

به نام خدا

آموزش سی و نهم

اهداف آموزشی این قسمت عبارتند از:

۱. معرفی دستور return در زبان جاوا
۲. معرفی متدهایی که می توانند return داشته باشند

در قسمت سی و هشتم با نحوه ارسال یک پارامتر و تغییر آن در هر لحظه ای که تمایل داشتیم آشنا شدیم. در این قسمت قصد داریم تا مبحث return را مورد بررسی قرار دهیم. برای درک بهتر این موضوع بهتر است سناریویی که در قسمت پیشین در مورد رفتن به سوپر مارکت بیان کردیم را به خاطر آوریم. در واقع در آن مثال زمانی که پدر یا مادرمان از ما درخواست کرد که به سوپر مارکت برویم و نان یا پنیر یا هر چیز دیگری بخریم هیچ گونه دستوری مبنی بر اینکه پس از خرید نان بلافاصله آن را به خانه بیاور وجود نداشت. در زبان برنامه نویسی جاوا در مورد متدها این دستور از طریق کلید واژه return انجام می شود.

سناریویی که در این قسمت از آموزش قصد داریم مطرح کنیم به این صورت است که فرض کنیم که ما برای برنامه نویسی از یک PC استفاده می کنیم و از آنجا که قصد داریم پروژه های خود را به موقع به پایان برسانیم نیاز به خرید یک لپ تاپ داریم تا همواره همراه ما باشد که بتوانیم حتی در زمان هایی هم که در منزل نیستیم روی پروژه خود کار کنیم. نکته ای که در اینجا وجود دارد این است که نمی دانیم کدام مدل لپ تاپ برای کار ما مناسب است که از قبل هزینه آن را تهیه کرده و همراه خود به خرید ببریم. برای این منظور به مرکز خرید کامپیوتر رفته و جستجوی خود را آغاز می کنیم.

به منظور درک بهتر این موضوع، پروژه ای جدید تحت عنوان 39th Session به معنی "جلسه سی و نهم" ایجاد کرده و کلاسی به اسم MainClass به معنی "کلاس اصلی" در آن می سازیم. حال کلاس اصلی خود را به شکل زیر آغاز می کنیم:

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

```
public class MainClass {

    public int goToComputerComplex() {
        int laptopPrice = 0;
        return laptopPrice;
    }
}
```

در کد فوق یک متد از جنس `int` تحت عنوان `goToComputerComplex` به معنی "برو به مجتمع کامپیوتری" ایجاد کرده و متغیری از جنس `int` به نام `laptopPrice` به معنی "قیمت لپ تاپ" ایجاد کرده و مقدار اولیه آن را معادل با صفر قرار می دهیم.

پیش از نام متد کلید واژه `public` دیده می شود که همانطور که قبلاً هم گفته شد به این معنا است که این متد از هر جای برنامه قابل دسترسی خواهد بود. هدف از ایجاد متغیر `laptopPrice` این است که هزینه خرید لپ تاپ به آن اختصاص داده شود و از سوی دیگر این متغیر یک `Local Variable` است و این بدان معنا است که این متغیر از هر جای دیگری خارج از متدی که داخل آن قرار گرفته است غیر قابل دسترسی خواهد بود. به طور کلی هر زمانیکه بخواهیم متد ما داده ای را بازگرداند یا اصطلاحاً `return` کند، نیاز است تا نوع متغیری که می خواهیم مقدار آن `return` شود را پیش از نام متد خود بنویسیم. به طور مثال اگر بخواهیم که متد ما یک عدد اعشاری را `return` کند پس بایستی متغیر `double` را نوشته و یا اگر بخواهیم مقدار بازگشتی متد ما یک داده از جنس `true` یا `false` باشد پس می بایست پیش از نام متد متغیری از جنس `boolean` بنویسیم. در این مثال از آنجا که می خواهیم داده ای از جنس `int` بازگردانده شود، پس کلید واژه `int` را قبل از نام متد خود می نویسیم (در حقیقت نام متغیر به جای کلید واژه `void` قرار خواهد گرفت). اکنون نیاز داریم تا متد خود را از داخل کلاس دیگری فرا بخوانیم. برای همین منظور یک کلاس جدید تحت عنوان `ActionClass` به معنی "کلاس عملیاتی" ایجاد کرده و از آنجا که می خواهیم این کلاس به منزله نقطه شروع برنامه باشد، در حین ساخت آن گزینه `public static` `void main` را تیک می زنیم. حال کد کلاس ما می بایست به شکل زیر باشد:

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

هم اکنون نیاز است تا یک شیء از روی کلاس MainClass داخل کلاس ActionClass ایجاد کنیم. 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();
    }
}
```

همانطور که قبلاً آموزش داده شد یک شیء جدید تحت عنوان myObject به معنی "شیء من" از روی کلاس MainClass ایجاد می کنیم. حال با این کار به متد ساخته شده در کلاس MainClass هم دسترسی خواهیم داشت. همانطور که قبلاً گفته شد، متد goToComputerComplex داده ای را return خواهد کرد که جنس آن داده int است. از این رو یک متغیر جدید در کلاس ActionClass تحت عنوان totalPrice به معنی "قیمت کل" ایجاد کرده و مقدار آن را برابر با متد goToComputerComplex قرار می دهیم. حال کد ما می بایست به شکل زیر باشد:

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

به طور خلاصه هر مقداری که برای متغیر laptopPrice در متد goToComputerComplex واقع در کلاس MainClass در نظر گرفته شود، حال به متغیر totalPrice که در کلاس ActionClass قرار دارد return خواهد شد. اکنون برای آنکه بتوانیم مقدار return شده را روی صفحه مانیتور مشاهده کنیم، نیاز است تا از دستور System.out.println() استفاده کنیم. بنابراین کد ما به شکل زیر در خواهد آمد:

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

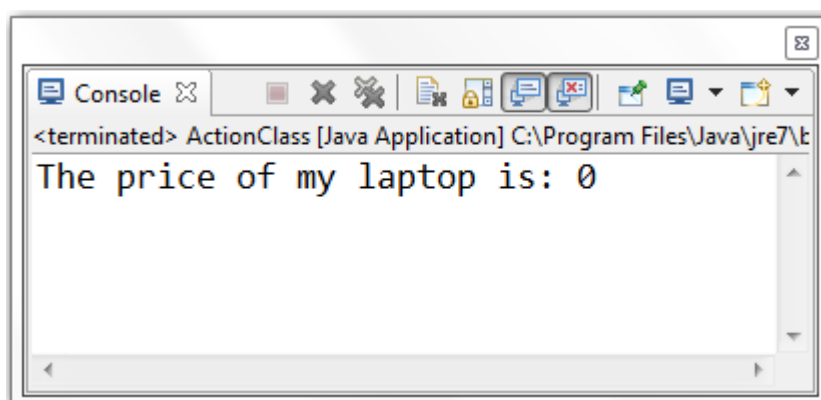
```

public class ActionClass {

    public static void main(String[] args) {
        MainClass myObject = new MainClass();
        int totalPrice = myObject.goToComputerComplex();
        System.out.println("The price of my laptop is: " + totalPrice);
    }
}

```

همانطور که در کد فوق مشاهده می شود، دستور `System.out.println()` را نوشته و چیزی را که می خواهیم روی صفحه مانیتور به نمایش در آید را داخل پرانتز آن قرار می دهیم. از آنجا که در این مثال می خواهیم مقدار متغیر `totalPrice` که به عبارتی همان مقدار `return` شده متغیر `laptopPrice` است را نمایش دهیم، پس نام `totalPrice` را داخل پرانتز می نویسیم. از آنجا که می خواهیم بدانیم که این مقدار نمایش داده شده متعلق به چیست، پیش از نام این متغیر عبارت `The price of my laptop is:` به معنی **"قیمت لپ تاپ من معادل است با:"** را که یک `string` است به آن اضافه می کنیم. حال پس از اجرای برنامه خروجی زیر مشاهده خواهد شد:



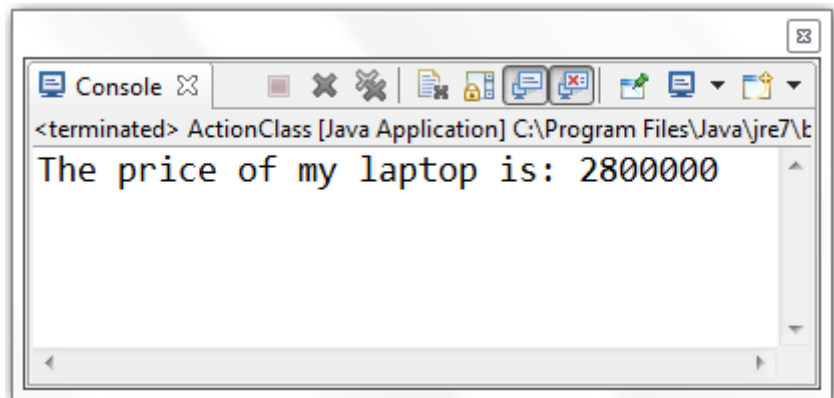
همانطور که در تصویر فوق مشاهده می شود برنامه به درستی کار کرده و مقدار `return` شده که هم اکنون معادل با صفر است را در کلاس `ActionClass` نمایش می دهد. حال مقدار `return` شونده را در کلاس `MainClass` را به عدد `2800000` تغییر می دهیم:

```
public class MainClass {

    public int goToComputerComplex() {
        int laptopPrice = 2800000;
        return laptopPrice;
    }

}
```

حال مجدد برنامه را اجرا می کنیم:



همانطور که ملاحظه می شود، به محض تغییر مقدار متغیر، مقدار `return` شونده از سوی متد `goToComputerComplex` نیز تغییر پیدا می کند. اکنون که قیمت لپ تاپ نهایی شد، اگر خود سر کار می رویم که می بایست به حساب بانکی مان رجوع کنیم و در غیر این صورت این مقدار `return` شده را به شماره همراه یکی از والدین خود مجدد `return` می کنیم تا این مبلغ را برای ما واریز کنند!

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

۱. منظور از کلید واژه `return` در زبان جاوا چیست؟
۲. آیا متدی از جنس `void` را می توان `return` کرد؟
۳. اگر بخواهیم متدی داده ای از جنس درست یا غلط را بازگرداند، آن متد بایستی از چه جنسی باشد؟