

اصول و مبانی برنامه‌نویسی



مجید شبیری

کارشناسی ارشد IT، گرایش شبکه
از دانشگاه صنعتی امیرکبیر



برنامه‌نویسی شیء گرا

- (۱) شیء گرایی (object oriented)
- (۲) شیء (object)
- (۳) کلاس (class)
- (۴) اجزای کلاس
- (۵) نمونه سازی (instantiation)
- (۶) بسته بندی (Encapsulation)
- (۷) چندریختی (Polymorphism)
- (۸) وراثت (Inheritance)
- (۹) مزایا و کاربرد وراثت
- (۱۰) انواع وراثت
- (۱۱) نمودار کلاس (Class Diagram)

■ شیء گرایی یک سبک یا الگوی طراحی (paradigm) است.

■ برنامه نویس برای پیاده سازی **موجودیت های دنیای واقعی**، و خلق محصولی **بهینه و اصولی**، ابتدا باید **مدل سازی و طراحی** انجام دهد.

■ برنامه نویس برای پیاده سازی مسائل دنیای واقعی و ساخت نرم افزار، ابتدا باید **مدل سازی دقیقی از مسئله** داشته باشد.

■ مدل سازی بستگی به **زاویه دید** ما دارد و به روش های مختلف قابل انجام است.

■ توصیه **الگوی شیء گرایی** این است که در یک مسئله در مرحله مدل سازی و طراحی، باید به صورت مجموعه ای از **اشیاء (object)** دیده شود که

با هم در تعامل هستند.

■ شیء (Object) یک عنصر (المان) دنیای واقعی است که با در نظر گرفتن **نیازهای مسئله**، مدل سازی می شود

■ اجزاء یک آبجکت

○ **هویت (identity)** که با یک مقدار منحصر بفرد (نام) آبجکت را از آبجکت های دیگر سیستم متمایز می کند

○ **حالت (state)** که مقدار کنونی **ویژگی (property)** یا **خصیصه (attribute)** های آبجکت در آن ذخیره می شود

○ **رفتار (behavior)** که **فعالیت (activity)** های قابل مشاهده از آن شیء را به شکل تغییرات روی state نشان می دهد

○ یک آبجکت ممکن است **وجود فیزیکی و قابل لمس** داشته باشد (مانند کامپیوتر، خودرو، کارمند، مشتری و ...) یا اینکه **وجود آن مفهومی**

و غیر قابل لمس باشد (مانند پروژه، حساب بانکی، مسابقه و ...)

■ کلاس (class) عبارت است از مجموعه‌ای از آبجکت‌ها که:

○ ویژگی یکسان دارند.

○ رفتار مشترک از خود نشان می‌دهند.

■ برای مثال، همه مشتریان یک شرکت، دارای ویژگی و رفتار یکسان هستند و می‌توان آنها را به شکل آبجکت‌هایی دید که در قالب یک کلاس می‌توان آنها را مدل‌سازی کرد.

■ کلاس مانند نقشه یا طرح اولیه (blueprint) است.

■ کلاس توصیف آبجکت‌هایی را ارائه می‌دهد که می‌توانند از آن کلاس ساخته (create) شوند.

○ **خصیصه (attribute)** های آبجکت‌هایی که می‌توانند از کلاس نمونه‌برداری شوند.

- هر آبجکت که از کلاس ساخته می‌شود، مقدار (value) خصیصه‌هایش با بقیه متفاوت است.
- به خصیصه‌های کلاس، دیتای کلاس (class data) نیز گفته می‌شود.

○ **عملیات (operation)** ای که رفتار آبجکت‌های کلاس را مدل‌سازی می‌کند.

- به این عملیات، **فانشکن (function)** یا **متد (method)** نیز گفته می‌شود.

○ **مثال: حساب بانکی**

○ **خصیصه‌ها:** شناسه حساب - موجودی - تاریخ افتتاح - وضعیت فعال بودن

○ **عملیات:** افتتاح حساب - بستن حساب - واریز به حساب - برداشت از حساب - فعال/غیرفعال کردن حساب

- آبجکت، یک نمونه (instance) از کلاس است.
- به ساخت یک شیء (آبجکت) از کلاس، **نمونه سازی (instantiation)** گویند.
- در فرآیند **نمونه سازی**، در واقع به یک آبجکت، هویت واقعی می بخشیم و از کلاس که به صورت **نقشه و طرح** بود برای ساخت یک **موجودیت واقعی**، با ویژگی های خاص استفاده می کنیم.

- بسته‌بندی، یکی از بندهای الگوی طراحی شیء‌گرا است.
- تأکید بسته‌بندی روی تجميع attribute ها و method های آبجکت‌های هم خانواده، داخل یک ساختار به نام کلاس (class) است.

○ ویژگی‌های بسته‌بندی

- برخی جزئیات داخلی کلاس می‌تواند از بیرون غیر قابل مشاهده شود.
- خصیصه‌هایی که از بیرون قابل مشاهده نیستند فقط از طریق فانکشن‌های کلاس قابل مشاهده و تغییر هستند.
- دسترسی به عناصر کلاس از بیرون فقط از طریق اینترفیس (interface) امکانپذیر خواهد بود.

- چندریختی نیز یکی از بندهای الگوی طراحی شیء گرا است.
- تأکید چندریختی روی امکان داشتن آبجکت‌هایی با ساختار داخلی متفاوت ولی اینترفیس یکسان می‌باشد.
- این مفهوم در واقعیت از طریق **وراثت** (inheritance) پیاده‌سازی می‌شود.

○ مثال

- می‌توانیم یک کلاس داشته باشیم که ویژگی‌ها و رفتار یک **وسیله نقلیه** را توصیف کند.
- برای ساخت نوع خاصی از **وسیله نقلیه** (برای مثال **خودرو**) می‌توان از کلاس **وسیله نقلیه** یک نسخه دیگر ایجاد کرد (چندریختی) که برای مثال **فانشکن حرکت کردن** را به شیوه خاصی که مختص آن **وسیله نقلیه** خاص است پیاده‌سازی می‌کند. ولی همچنان نام **فانشکن حرکت کردن** است و تغییری در اینترفیس نخواهیم داشت.

حال از روی این طرح (کلاس) می‌توان چندین **خودرو** (آبجکت) جدید تولید (نمونه‌سازی) کرد.

○ وراثت، مکانیزمی است که اجازه می دهد با توسعه (extend) و بهبود (refine) قابلیت های کلاس های موجود، کلاس های جدیدی ساخته شود.

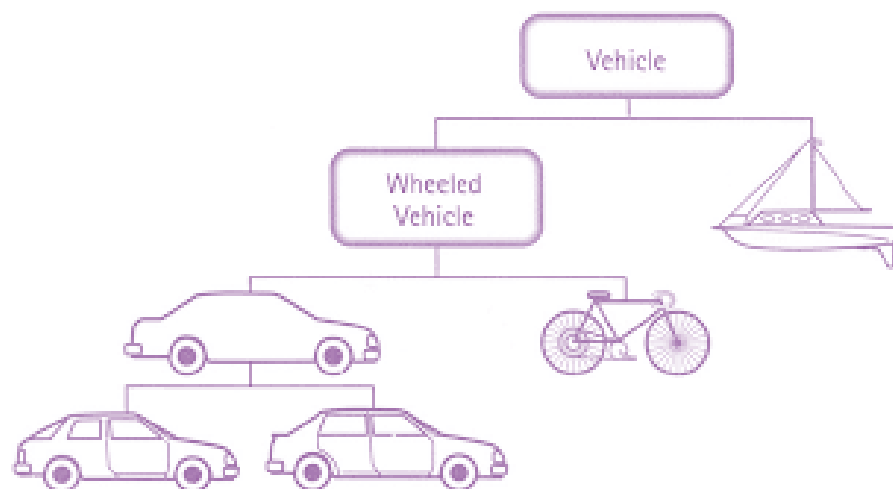
● **کلاس موجود** : پایه (base) ، والد (parent) ، سوپر (super)

● **کلاس جدید** : مشتق (drived) ، فرزند (child) ، زیر کلاس (subclass)

○ کلاس فرزند، متدها و خصیصه های کلاس (های) پدر را به ارث می برد.

○ کلاس فرزند، می تواند خصیصه ها و متدهای شخصی خود را به کلاس اضافه کند یا متدهای کلاس پدر را تغییر (modify) دهد.

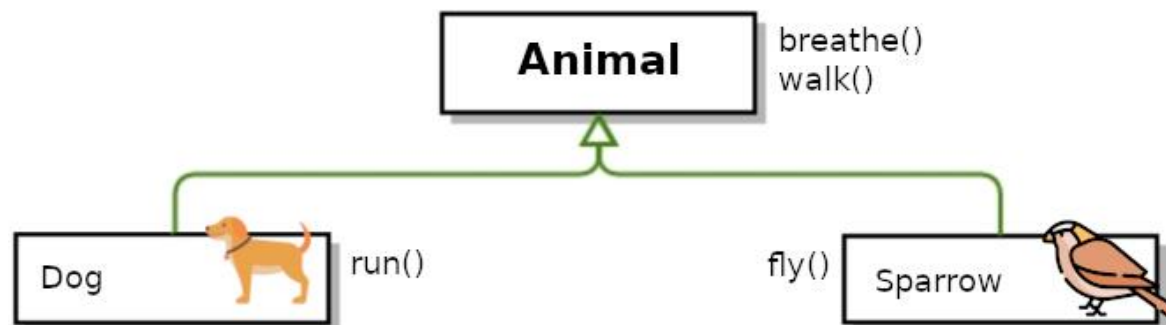
- کلاس vehicle با خصوصیات و رفتار عمومی و مشترک وسیله‌های نقلیه به عنوان کلاس پایه (پدر).
- کلاس Wheeled Vehicle از کلاس vehicle مشتق می‌شود و تمام ویژگی‌های عمومی یک وسیله نقلیه را دارد ولی ویژگی‌ها و رفتارهای جدیدی را نیز اضافه کرده و حتی رفتار حرکت کردن را به شیوه خاص خودش (حرکت روی چرخ) پیاده‌سازی کرده است.



■ کلاس Animal با خصوصیات و رفتار عمومی و مشترک حیوانات (تنفس و راه رفتن) به عنوان کلاس پایه (پدر).

■ کلاس Sparrow از کلاس Animal مشتق شده و تمام ویژگی‌های عمومی یک Animal را دارد ولی ویژگی و رفتارهای جدیدی را نیز اضافه کرده

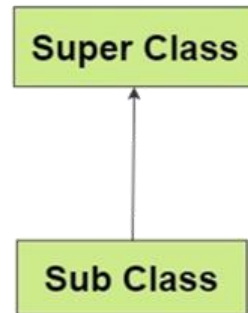
(fly) و حتی رفتار walk (راه رفتن) را به شیوه خاص خودش پیاده‌سازی کرده است.



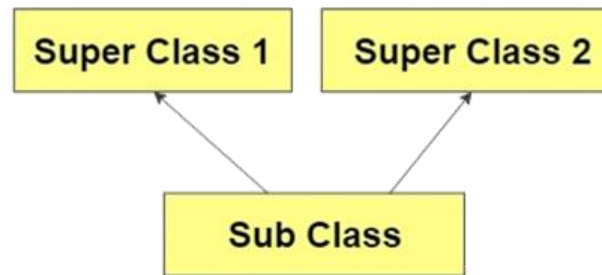
۱. استفاده مجدد از کدهای نوشته شده (reusability) و جلوگیری از دوباره‌نویسی کد
۲. جلوگیری از تکرار کدهای مشابه
۳. کاهش حجم کد برنامه
۴. افزودن ویژگی‌های جدید به کلاس بدون تغییر پیاده‌سازی کلاس پایه
۵. افزایش سرعت کدنویسی و توسعه نرم افزار
۶. تسریع در فرآیند نگهداری سورس کد (بروزرسانی و اعمال تغییرات در کد)
۷. کاهش احتمال اشتباهات کدنویسی.

$A \rightarrow B$	منفرد (Single)
$A, B \rightarrow C$	چندگانه (Multiple)
$A \rightarrow B \rightarrow C$	چندسطحی (Multilevel)
$A \rightarrow B, C$	سلسله مراتبی (Hierarchical)
متشکل از انواع وراثت	ترکیبی (Hybrid)

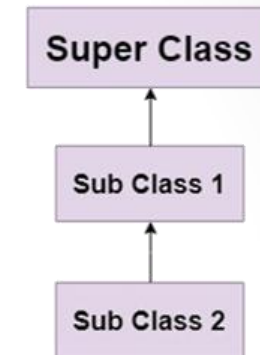
منفرد (Single)



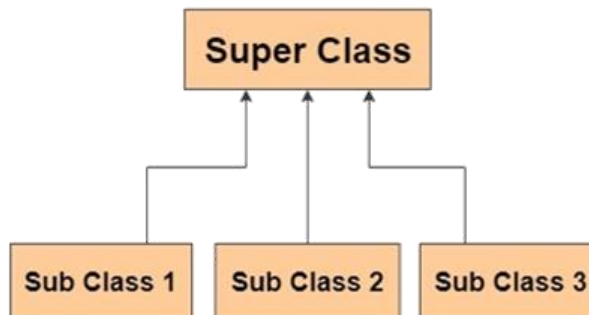
چندگانه (Multiple)



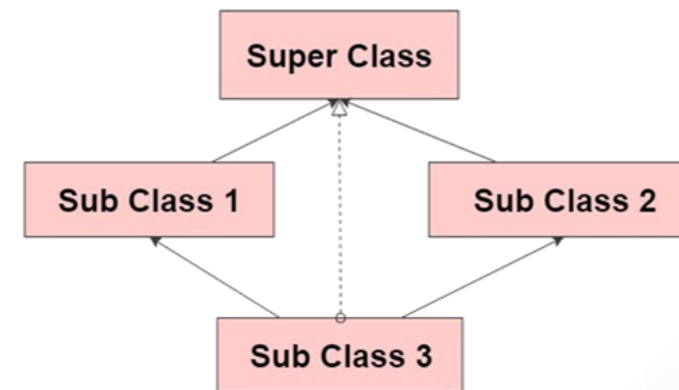
چندسطحی (Multilevel)

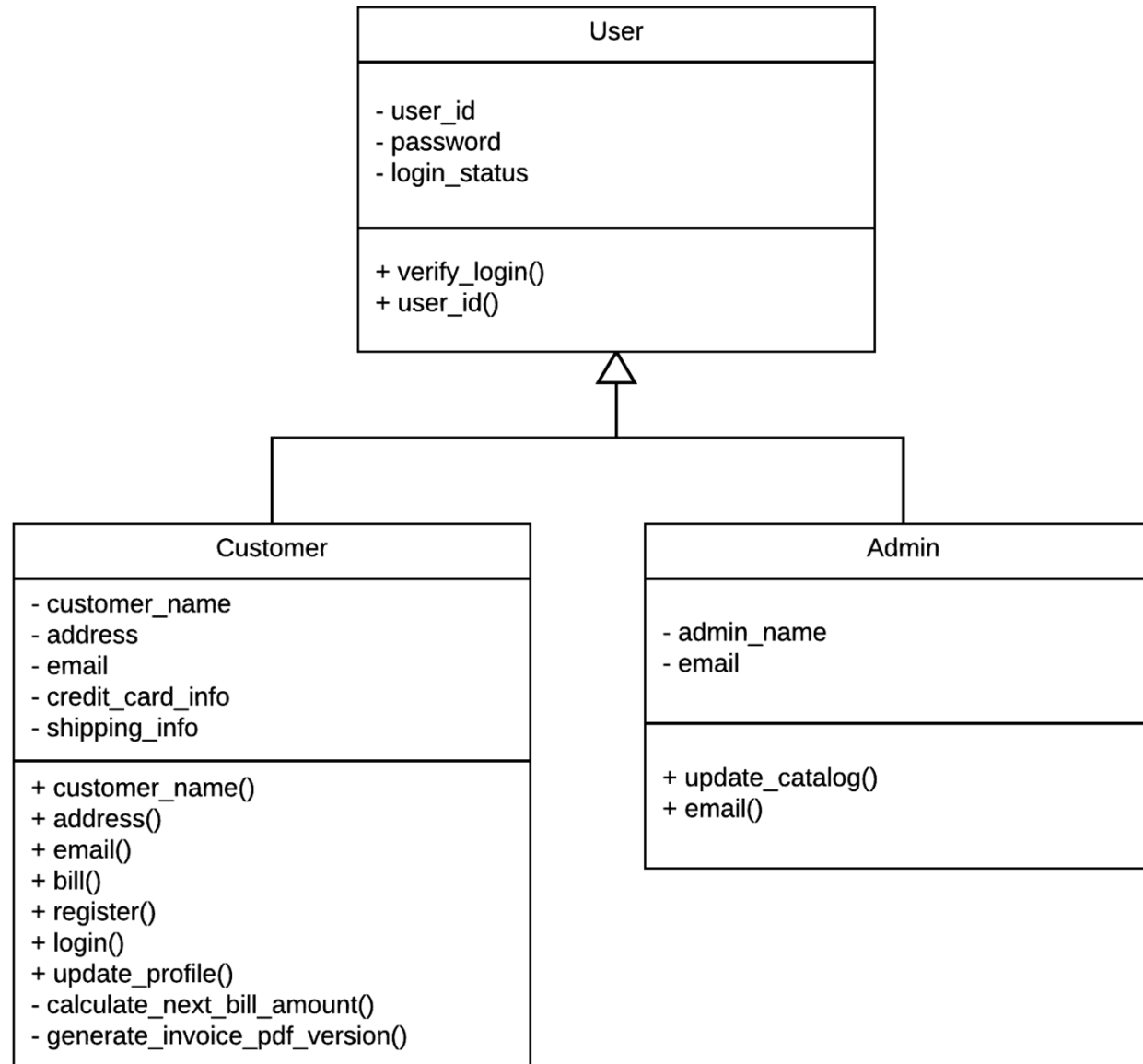


سلسله مراتبی (Hierarchical)



ترکیبی (Hybrid)





https://www.tutorialspoint.com/software_architecture_design/object_oriented_paradigm

اصول و مبانی برنامه نویسی



مجید شبیری

کارشناسی ارشد IT، گرایش شبکه
از دانشگاه صنعتی امیرکبیر

