جزوه جهت آموزش مجازی

# برنامه نویسی موبایل

# جلسه ۲

مدرس:

زینب نادری

## ۲. شی گرایی و وراثت در جاوا

شی گرایی در جاوا باعث سازماندهی کدها می شود. شی گرایی در واقع سبکی از برنامه نویسی است که ساختار اصلی آن از شی ها تشکیل می شود. در شیوه برنامه نویسی شی گرا یا Object-Oriented Programming برنامه به شی گرایش دارد، به این معنا که داده ها و توابعی که قرار است بر روی داده ها اعمال شوند تا حد ممکن در قالبی به نام شی در کنار هم قرار می گیرند و نسبت به محیط بیرونی خودکپسوله می شوند.

منظور از شی یا Object یک جسم در دنیای واقعی مثل میز، کتاب، لپ تاپ و... است. برنامه نویسی شی گرا در واقع الگویی برای طراحی برنامه با استفاده از کلاس ها و اشیاء است.

**هر شی می تواند به عنوان یک کلاس تعریف شود.**

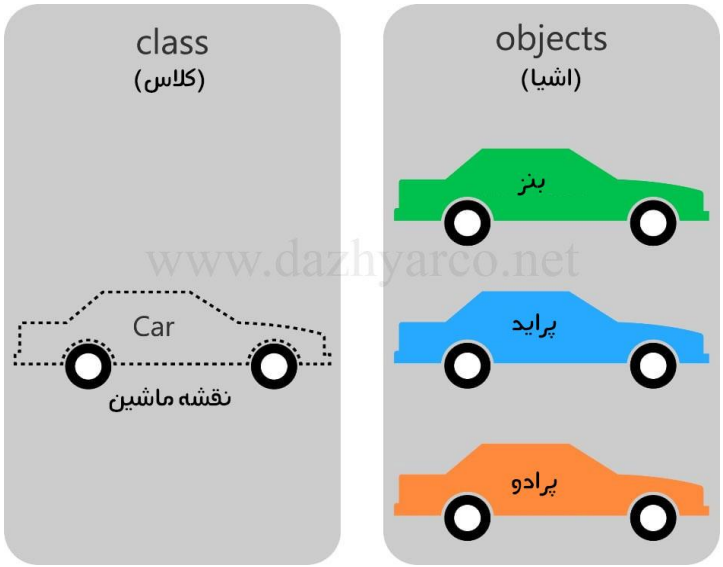
مفاهیم و اصطلاحات شی گرایی در تمامی زبان های برنامه نویسی یکسان است اما شاید نحوه کاربرد آنها با یکدیگر متفاوت باشند. این مفاهیم عبارت اند از: کلاس، شی، رفتار، صفت، ارث بری، کپسوله سازی ، چند ریختی ، انتزاع یا تجرید .

- **کلاس (Class)**

کلاس یک الگویی است که ما تعریف می کنیم که بتوانیم بر اساس آن یک Object را ایجاد کنیم.

- **شی (Object)**

اگر به عالم واقعیت نگاه کنید در عالمی زندگی می کنید پر از اشیا، مثل ماشین ،صندلی،چتر یا هر چیز دیگری که در دنیای واقعی می بینید یک شی است. شی (Object) نمونه هایی است که از روی کلاس می سازیم.



- **صفت(Attribute)**

یک خودرو رنگ دارد ،کمربند دارد ،تعدادی لاستیک دارد ،موتور دارد و الی آخر که به این ویژگی ها صفت می گویند.

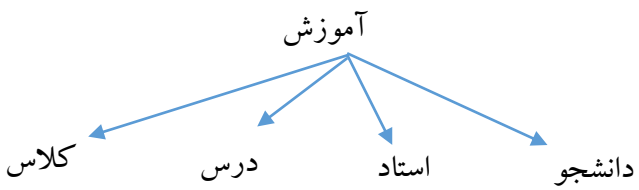

- **رفتار(Behavior)**

همین ماشین می تواند حرکت کند، سرعت بگیرد، ترمز کند، خاموش شود و الی آخر که به آن رفتار یا متد می گویند.

شی گرایی در جاوا می تواند به شما کمک کند تا در زمان خود صرفه جویی کنید و خطاهای برنامه را راحت تر و سریع تر شناسایی کنید. شی گرایی باعث می شود تکرار کدها تا حد زیادی کاهش یابد و در مقابل نیز خوانایی کدها به مراتب افزایش بیابد.

اگر بخواهیم رو راست باشیم باید بگوییم هر برنامه نویسی باید بر مفاهیم اساسی شی گرایی تسلط داشته باشد. ما در این جلسه قصد داریم شی گرایی در جاوا را به شما آموزش دهیم.

تفکر برنامه نویسی شی گرا از آنجا ناشی شد که زبان های برنامه نویسی برای حل مساله های دنیای واقعی طراحی و پیاده سازی شده اند و هر آنچه را که در دنیای واقعی وجود دارد می توان به عنوان یک شی تصور کرد. مانند برنامه ای که در دانشگاه مدیریت آموزش به کار گفته می شود.



خوب، برای نوشتن برنامه مبتنی بر مفهوم شی گرا ابتدا اشیا مرتبط با آن شناسایی می شود و سپس هر شی به طور مجزا تعریف می شود؛ به طور مثال برای ساخت شی دانشجو باید یک کلاس نوشت، سپس در طول برنامه می توانیم با استفاده از این کلاس هر تعداد شی دانشجو که می خواهیم، بسازیم.

## نحوه تعریف کلاس در جاوا

نحوه دسترسی class نام کلاس

{

تعریف متغییر ها و متدها

}

نحوه دسترسی که در تعریف بالا وجود دارد، مشخص می کند کلاس در چه جاهایی قابل استفاده است و میتواند دو مقدار زیر را بگیرد:

Public: کلاس توسط کلاسهای دیگر موجود در برنامه قابل دستیابی است.

Private: سایر کلاس ها اجازه دستیابی به این کلاس را ندارند.

**نکته:** نحوه تعریف متغییر و متد درون کلاس همانند تعریف آنها در برنامه اصلی است و تفاوتی ندارد و متد نیز همان تابع است که در جلسه قبل در مورد آن توضیح داده ایم.

**مثال:** می خواهیم شی دانشجو با دو ویژگی نام و نام خانوادگی تعریف کنیم و متدی هم در کلاس قرار دهیم که نام و نام خانوادگی آن را در کنار هم قرار داده و برگرداند.

```
Public   class   student
{
      String  name,family;
      Public string fullName()
      {
         string f = name + family;
         return f;
      }
}
```

## نحوه‌ی استفاده از کلاس

کلاس تعریف یک شیء است. کلاس می‌گوید که اشیایی که از روی آن ساخته می‌شوند، چه ویژگی‌ها و چه رفتاری دارند. اما یک کلاس چیزی جز یک تعریف نیست. برای استفاده از این تعریف باید یک نمونه یا

شیء از روی آن ساخت. برای مثال کلاس دانشجو که تعریف کردیم، به ما امکان می‌دهد که چندین دانشجو با نام و نام خانوادگی مختلف از روی آن بسازیم. می‌توانیم دانشجویی به نام stu1 با نام و نام خانوادگی علی مهدوی و دانشجویی دیگر با نام stu2 با نام مهدیار ستاری بسازیم و...

حال فرض کنید توی تابع main میخوایم از این کلاس استفاده کنیم که ابتدا یه نمونه به نام stu1 تعریف می کنیم و بعد به متغییر هایname و family مقداردهی کرده و در انتها با فراخوانی متد fullname مقدار نهایی نام+نام خانوادگی را چاپ می کنیم.

```
Public  TestClass
{
  public  static  void  main (String[] args)
  {
    Student stu1=new student();
    Stu1.name="Ali";
    Stu1.family="Mahdavi";
    System.out.println("StudentFullName:" +stu1.fullName());
  }
}
```

خروجی برنامه:

```
StudentFullName: Ali Mahdavi
```

**مثال :** کلاسی تعریف کنید که شعاع دایره را دریافت کند و محیط و مساحت آن را نمایش دهد.

```
public class Circle
{
   double r;
   public double mohit()
   {
      return (2 * Math.PI * r);
  }
  public double masahat()
  {
        return (Math.PI * r * r);
  }
}
```

در کلاس تعریف شده در بالا، ابتدا متغییر شعاع تعریف می شود. در متد اول محیط دایره را محاسبه کرده و مقدار آن را بر می‌گرداند. عبارتreturn مقدار مقابل خود را برمی‌گرداند. Math.PI همان عدد پی (۳٫۱۴)

معروف است. متد دوم هم مساحت دایره را محاسبه می کند و مقدار آن را بر می گرداند. حال می‌خواهیم برنامه‌ای بنویسیم و از کلاس فوق در آن استفاده کنیم. یک کلاس دیگر می‌نویسیم. این کلاس جدید شامل متد main() خواهد بود تا بتواند اجرا شود:

```
Public  class  Prog_Circle
{
    public  static  void  main(String[] args)
    {
        double radius = 5;
        Circle C = new Circle();
        System.out.println("محیط:" + C.mohit());
        System.out.println("مساحت:" + C.masahat());
    }
}
```

**سوال:** خروجی برنامه بالا چیست؟

## مفهوم وراثت در برنامه نویسی شی گرا

وراثت درجاوامکانیزمی است که توسط آن آبجکتی تمامی ویژگی (property) و قابلیت های (behavior) آبجکت پدر را کسب کرده و در اصطلاح به ارث ببرد .

ایده ی پشت وراثت در جاوا این است که برنامه نویس بتواند بر اساس کلاس های جاری(فعلی)، کلاس های جدید تولید نماید. زمانی که از کلاس جاری ارث بری می کنید، در واقع می توانید متدها و فیلدهای کلاس پدر را مجددا استفاده کرده و در صورت نیاز متدها و فیلدهای جدید اضافه نمایید .

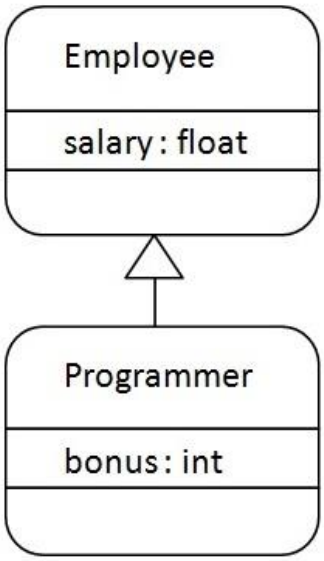
### دستور پیاده سازی وراثت در جاوا

```
class  نام زیرکلاس یاکلاس فرزند    extends    نام کلاس پدر
{
    متدها و فیلدها
}
```

کلیدواژه ی extend یک کلاس جدید ایجاد می کند که علاوه بر اعضای اختصاصی خود، فیلدها و متدهای کلاس پدر را به ارث می برد یعنی حق استفاده از فیلدها و متدهای پدر خود را هم دارد. واژه ی extend به معنای افزایش قابلیت های یک موجودیت است. در زبانJava، آن کلاسی که اعضای داخلی و محتوای آن به کلاس دیگری به ارث داده می شود، کلاس پدر یا کلاس ارشد (super/parent class)گفته می شود و آن

کلاسی که از کلاس پدر ارث بری می کند، در اصطلاح کلاس فرزند یا زیرمجموعه (subclass) خوانده می شود .

## مثالی از ارث بری در جاوا




همان طور که در تصویر بالا مشاهده می کنید، کلاس programmer زیرمجموعه و فرزند کلاس Employee یا کلاس پدر است.

```
Class   Employee
{
  float salary=40000;
}

Class  Programmer  extends  Employee
{
   int bonus=10000;
   public static  void  main(String args[])
   {
      Programmer p=new Programmer();
      System.out.println("Programmer salary is:"+p.salary);
      System.out.println("Bonus of Programmer is:"+p.bonus);
   }
}
```

خروجی:

```
Programmer salary is:40000.0
Bonus of programmer is:10000
```

در مثال بالا، آبجکت programmer می تواند علاوه بر فیلد کلاس خود به کلاس Employee دسترسی داشته باشد. برای همین است با اینکه نمونه p از نوع کلاس Programmer تعریف شده اما علاوه بر متغییر bonus توانسته به متغییر salary هم دسترسی داشته و آن را نمایش دهد.

**تمرین ۲:** خروجی کد زیر چیست؟خطا دارد؟

```
class Animal
{
    void eat()
    {
      System.out.println("eating...");
    }
}
class Dog extends Animal
{
    void bark()
    {
       System.out.println("barking...");
    }
}

class TestInheritance
{
   public static void  main(String args[])
   {
     Dog d=new Dog();
     d.bark();
     d.eat();
   }
}
```