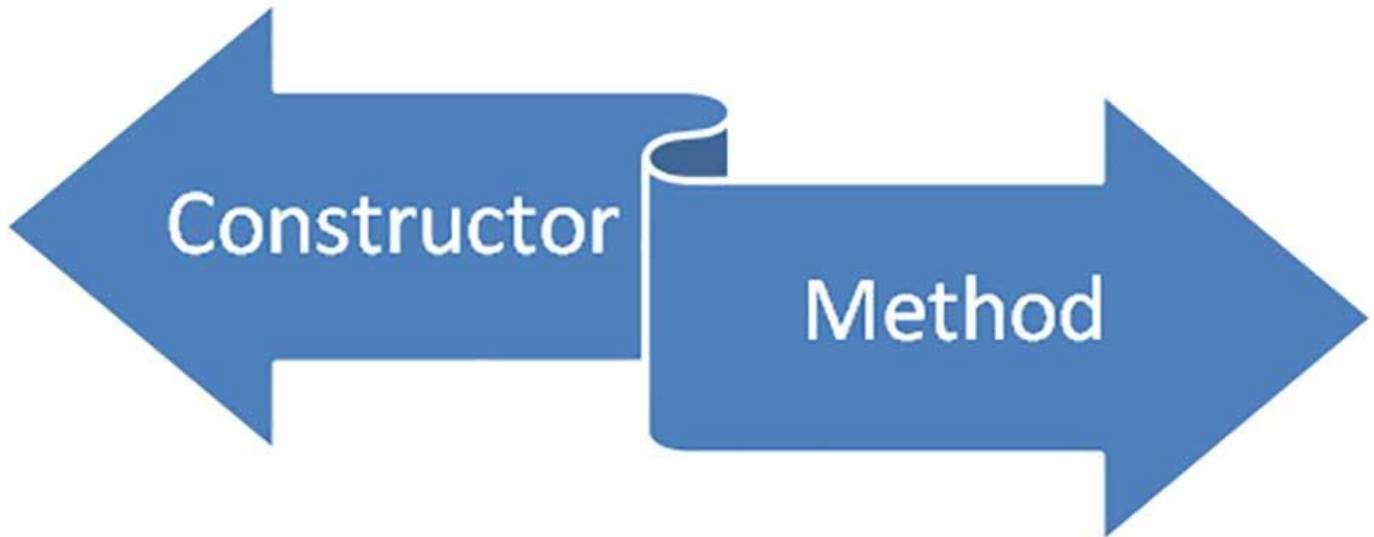


جلسه 6 - متد سازنده یا متد Constructor



متد سازنده که همان متد Constructor نامیده میشود، یک متد است که در درس های قبلی درباره آن صحبت کردیم. این متد درون کلاس قرار میگیرد و همنام با همان کلاس است. هر کلاس باید حتما یک متد Constructor داشته باشد. ما خودمان باید در زمان ساختن یک کلاس این متد را هم برای آن بنویسیم. اگر متد سازنده را خودمان ننویسیم، زبان جاوا یک متد پیشفرض و خالی را بصورت اتوماتیک برای کلاس هایی که ما میسازیم در نظر میگیرد. برای آشنایی با متد سازنده با **برنامه چی** همراه باشید.

دیدن این مقاله در سایت

در این مقاله سرفصل های زیر بررسی میشوند:

وظیفه متد سازنده (متد Constructor) چیست؟

قوانین سینکس متد سازنده

متد Constructor بدون آرگومان

Constructor های پارامتری

وظیفه متد سازنده (متد Constructor) چیست؟

متد سازنده یا همان متد Constructor وظیفه دارد اشیاء را هنگامی که ساخته میشوند، مقدار دهی اولیه (یا به اصطلاح Initialize) کند. این متد هم نام کلاس خودش میباشد و از لحاظ سینتکس، مانند یک متد محسوب میشود. اما متد Constructor هیچگونه مقدار برگشتی مشخصی ندارد. یعنی هنگام ساختن آن هیچ مقدار برگشتی را برای آن تعریف نمیکنیم. (برای متد سازنده حتی void را هم مشخص نمیکنیم).

به طور معمول، ما از متد Constructor برای دادن مقادیر اولیه به متغیرهای نمونه که درون کلاس قرار دارند، استفاده میکنیم. همچنین همه اعمال اولیه ای که نیاز داریم انجام بدهیم تا یک Object یا شیء ساخته شود را درون آن قرار میدهیم.

مقدار دهی اولیه یا چیزی که در برنامه نویسی به آن Initialize گفته میشود، یک عمل ضروری برای متغیرهای نمونه میباشد. زیرا اگر این کار را نکنیم و این متغیرها در طول برنامه نویسی مقدار دهی نشوند، هنگام اجرا شدن کدها، برنامه شما خطای معروف Null Pointer Exception را به شما نشان میدهد. کلا در برنامه نویسی جاوا باید حواستان به متغیرهای نمونه و آبجکتها باشد که بدون مقدار باقی نمانند. عمل Initialize در متد سازنده، به شما کمک میکند که روی این فرآیند کنترل خوبی داشته باشید. یک پیش بینی که درباره برنامه نویسی جاوا میتوانم بکنم این است که در آینده خطای Null Pointer Exception را به وفور خواهید دید.

همانطور که اشاره کردیم چه شما متد سازنده را برای کلاس تعریف کنید چه نکنید، این متد برای کلاس شما ساخته میشود. حتی اگر جاوا هم متد Constructor را بصورت اتوماتیک برای کلاس شما معین نکند، تمامی متغیرهای نمونه که در کلاس شما وجود دارند، برابر با "صفر" مقدار دهی اولیه خواهند شد. اما اگر خودتان متد Constructor را برای کلاس بسازید، دیگر از مقدار صفر استفاده نمیشود.

قوانین سینکس متد سازنده

در زیر قواعد نوشتن یک متد سازنده را مشاهده میکنید.

```
class ClassName{  
    ClassName() {  
    }  
}
```

زبان جاوا اجازه میدهد دو نوع متد سازنده داشته باشید:

- Constructor های بدون آرگومان
- Constructor های پارامتری

متد Constructor بدون آرگومان

همانطور که از نام این متد ها مشخص است، هیچگونه پارامتری را قبول نمیکنند. اگر از این مدل از Constructor ها استفاده کنید، متغیر های نمونه برای همه اشیائی که از روی کلاس ساخته میشوند با یک سری مقادیر ثابت، Initialize خواهند شد.

مثال

```
Public class MyClass {  
    Int num;  
    MyClass() {  
        num = 100;  
    }  
}
```

شما باید برای مقدار دهی اولیه Object باید مانند زیر، متد سازنده را صدا بزنید.

```
public class ConsDemo {
public static void main(String args[]) {
MyClass t1 = new MyClass();
MyClass t2 = new MyClass();
System.out.println(t1.num + " " + t2.num);
}
}
```

این کدها نتیجه زیر را تولید خواهند کرد:

```
100 100
```

Constructor های پارامتری

در بیشتر مواقع، شما نیاز به یک متد Constructor دارید که چندین پارامتر را دریافت کند. این پارامترها همانگونه که در متد های معمولی وارد میشوند، در متد سازنده هم وارد خواهند شد. فقط کافیست این پارامترها را درون پرانتز، بعد از نام متد بنویسید.

مثال

```
// Yek Method Constructor sade.
class MyClass {
int x;

// Dar zir method constructor ra mibinid.
MyClass (int i) {
x = i;
}
}
```

در کدهای بالا، یک مثال ساده استفاده از Constructor را میبینید. شما باید برای مقدار دهی اولیه اشیاء، متد سازنده را مانند زیر صدا بزنید:

```
public class ConsDemo {  
    public static void main(String args[]) {  
        MyClass t1 = new MyClass( 10 );  
        MyClass t2 = new MyClass( 20 );  
        System.out.println(t1.x + " " + t2.x);  
    }  
}
```

نتیجه باید عبارت زیر باشد:

10 20

در جلسه بعدی چه چیزی یاد خواهید گرفت؟

در جلسه بعدی از دوره آموزش جاوا، با انواع داده ای پایه و مبتدی در زبان برنامه نویسی جاوا آشنا خواهید شد. جلسه بعد پایه ای خواهد شد که از طریق آن میتوانید کار با متغیرها در جاوا را به خوبی یاد بگیرید. با ما همراه باشید.

جلسه بعد: دیتا تایپ های جاوا

جلسه قبل: کلاس جاوا و شیء (object)