

به نام خدا

آموزش سی و هفتم

اهداف آموزشی این قسمت عبارتند از:

۱. معرفی متدها در زبان برنامه نویسی جاوا
۲. آشنایی با اجزای تشکیل دهنده متدها
۳. آشنایی با وظیفه پارامترها

در این قسمت قصد داریم تا متدها را به طور دقیق تر مورد بررسی قرار دهیم. همانطور که قبلاً هم گفته شده هر شیء در زبان برنامه نویسی جاوا دارای یکسری Attribute و یکسری Behavior می باشد. به طور مثال اگر شیء ما دارای یک متغیر از جنس int است که مقداری را در خود ذخیره می سازد، این متغیر به منزله یک Attribute برای شیء ما محسوب می شود. از سوی دیگر Behavior ها بخشی از یک کلاس را تشکیل می دهند که قرار است کاری را انجام دهد که هر یک از این بخش ها یک Method نامیده می شود. تا این بخش از آموزش ما بارها و بارها متدهایی را به صورت ناخود آگاه مورد استفاده قرار داده ایم که یکی از این متدها print() می باشد که این وظیفه را دارا است که هر آنچه را که مابین دو پرانتز قرار گرفت را روی صفحه مانیتور به نمایش در آورد. متدها در زبان برنامه نویسی جاوا، هم به همراه Object ها و هم به همراه Class ها مورد استفاده قرار می گیرند.

نحوه ساخت یک متد تا حدودی شبیه به ساخت یک کلاس است. هر دو می توانند Modifier هایی مثل public, private و غیره بگیرند (برای آشنایی بیشتر با Modifier ها به آموزش سی و ششم مراجعه نمایید)، هر دو می بایست با یک علامت { شروع شده و با یک علامت } تکمیل گردند. متد وظیفه ای که بر عهده دارد این است که می بایست پس از آنکه فرا خوانده شد داده یا مقداری را بازگرداند. حال این مقدار می تواند مقدار یک متغیر ساده مثل int و یا double باشد و یا می تواند کلاسی از Object ها باشد.

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

حال برای درک بهتر متدها، یک پروژه جدید تحت عنوان 37th Session به معنی جلسه "سی و هفتم" ساخته و یک کلاس به نام MainClass به معنی "کلاس اصلی" در آن ایجاد می کنیم. حال کد ما در ابتدای کار می بایست به این شکل باشد:

```
public class MainClass {  
  
}
```

حال دو متغیر یکی از جنس int تحت عنوان price به معنی "قیمت" به مقدار اولیه 3000000 ایجاد کرده و یک شیء از روی کلاس String تحت عنوان laptopName به معنی "نام لپ تاپ" به مقدار اولیه The Laptop Name Is Lenovo. به معنی "نام لپ تاپ لنوو هست." ایجاد می کنیم. حال کد تکمیل شده ما می بایست به شکل زیر باشد:

```
public class MainClass {  
    int price = 3000000;  
    String laptopName = "Laptop Name Is Lenovo";  
  
}
```

حال متدی ایجاد می کنیم که وظیفه خواهد داشت این داده ها را در پنجره Console نمایش دهد. کد تکمیل شده به شکل زیر خواهد بود:

```
public class MainClass {  
    int price = 3000000;  
    String laptopName = "Laptop Name Is Lenovo";  
    public void showData() {  
        System.out.println(laptopName + " and its price is " + price);  
    }  
  
}
```

در واقع هر متد از دو بخش تشکیل شده است، بخش Signature و بخش Body که در مثال فوق Signature با کلید واژه public شروع شده است به این معنی که این متد از هر کجای برنامه ما قابل دسترسی خواهد بود. سپس کلید واژه void مشاهده می شود که به این معنا است که هر موقع متد showData در برنامه فرا خوانده شد، این متد قرار نیست هیچ چیزی به طور مثال داده ای را اصطلاحاً return کند یا بازگرداند و صرفاً وظیفه دارد هر آنچه را مابین دو علامت { } قرار گرفته است را مثلاً روی مانیتور نمایش دهد.

سپس به نام متد می رسم که در مثال فوق نام `showData` به معنی "داده ها را نشان بده" می باشد (همانطور که در قسمت های پیشین توضیح داده شد، نحوه نام گذاری متدها به این شکل است که کلمه اول در نام متد می بایست یک فعل باشد و علت این مسئله هم این است که از آنجا که متدها وظیفه دارند کاری انجام دهند، پس بهتر است در نامگذاری این دقت شود که نام متد هم با کاری که قرار است انجام دهد یکی باشد).

سپس دو پرانتز به شکل () مقابل نام متد قرار می دهیم که وظیفه این پرانتز ها این است که اگر خواستیم پارامترهایی را برای متد خود در نظر بگیریم داخل این دو پرانتز بنویسیم. در واقع ما با این کار می توانیم زمانیکه متد را فرا می خوانیم داده هایی را هر جنس در لحظه به متد خود منتقل نماییم. در این مثال ما قصد نداریم تا هیچ داده ای را به این شکل در حین فرا خواندن متد خود منتقل نماییم.

در آموزش سی و هشتم نمونه ای از متدی را توضیح خواهیم داد که داخل پرانتز مقابل نام متد داده ای را قرار داده تا در حین فرا خواندن آن متد، آن داده هم منتقل شود. حال نوبت به بررسی `Body` متد خود می رسد. در حقیقت همانطور که در کد فوق ملاحظه می شود `Body` متد ما تحت عنوان `showData` حاوی یک دستور `System.out.println()` برای نمایش داده چیزی روی صفحه مانیتور است. نکته ای که در اینجا حائز اهمیت است این است که در این دستور ما به متغیرهایی که قبلاً ایجاد کرده بودیم لینک داده ایم (به عبارت دیگر `Body` یک متد داخل دو کروشه قرار می گیرد).

تا این مرحله از کدنویسی، ما موفق شده ایم تا یک متد که قرار است جمله ای را روی صفحه مانیتور به نمایش درآورد را ایجاد کنیم. حال می بایست این متد را فرا بخوانیم، برای این منظور یک کلاس جدید تحت عنوان `ActionClass` به معنی "کلاس عملیاتی" ایجاد کرده و از آنجا که این کلاس جدید قرار است به منزله نقطه شروع برنامه باشد پس نیاز داریم تا در حین ساخت آن گزینه `public static void main` را تیک بزنیم. حال کد ما می بایست به صورت زیر باشد:

```
public class ActionClass {
    public static void main(String[] args) {
    }
}
```

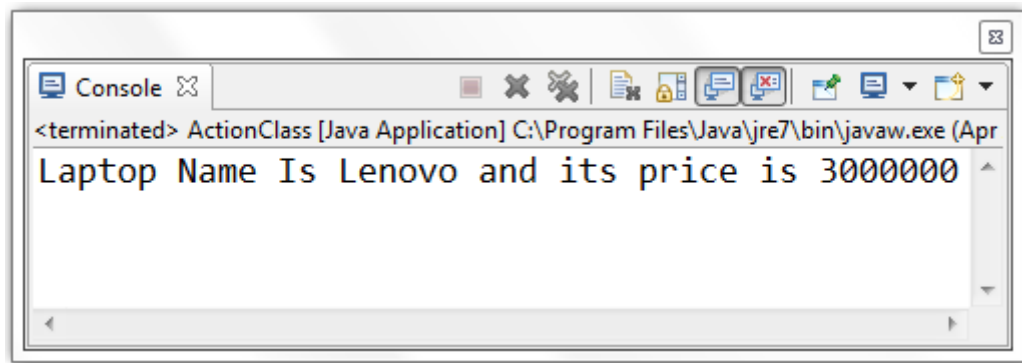
حال در متد main در کلاس جدیدی که ایجاد کردیم یک شیء از روی کلاس قبلی به صورت زیر می سازیم:

```
public class ActionClass {
    public static void main(String[] args) {
        MainClass myObject = new MainClass();
    }
}
```

در کد فوق ما نام کلاسی که می خواهیم از روی آن یک شیء ایجاد کنیم را نوشته سپس نامی برای آن انتخاب کرده که در این مثال نام myObject به معنی "شیء من" در نظر گرفته شده است. سپس یک علامت مساوی قرار داده و مجدداً نام کلاسی که می خواهیم شیء از روی آن ساخته شود را می نویسیم و در نهایت یک علامت (;) قرار می دهیم. شیء ما ساخته شد. حال می خواهیم متدی که در کلاسی که از روی آن یک شیء ساختیم را فرا بخوانیم. برای این منظور کد خود را به شکل زیر تکمیل می کنیم:

```
public class ActionClass {
    public static void main(String[] args) {
        MainClass myObject = new MainClass();
        myObject.showData();
    }
}
```

برای این منظور نام شیئی که جدیداً ساخته را نوشته سپس یک نقطه قرار داده و سپس نام متدی که در کلاس اصلی خود ساخته بودیم را می نویسیم و در نهایت علامت (;) را در انتهای نام متد قرار می دهیم. حال برنامه را اجرا می کنیم:



خواهیم دید که برنامه ما پس از اجرا، متدی را فراخواند که وظیفه داشت عبارت فوق را روی صفحه مانیتور به نمایش در آورد.

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

۱. وظیفه متدها چیست؟
۲. هر متد از چند بخش تشکیل شده است؟
۳. هدف از به کارگیری پارامترها چیست؟

در جلسه قسمت سی و هشتم خواهیم دید که به چه نحوی می توان در حین ساخت یک متد، داده هایی از هر جنس را به عنوان پارامتر برای آن متد در نظر گرفت.

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی