

کلاس و اشیاء در جاوا

جاوا یک زبان برنامه نویسی شی گرا (OOP) است. در این آموزش با شی گرایی و نحوه ایجاد کلاس و اشیاء دلخواه در برنامه آشنا می شوید.

جاوا یک زبان برنامه نویسی شی گرا است و این امکان را به شما می دهد تا با ایجاد اشیاء ، مشکلات پیچیده را به مجموعه های کوچک تر تقسیم کنید. این اشیاء دو ویژگی دارند:

- حالت
- رفتار

بباید چند مثال بزنیم:

۱- لامپ یک شی است

- می تواند در حالت خاموش یا روشن باشد.
- می توانید لامپ را روشن و خاموش کنید. (رفتار)

۲- دو چرخه یک شی است

- دارای دنده ، دو چرخ ، تعدادی چرخ دنده و غیره است. (حالت)
- قابل ترمز گرفتن ، شتاب دادن ، تغییر دنده و غیره است. (رفتار)

کلاس جاوا

قبل از ایجاد اشیاء در جاوا ، باید یک کلاس تعریف کنید.

کلاس یک طرح برای شی است. می توانیم کلاس را به عنوان یک طرح (نمونه اولیه) یک خانه تصور کنیم. این شامل تمام جزئیات در مورد طبقات ، درب ، پنجره ها و غیره است. بر اساس این توصیفات خانه را می سازیم. خانه شی است.

از آنجا که بسیاری از خانه ها با همان توصیف ساخته می شوند ، می توانیم اشیاء زیادی را از یک کلاس ایجاد کنیم.

چگونه کلاس را در جاوا تعریف کنیم؟

در اینجا نحوه تعریف یک کلاس در جاوا آورده شده است:

```
1. class ClassName {  
2. // variables  
3. // methods  
4. }
```

مثال دیگر:

```
1. class Lamp {  
2. // instance variable  
3. private boolean isOn;  
4. // method  
5. public void turnOn() {  
6. isOn = true;  
7. }  
8. // method  
9. public void turnOff() {  
10. isOn = false;  
11. }  
12. }
```

در اینجا ، ما یک کلاس به نام Lamp تعریف کردیم.

کلاس دارای یک متغیر نمونه (متغیر تعریف شده در داخل کلاس) isOn و دو متد turnOn() و turnoff() است. این متغیرها و متد های تعریف شده در یک کلاس ، اعضای کلاس نامیده می شوند.

در برنامه فوق به دو کلمه کلیدی ، `public` و `private` توجه کنید. این اصطلاحات سطح دسترسی را مشخص می کنند که در آموزش های بعد توضیح خواهیم داد. در حال حاضر ، فقط به یاد داشته باشید:

- کلمه کلیدی `private` متغیرها و متدها را خصوصی می کند که فقط در داخل همان کلاس قابل دسترسی هستند.
- کلمه کلیدی `public` متغیرها و متدها را عمومی می کند که در خارج از کلاس هم قابل دسترسی هستند.

در برنامه فوق متغیر `isOn` از نوع `private` است در حالی که روشهای `turnOn()` و `turnoff()` از نوع `public` هستند.

اگر سعی کنید از خارج از کلاس به اعضای خصوصی دسترسی پیدا کنید ، کامپایلر خطا نشان می دهد.

اشیاء جاوا

وقتی کلاس تعریف می شود ، فقط مشخصات شی تعریف می شود. هیچ حافظه یا محل ذخیره ای اختصاص داده نمی شود.

برای دسترسی به اعضای تعریف شده در کلاس ، باید شی ایجاد کنید. در قطعه کد زیر از کلاس `Lamp` شی ایجاد کردیم.

```
1. class Lamp {
2.     boolean isOn;
3.     void turnOn() {
4.         isOn = true;
5.     }
6.     void turnOff() {
7.         isOn = false;
8.     }
9. }
10. class ClassObjectsExample {
11.     public static void main(String[] args) {
12.         Lamp l1 = new Lamp(); // create l1 object of Lamp class
13.         Lamp l2 = new Lamp(); // create l2 object of Lamp class
14.     }
15. }
```

این برنامه دو شیء 11 و 12 از کلاس Lamp ایجاد می کند.

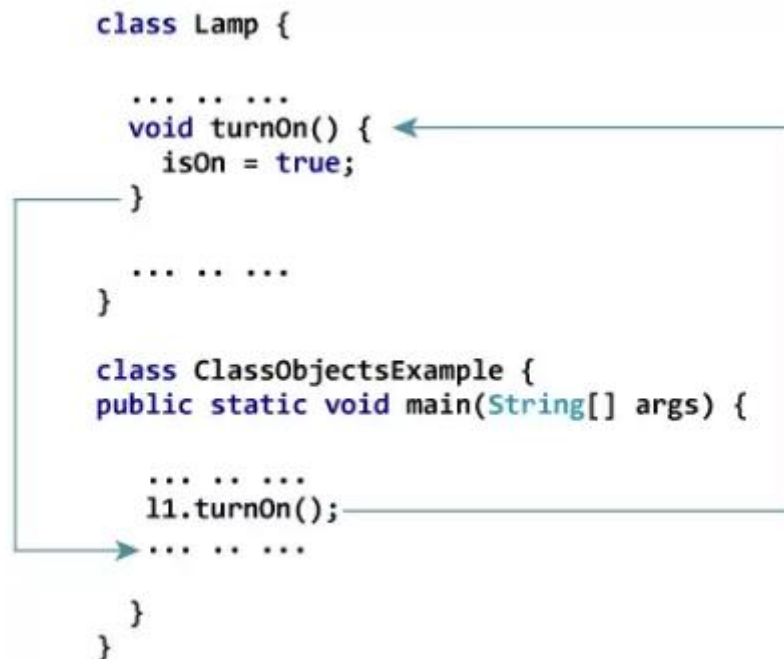
چگونه به اعضا دسترسی پیدا کنیم؟

با استفاده از (.) می توانید به اعضا (توابع و متغیرها) دسترسی پیدا کنید. مثلاً،

```
l1.turnOn();
```

این عبارت متد turnOn() را در کلاس Lamp برای شیء 11 فراخوانی می کند.

هنگامی که با استفاده از عبارت بالا متدی را فراخوانی می کنید، تمام عبارات موجود در بدنه ی متد turnOn() اجرا می شوند. سپس ، کنترل برنامه به زیر عبارت l1.turnOn() می پرد.



به طور مشابه ، به متغیر نمونه می توان دسترسی داشت:

```
l2.isOn = false;
```

توجه به این نکته مهم است که ، اعضای private فقط در داخل کلاس قابل دسترسی هستند. اگر کد l2.isOn = false; در تابع main() (خارج از کلاس Lamp) قرار بگیرد ، کامپایلر خطا نشان می دهد.

مثال: کلاس و اشیاء در جاوا

```
1. class Lamp {
2.     boolean isOn;
3.     void turnOn() {
4.         isOn = true;
5.     }
6.     void turnOff() {
7.         isOn = false;
8.     }
9.
10.    void displayLightStatus() {
11.
12.        System.out.println("Light on?" + isOn);
13.    }
14. }
15. class ClassObjectsExample {
16.     public static void main(String[] args) {
17.
18.         Lamp l1 = new Lamp();
19.         Lamp l2 = new Lamp();
20.
21.         l1.turnOn();
22.         l2.turnOff();
23.
24.         l1.displayLightStatus();
25.         l2.displayLightStatus();
26.     }
27. }
```

خروجی

```
Light on? true
Light on? false
```

در برنامه بالا ،

- کلاس Lamp ایجاد می شود.
- کلاس دارای یک متغیر نمونه isOn و سه متد turnOn() ، turnOff() و displayLightStatus() است.
- دو شیء l1 و l2 از کلاس Lamp در تابع main() ایجاد می شوند.
- در اینجا ، متد turnOn() با استفاده از شیء l1 صدا زده می شود l1.turnOn()
- این متد متغیر isOn از شیء l1 را به صورت true تنظیم می کند.
- و متد turnOff() با استفاده از شیء l2 صدا زده می شود l2.turnOff()
- این متد متغیر isOff از شیء l2 را به صورت false تنظیم می کند.

- سرانجام ، `l1.displayLightStatus()` عبارت `Light on? true` را چاپ می کند. زیرا متغیر `isOn` برای شی `l1` به صورت `true` تنظیم شده است.
 - و `l2.displayLightStatus` عبارت `Light on? False` را چاپ می کند. زیرا متغیر `isOn` برای شی `l2` به صورت `false` تنظیم شده است.
- وقتی اشیا مقدار دهی اولیه می شود ، هر شی شامل نسخه خاص خود از متغیرها و متدها است. به عنوان مثال ، متغیر `isOn` برای اشیاء `l1` و `l2` متفاوت است.