقطه و نوشتن نام متدی از کلاس اصلی که می خواهیم نمایش داده شود، برنامه خود را مجبور می کنیم تا اینکار انجام دهد. اکنون برنامه را مجدد Run می کنیم و انتظار می رود که خروجی زیر ایجاد گردد:



همانطور که مشاهده می شود علاوه بر متد موجود در کلاس SubClass متد موجود در کلاس SuperClass نیز با استفاده از دستور super فرا خوانده شده است.

پس از مطالعه این آموزش انتظار می رود بتوانیم به سؤالات زیر پاسخ بدهیم:

۱. چرا گاهی اوقات باید از دستور super استفاده کرد؟
۲. فایده این دستور در برنامه نویسی چیست؟
۳. به چه شکل از دستور super در مورد متغیرها و متدها بایستی استفاده کرد؟