

به نام خدا

آموزش چهل و یکم

اهداف آموزشی این قسمت عبارتند از:

۱. معرفی Constructor در زبان برنامه نویسی جاوا

۲. آشنایی با کلید واژه this

به نظر می رسد پس از آشنایی با تفاوت مابین Local Variable ها و Field ها در زبان جاوا بتوانیم بهتر مفهوم Constructor ها را در زبان برنامه نویسی جاوا درک کنیم. برای همین منظور پروژه ای تحت عنوان 41st Session به معنی "جلسه چهارم و یکم" در محیط اکیپس ایجاد کرده و کلاسی تحت عنوان MyClass به معنی "کلاس من" در آن می سازیم. چنانچه خواهیم به طور خلاصه بگوییم که Constructor چیست، بایستی گفت که یک Constructor همانند یک متد است با این تفاوت که نام آن دقیقاً با نام کلاس یکی بوده و از سوی دیگر به هیچ وجه نمی تواند داده ای را return کند (به منظور آشنایی با دستور return به آموزش سی و هشتم مراجعه فرمایید). واقعیت دیگری که در مورد Constructor ها صدق می کند این است که یک Constructor می تواند هیچ پارامتری نداشته باشند، یک پارامتر داشته و یا حتی بیش از یک پارامتر داشته باشد.

اکنون ببینیم که کاربرد یک Constructor چیست. در واقع زمانی که ما یک شیء از روی کلاس خود می سازیم، Constructor آن کلاس فرا خوانده می شود. کار این Constructor این است که Field های آن شیء جدید را Initialize کرده یا به عبارت دیگر آغاز کند. نکته ای که در اینجا می بایست حتماً مد نظر قرار داده شود این است که ما برای ساخت یک شیء جدید از روی یک کلاس نیاز به یک Constructor داریم و اگر ما این Constructor را ننویسیم، خود Compiler در حین ساخت شیء جدید یک Constructor که فاقد هرگونه پارامتری باشد را برای ما در حین Compile شدن خواهد ساخت. این Constructor تحت عنوان Default Constructor یا **پیش فرض** شناخته می شود. کاری که این Constructor انجام می دهد این است که کلیه متغیرهای Field را با مقدار اولیه Default فرا می خواند (مقدار

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

اولیه صفر برای متغیرهای عددی، مقدار false برای متغیر boolean و مقدار null برای مواردی همچون string).

حال اگر ما یک Constructor برای کلاس خود طراحی کنیم Compiler هیچ گونه Constructor یی از جنس default به صورت خود کار نخواهد ساخت و Constructor ما را مورد استفاده قرار می دهد.

اکنون کد موجود در کلاس خود را به شکل زیر تکمیل می کنیم:

```
public class MyClass {
    String name;
    int age;
    boolean trueOrFalse;

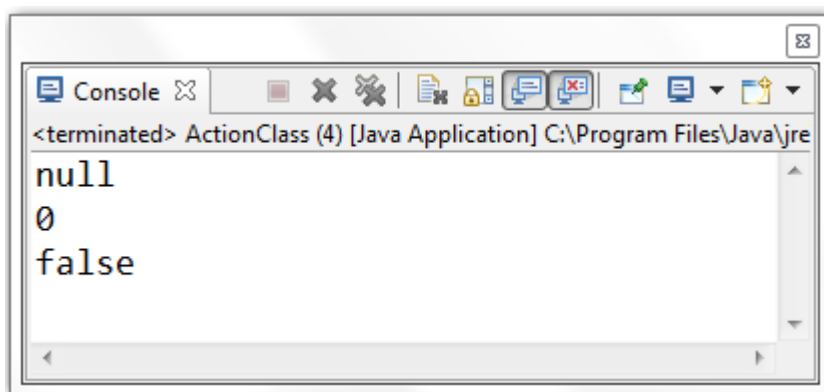
    public void display() {
        System.out.println(name);
        System.out.println(age);
        System.out.println(trueOrFalse);
    }
}
```

در کد فوق همانطور که مشاهده می شود سه Instance Variable یا متغیرهای Field یکی از جنس متغیر عددی، یکی از جنس متغیر boolean و یکی هم از جنس کلاس String تحت عناوین name و age و trueOrFalse به معنی به ترتیب "نام، سن و درست یا غلط" ساخته ایم. سپس یک متد تحت عنوان display به معنی "نمایش بده" ساخته و در آن دستور داده ایم تا Field های ایجاد شده در داخل کلاس را روی صفحه مانیتور نمایش دهد. حال نوبت به ساخت کلاس دیگری می رسد که به منزله نقطه شروع برنامه باشد. به عبارت دیگر این کلاس جدید می بایست دارای متد main باشد. برای این منظور کلاس دیگری تحت عنوان ActionClass به معنی "کلاس عملیاتی" ساخته و در حین ساخت آن گزینه public static void main را تیک می زنیم. پس از ساخت این کلاس، یک شیء از روی کلاس MyClass ساخته و نام آن را object به معنی "شیء" می گذاریم و در نهایت از طریق شیء ساخته شده متد موجود در کلاس MyClass را فرا می خوانیم:

```
public class ActionClass {

    public static void main(String[] args) {
        MyClass objectOne = new MyClass();
        objectOne.dispaly();
    }
}
```

حال برنامه خود را اجرا می کنیم:



همانطور که در تصویر فوق مشاهده می شود، Compiler به صورت خود کار یک Default Constructor برای شیئی ساخته شده ما از روی کلاس MyClass ایجاد می کند و همانطور که قبلا گفته شد کار این Constructor این است که مقادیر پیش فرض را برای Field های کلاس MyClass را ایجاد کند (البته این Constructor از آنجا که در حین Compile شدن ساخته می شود از دید ما مخفی است). در تصویر فوق مقدار پیش فرض null برای کلاس String بوده، مقدار پیش فرض صفر برای متغیر int بوده و مقدار پیش فرض آخر که معادل با false است برای متغیر boolean می باشد.

در حقیقت خود Compiler جاوا این Constructor را برای شیئی ما ساخته است. حال ببینیم Compiler به چه شکلی این Constructor را در کلاس ما ایجاد کرده است. برای این منظور کد کلاس MyClass را به شکل زیر تکمیل می کنیم:

```

public class MyClass {
    String name;
    int age;
    boolean trueOrFalse;

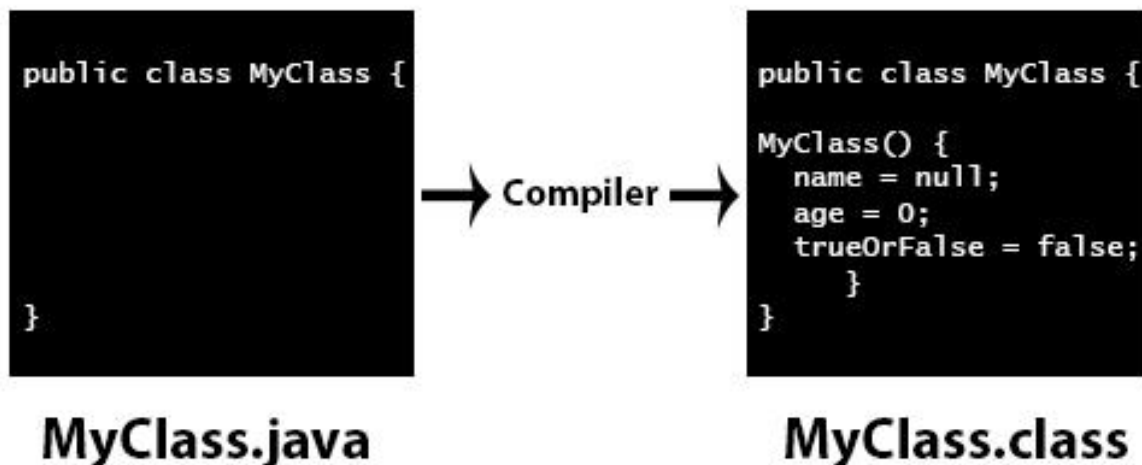
    MyClass() {
        name = null;
        age = 0;
        trueOrFalse = false;
    }

    public void dispaly() {
        System.out.println(name);
        System.out.println(age);
        System.out.println(trueOrFalse);
    }
}

```

همانطور که در کد فوق مشاهده می شود، یک Constructor تحت عنوان MyClass(){ } ساخته ایم (همانطور که در بالا گفته شده نحوه ساخت Constructor همانند ساخت یک متد است با تفاوت که نام آن می بایست دقیقاً هم نام با کلاس بوده و هیچ گونه مقدار بازگشتی مثل int و double و String و غیره نمی بایست برای آن تعریف شود).

اکنون داخل Constructor ساخته شده برای Field های کلاس خود مقدار اولیه تعریف می کنیم. مقدار Default برای string ها همواره null بوده، برای متغیرهای عددی مثل int همواره صفر بوده و برای متغیر boolean معادل با false می باشد. به طور خلاصه نوشتن و ننوشتن این Constructor هیچ فرقی برای ما نخواهد کرد چرا که Compiler این کار را به عهده گرفته است. به عبارت دیگر اگر ما برای شیئی جدیدی که می سازیم مقادیر اولیه در نظر نگیریم، Compiler در حین Compile کردن برنامه این کار را به صورت خود کار برای ما انجام خواهد داد (این کار در تصویر زیر نشان داده شده است).



برای این که تست کنیم که Compiler این کار را به صورت خودکار انجام می دهد یا خیر می توان Constructor ساخته شده را Comment کرده و برنامه را مجدد اجرا کنیم و خواهیم دید که بازهم همان مقادیر اولیه توسط Compiler ایجاد می شوند. هم اکنون می خواهیم تا دیگر به Compiler اجازه ندهیم تا مقادیر Default را برای Field ها در نظر گیرد، بلکه خود این مقادیر را تعریف کنیم. برای این منظور کد کلاس MyClass را به شکل زیر تغییر می دهیم:

```

public class MyClass {
    String name;
    int age;
    boolean trueOrFalse;

    MyClass(String myName, int myAge, boolean myTrueOrFalse) {
        name = myName;
        age = myAge;
        trueOrFalse = myTrueOrFalse;
    }

    public void dispaly() {
        System.out.println(name);
        System.out.println(age);
        System.out.println(trueOrFalse);
    }
}

```

همانطور که در کد فوق ملاحظه می شود، سه پارامتر دقیقاً از همان جنسی که Field ها هستند برای Constructor در نظر گرفته ایم به نام های myName, myAge, myTrueOrFalse و هر کدام از آن ها را معادل با هم تایی خود که از جنس Field است قرار داده ایم. علت اینکه نام پارامترهای Constructor خود را با نام Field های یکی قرار نداده ایم این است که اگر هر دو نام یکی باشد، Compiler در حین Compile کردن مقادیر Field ها را مد نظر قرار خواهد داد. کد فوق بدون هیچ مشکلی اجرا خواهد شد اما این در حالی است که راه کاری برای اینکه نام پارامترها و Field ها یکی باشد نیز وجود دارد و آن هم استفاده از کلید واژه this می باشد. برای درک بهتر این موضوع کد فوق را به شکل زیر بازنویسی می کنیم:

```
public class MyClass {
    String name;
    int age;
    boolean trueOrFalse;

    MyClass(String name, int age, boolean trueOrFalse) {
        this.name = name;
        this.age = age;
        this.trueOrFalse = trueOrFalse;
    }

    public void display() {
        System.out.println(name);
        System.out.println(age);
        System.out.println(trueOrFalse);
    }
}
```

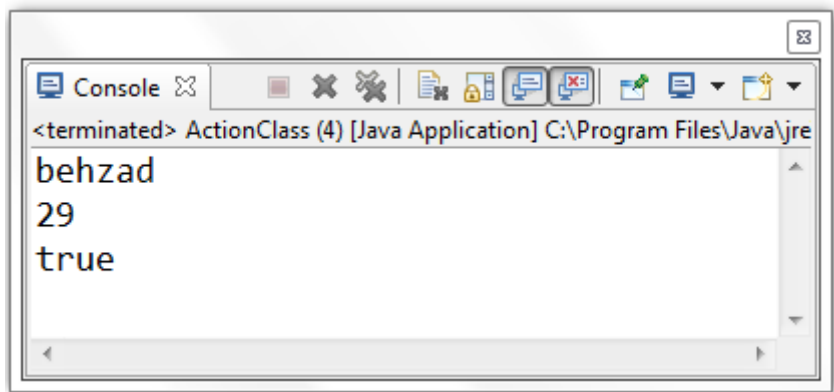
همانطور که ملاحظه می شود کلید واژه this را قرار داده سپس یک نقطه می گذاریم و نام پارامتر مد نظر را پس از نقطه قرار می دهیم و در نهایت یک علامت مساوی قرار داده و نام Field خود را پس از آن می نویسیم و در انتها یک علامت ; قرار می دهیم. با این کار به Compiler دستور می دهیم که مثلاً this.name منظور name یی است که پارامتر است و یا this.age منظور از age همان age یی است که داخل پارامتر متد نوشته شده است نه age یی که به عنوان متغیر Field در نظر گرفته شده است. از سوی دیگر نام های name, age,

trueOrFalse که پس از علائم مساوی قرار گرفته اند همان Field هایی هستند که برای کلاس MyClass در نظر گرفته ایم.

اکنون پس از آنکه مقدار Field های کلاس MyClass را از طریق یک Constructor که خودمان ایجاد کرده ایم را به پارامترها نسبت داده ایم می توانیم برنامه خود را اجرا کنیم. برای همین منظور به کلاس ActionClass رفته و کد خود را به شکل زیر تکمیل می کنیم:

```
public class ActionClass {  
  
    public static void main(String[] args) {  
        MyClass objectOne = new MyClass("behzad", 29, true);  
        objectOne.dispaly();  
    }  
}
```

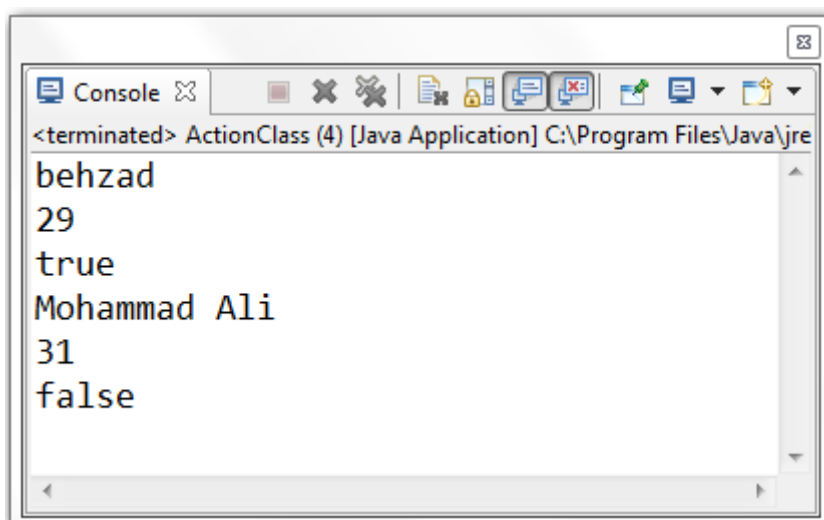
اگر خاطرمาน باشد در کلاس MyClass زمانیکه Constructor خود را ایجاد می کردیم سه پارامتر به ترتیب از جنس String و int و boolean ایجاد کردیم. حال در کلاس ActionClass با قرار دادن مقادیری داخل پرانتزها متناظر با جایگاه پارامترها (یعنی اول یک داده از جنس string و دوم یک داده از جنس متغیر int و سوم یک داده از جنس boolean)، می توانیم مقدار اولیه پارامترها را تعیین کرده و بالتبع از آنجا که مقدار پارامترها برابر با مقدار Field ها در نظر گرفته شده است، از این رو مقدار اولیه Field ها تعیین خواهد شد (به طور خلاصه همانطور که در کد فوق مشاهده می شود مقدار "Behzad" برای پارامتری از جنس string تحت عنوان name در نظر گرفته شده است و از آنجا که مقدار Field یی تحت عنوان name برابر با this.name یا همان پارامتری که از جنس string است قرار داده شده است، پس مقدار "Behzad" به Field کلاس MyClass تحت عنوان name که از جنس کلاس String است اختصاص خواهد یافت. دو پارامتر دیگر را نیز به همین شکل می توان تفسیر کرد). پس از اجرای برنامه خروجی زیر مشاهده خواهد شد:



همانطور که می بینیم با موفقیت توانستیم مقدار اولیه Field های کلاس اول را مقدار دهی کنیم. اکنون فرض کنیم که از روی کلاس MyClass دو عدد شیء داریم، یکی تحت عنوان objectOne و دیگری تحت عنوان objectTwo. برای مقدار دهی هر دوی آن ها کد خود را به شکل زیر تکمیل می کنیم:

```
public class ActionClass {  
  
    public static void main(String[] args) {  
        MyClass objectOne = new MyClass("behzad", 29, true);  
        objectOne.dispaly();  
  
        MyClass objectTwo = new MyClass("Mohammad Ali", 31, false);  
        objectTwo.dispaly();  
    }  
}
```

همانطور که مشاهده می شود در کد فوق دو شیء از روی کلاس MyClass تحت عناوین objectOne و objectTwo به معنی به ترتیب "شیء یک" و "شیء دو" ساخته ایم. از آنجا که می خواهیم شیء دوم ما هم دارای مقادیر اولیه ای باشد، سه پارامتر با مقادیر Mohammad Ali و 31 و false را برای آن در نظر می گیریم. حال برنامه را اجرا می کنیم:



```
<terminated> ActionClass (4) [Java Application] C:\Program Files\Java\jre
behzad
29
true
Mohammad Ali
31
false
```

همانطور که در تصویر فوق مشاهده می شود هر دو شیء ساخته شده از روی کلاس MyClass با مقادیر اختصاصی خودشان در پنجره Console نمایش داده شده اند.

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

۱. منظور از Constructor در زبان برنامه نویسی جاوا چیست؟
۲. چرا از Constructor ها در جاوا می بایست استفاده کنیم؟
۳. مقادیر پیش فرض برای متغیرها در Constructor چیست؟
۴. چگونه می توان پارامتری را برای یک Constructor تعیین کرد؟
۵. چگونه می توان پارامترهای یک Constructor را با نام Field ها یکی کرد؟