



# آموزش برنامه نویسی جاوا

قسمت اول

مدرس: آقای چگانی

[ILikePHP.ir](http://ILikePHP.ir)



ILIKEPHP.IR

مرجع آموزش ویدیویی برنامه نویسی تحت وب



مدرس : آقای چگانی

آموزش JAVA قسمت اول

سلام

بنده چنگانی هستم. با آموزش برنامه سازی جاوا در خدمتون هستم.

امیدوارم که بتونم کمکی به دوستان بکنم و آموزش خوبی داشته باشیم.

قبل از اینکه شروع به آموزش جاوا کنیم، لازمه یک جلسه در مورد اصول شی گرای صحیح کنیم که بین تمامی زبان های برنامه نویسی (زبان های شی گرا) مشترک هستند. این جلسه خیلی خلاصه در مورد این ۴ اصل صحبت می کنیم. جلسات آموزشی ما هم بر اساس همین ۴ اصل ادامه پیدا می کنه. فقط این جلسه مرور خیلی سریعی بر روی این اصول داریم، بنابراین اگه این جلسه خیلی متوجه نشدین نگران نباشید.

هدف این کلاس این نیست که فقط یاد بگیریم چطوری با جاوا برنامه بنویسیم، قرار هست یاد بگیریم چطور به صورت شی گرا فکر کنیم و با زبان جاوا پیاده سازی کنیم.

در کنار آموزش سعی میکنم پایان هر جلسه چند مورد از قواعد نگارشی استاندارد جاوا رو هم بگم که با استاندارد نویسی جاوا هم آشنا بشیم.

این جلسه پس مفاهیم شی گرای رو به صورت مختصر توضیح میدیم و از جلسه بعدی میریم سراغ خود جاوا

بریم سراغ برنامه نویسی شی گرای یا Object-oriented programming

OOO چیست؟

اگه بخواهیم خیلی ساده توضیح بدم، یک نوع روش برنامه نویسی هست که در آن برنامه نویس از اشیا و ایده های جهان واقعی استفاده میکنه و اصول برنامه نویسی شی گرا هم از جهان واقعی گرفته شده اند.

نکته ای که هست اینکه به همون اندازه که یک طراحی شی گرای خوب باعث کاهش پیچیدگی و مدل کردن ساده مسئله هست، یک طراحی نامناسب با استفاده از OOO به فاجعه رو به وجود میاره. پس به طراحی خوب لازم هست.



مدرس : آقای چگانی

## آموزش JAVA قسمت اول

این نکته رو هم بگم که اصلا دنبال این نیستیم که شی گرایبی محض داشته باشیم، ولی هدف اصلی ما این هست که ۴ اصل اساسی شی گرایبی رو در کنار هم در پیاده سازی استفاده کنیم.

اما این ۴ اصل:

اصول برنامه نویسی شی گرا (oop)

۱- تجرید یا انتزاع (Abstraction)

۲- کپسوله سازی (Encapsulation)

۳- ارث بری (Inheritance)

۴- چندریختی (Polymorphism)

شاید در مورد این اصول خیلی شنیده باشید و استفاده هم بکنید ولی گفتنش خالی از لطف نیست

۱- تجرید یا انتزاعی (Abstraction)

در oop ما سه مفهوم کلاس (class) و موجودیت (Object) و نمونه (instance)

همه موجودیت ها در یک الگو شی گرا باید به صورت انتزاع یا abstract باشه. یعنی اینکه توی دنیای واقعی همچین موجودیتی وجود نداشته باشه و فقط توی ذهن ما باشه.

مثلا انسان، کاربر، کتاب.

در واقع وقتی شما کلمه انسان رو می شنوید یه مدلی توی ذهنتون شکل میگیره که دارای یه سری خصیصه (attribute) و توانایی (method) هست ولی شخص خاصی توی ذهنتون نیست. در واقع شما دارین به صورت انتزاعی فکر می کنید. هدف در شی گرایبی این هست که این مفهوم و طرز فکر را پیاده سازی کنیم.



مدرس : آقای چگانی

## آموزش JAVA قسمت اول

پس تا اینجا موجودیت‌های سیستم ما همیشه انتزاعی هستند. هر موجودیت یک کلاس هست ولی هر کلاس یک موجودتی نیست. بعضی از کلاس‌ها برای این ساخته می‌شوند که ابزاری در اختیار ما بگذارن. مثلا کلاس‌های `utils` یا کلاس `math` و ...

توی یک سیستم ما میتونیم `n` تا موجودیت داشته باشیم. ما میتونیم از هر کدوم از این موجودیت‌ها به هر تعداد نمونه داشته باشیم. مثلا از موجودیت انسان می‌تونیم نمونه محمد، علی و ... را داشته باشیم. دقت کنید برعکس موجودیت که انتزاعی بودند نمونه‌ها باید واقعی باشند و بتونیم به یک عامل واقعی نسبت بدیم.

### ۲- کپسوله‌سازی (Encapsulation)

این مفهوم خیلی خیلی نزدیک به مفهوم جعبه‌سیا (`black box`) هست. در واقع ما باید موجودیت‌ها یا کلاس‌ها رو به نحوه‌ای پیاده سازی کنیم که فقط ورودی و خروجی‌ها برای اعضای بیرونی قابل نمایش باشند و اعضای داخلی کلاس محافظت شوند.

برای مثال یک اتومبیل رو فرض کنید، همه ما میدونیم این اتومبیل داری فرمان و گاز و ... هست و میدونیم در صورت استفاده از اون‌ها چه اتفاقاتی رخ می‌دهد ولی اینکه این اتفاقات چطوری انجام می‌شود اطلاع نداریم. مثلا اگه گاز بدهیم می‌دونیم ماشین جلو میرود ولی اینکه چه اتفاقاتی می‌افتد تا ماشین جلو برود نمی‌دایم و برای ما هم مهم نیست.

در واقع در طراحی موجودیت‌ها و کلاس‌ها هم باید به این نکته توجه کرد که باید چندین لایه از این کپسوله سازی رعایت شود و فقط اجزایی قابل دسترسی از بیرون باشند که نیاز هست و سایر اجزا محدود شوند. که تعریف این سطح دسترسی در جاوا به صورت `public` یا `private` یا `protected` و یا محدودیت در سطح یک پکیج ایجاد میکنیم.

کپسوله سازی این کمک رو میکنه که بدون متوجه شدن از پیچیدگی‌های بخش‌های مختلف بتونیم از اون‌ها استفاده کنیم. کپسوله سازی می‌تونه تا چندین لایه انجام شود و هرچه بیشتر شود به اصطلاح گفته میشود ماژولارتر می‌شود.

### ۳- ارث‌بری (Inheritance)

توضیحات این قسمت به موقع خودش کامل گفته میشه فقط این مورد رو در مورد ارث‌بری باید بدونیم اینکه ارث‌بری با اهداف زیادی اضافه شده ولی مهمترین اون‌ها به این برمیگرده که قابلیت سلسه‌مراتب بین موجودیت‌ها را به ما می‌دهد. مورد بعدی اینکه قابلیت بازنویسی یک موجودیت را به ما می‌دهد. که کامل این دو مورد و موارد دیگه بحث می‌شود.



این مورد هم شبیه مورد ارث‌بری در زمان خودش به طور کامل توضیح داده می‌شه فقط اگه بخواهیم توضیح مختصری بدیم اینکه این قسمت تاکید بر روی این داره که ما به موجودیت‌ها و کلاس‌ها قابلیت بدیم. یعنی اینکه مثلا بگوییم این موجودیت قابلیت A را دارد ولی اینکه چگونه آن را انجام می‌دهد مد نظر ما نباشد و هر نمونه می‌تواند به هر نحوه‌ای این قابلیت A را پیاده‌سازی کند ولی همه این نمونه‌ها قابلیت A را دارا می‌باشند.

مثلا یک انسان، پرنده و یک خزنده را در نظر بگیرین. همه قابلیت حرکت کردن را دارند ولی نحوه پیاده‌سازی آن‌ها متفاوت هست. انسان راه می‌رود، پرنده پرواز می‌کند.

این مفهوم همان چند ریخته‌گی هست که یه قابلیت به چند شکل ظاهر می‌شود. البته این یک نمود از چندریختگی هست که بیشتر در موردش صحبت میشه.

تا اینجا ما چهار اصل شی‌گرایی رو گفتیم

که در همه زبان‌های شی‌گرا یکسان هست

در جلسه‌ای بعدی نحوه پیاده‌سازی و استفاده از این اصول رو می‌گیم

امیدوارم که این جلسه که بیشتر توضیح بود و کدنویسی نداشت خسته کننده نبوده باشه

از جلسه بعدی وارد کد میشیم

ایده‌وارم مفید بوده باشه

اگه سوالی بود بنده در خدمتم