

بسم الله الرحمن الرحيم



آموزش برنامه نویسی اندروید – جلسه چهارم

کلاس ها در جاوا

کلاس یک ساختار منطقی است که تمامیت زبان جاوا بر آن استوار شده.
کلاس همچنین شکل دهنده ی برنامه نویسی شی گرا در جاوا میباشد.
یک کلاس قالبی از یک شی است و یک شی نمونه ای از یک کلاس است.

تعریف

```
Class CalssName{  
    Type MethodName(Parameters){  
        //body of method  
        //...  
    }  
}
```

نحوه استفاده :

```
Class Box{  
    double width;  
    double height;  
    double depth;  
}
```

نحوه صدا زدن :

```
Box mybox = new Box(); // create a Box object called mybox
mybox.width=100;
mybox.height=200;
```

واژه کلیدی

this

گاهی لازم است یک روش به شی ای که آن را فراخوانی نموده ارجاع نماید.
در هر جایی که ارجاع به یک شی داریم ، در صورتی که مجاز باشد استفاده میکنیم.
مثلا در این متد :

```
Box(double w, double h, double d){
this.width=w;
this.height=h;
this.depth=d;
}
```

مخفی نمودن متغیر نمونه:

همانطور که میدانیم داخل یک بلاک کد نمیتوانیم دو متغیر همنام داشته باشیم.
اما وقتی یک متغیر نمونه باشد ، میتوان از همان نام برای متغیر های دیگر استفاده کرد.
بطور مثال :

```
Box(double width, double height, double depth){
this.width=width;
this.height=height;
this.depth=depth;
}
```

دستورات کنترلی

همانطور که در جلسه اول گفتیم ، برای این سری از آموزش های برنامه نویسی اندروید لازم است که تا حدودی با برنامه نویسی آشنا باشیم.

پس دیگه فکر نمیکنم نیازی به توضیحات زیادی باشه در این رابطه.

فقط یک مرور ساده میکنیم.

– انتخاب

– تکرار

– پرش

– انتخاب

```
If ( condition 1 ) statement 1 ;  
else if ( condition 2 ) statement 2 ;  
...  
else statement n ;
```

```
switch ( expression ){  
case value1 :  
    statements;  
    break;  
case value2 :  
    statements;  
    break;  
...  
default :  
    statements;  
    break;  
}
```

– تکرار

```
While ( condition ){  
// body of loop  
}
```

```
Do{  
// body of loop  
} While ( condition );
```

```
For ( [initialize] ; [ Boolean expression] ; [ iterates]; )  
{  
    statements;  
}
```

آشنایی با آرایه ها :

آرایه یک بعدی

```
Type var-name [ ] ;
```

مثال :

```
Int mounth_days [ ] ;
```

آرایه با اندازه مشخص :

```
Array-var = new type [ size ] ;
```

مثال :

```
Mounth_days = new int [12] ;
```

دسترسی به خانه های آرایه :

```
Mounth_days[10] = 30 ;
```

```
System.out.println(Mounth_days[3]);
```

روش دیگری برای این کار وجود دارد. توجه کنید :

```
Int Mounth_days [ ] = { 31,28,31,30,31,30,31,31,30,31,30,31};
```

آرایه های چند بعدی :

```
Int twoD[ ][ ] =new int [4][5] ;
```

یک شکل دیگر هم برای اعلان آرایه ها وجود دارد :

```
Type [ ] var-name ;
```

```
int a1[ ] = new int [3];
```

```
int [ ] a2 = new int [3];
```

```
Char twod1 [ ][ ] = n;
```

متد (یا روش) :

متد ها تقریباً شبیه به همان توابع در زبان های برنامه نویسی دیگر است.

```
Type name(parameters){  
//body of method  
}
```

اگر نوع متد

void

نباشد باید یک مقدار از تابع بازگردانده شود ، که باید بصورت زیر را داخل بلاک کد متد قرار گیرد.

Return value;

حالا میخواهیم کلاس

Box

را کمی تکمیل کنیم:

```
Class Box {  
//Properties  
double width;  
double height;  
double depth;  
//Methods  
void Volume() {  
System.out.print("Volume is : ");  
System.out.println( width * height * depth );  
}  
}
```

حالا میریم سراغ استفاده از این کلاس...

```
Class BoxDemo() {  
public static void main (string args[]){  
Box mybox1 = new Box();  
Box mybox2 = new Box();  
//assign values to mybox1  
mybox1.Depth=15;  
mybox1.height=20;
```

```
mybox1.width=10;  
//assign values to mybox2  
mybox2.Depth=3;  
mybox2.height=6;  
mybox2.width=9;  
//Display Volume 1  
mybox1.Volume();  
//Display Volume 2  
mybox2.Volume();  
}  
}
```

خروجی این برنامه بصورت زیر است :

```
Volume is 3000  
Volume is 162
```

خوب برای این جلسه هم تا همین جا کافیه.
ما را از نظراتتون محروم نکنید.
باتشکر.

Info@MastersCo.net
Develop@MastersCo.net
www.MastersCo.net
www.facebook.com/MastersCorporation
HosseinBadrnezhad@MastersCo.net