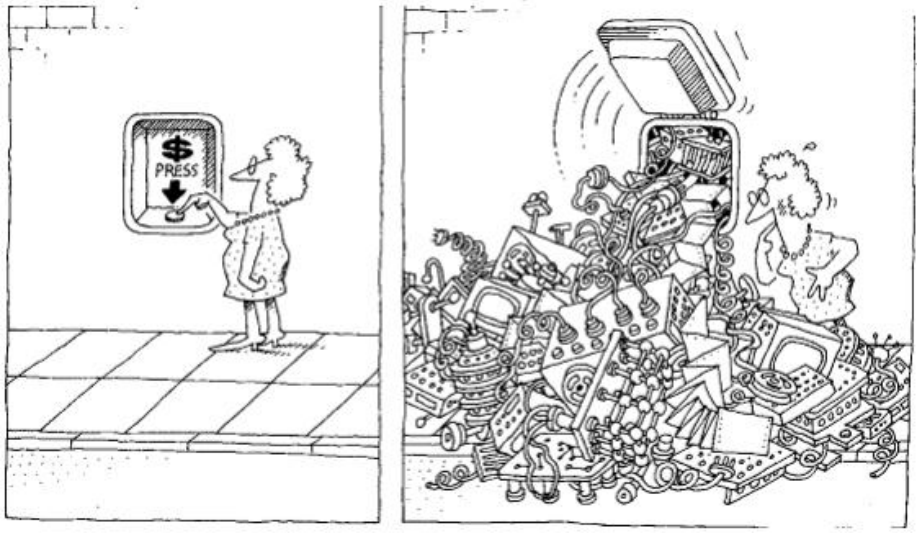
Software Engineering

مهندسی نرم افزار

جزوه عباسپور – بخش سوم

پیچیدگی سیستمها

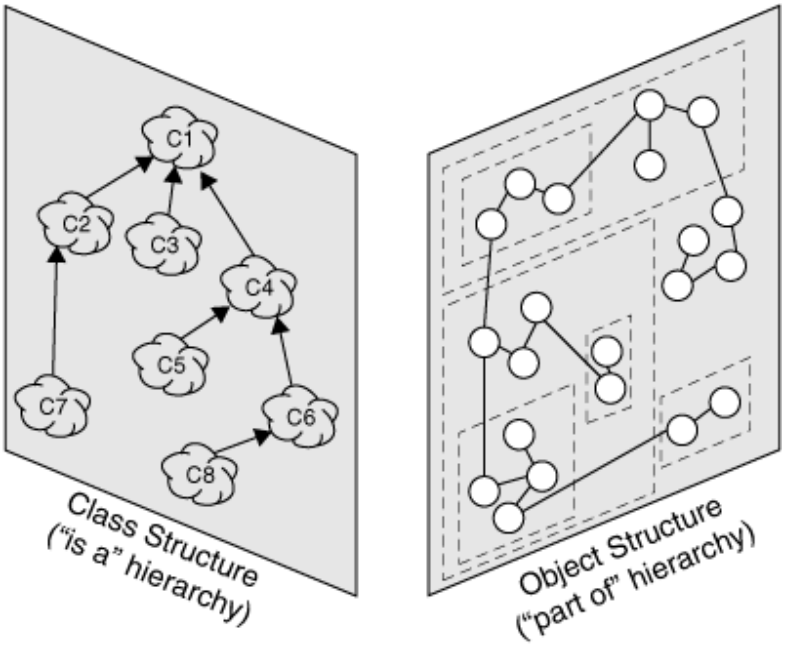
- برای مدل سازی یک سیستم باید پیچیدگی های آن را بشناسیم . منظور از پیچیدگی انواع ارتباط بین مولفه های آن سیستم می باشد . این ارتباطها مدل سازی سیستم و حتی گاهی شناخت سیستم را دشوار می نماید .



- ذهن انسان در پردازش پیچیدگی ها محدود است .
- انواع پیچیدگی ها :
  - ✓ پیچیدگی های سازمان یافته (Organized Complexity)
  - ✓ پیچیدگی های سازمان نیافته (Disorganized Complexity)
- **چطور میتوان یک سیستم پیچیده را بررسی و مدل سازی نمود ؟**
- **باید سعی کنیم پیچیدگی های سازمان نیافته را به پیچیدگی های سازمان یافته تبدیل کنیم . فهم و درک سیستم پیچیده به صورت سازمان یافته به مراتب ساده تر از سیستم پیچیده به صورت سازمان نیافته می باشد .**

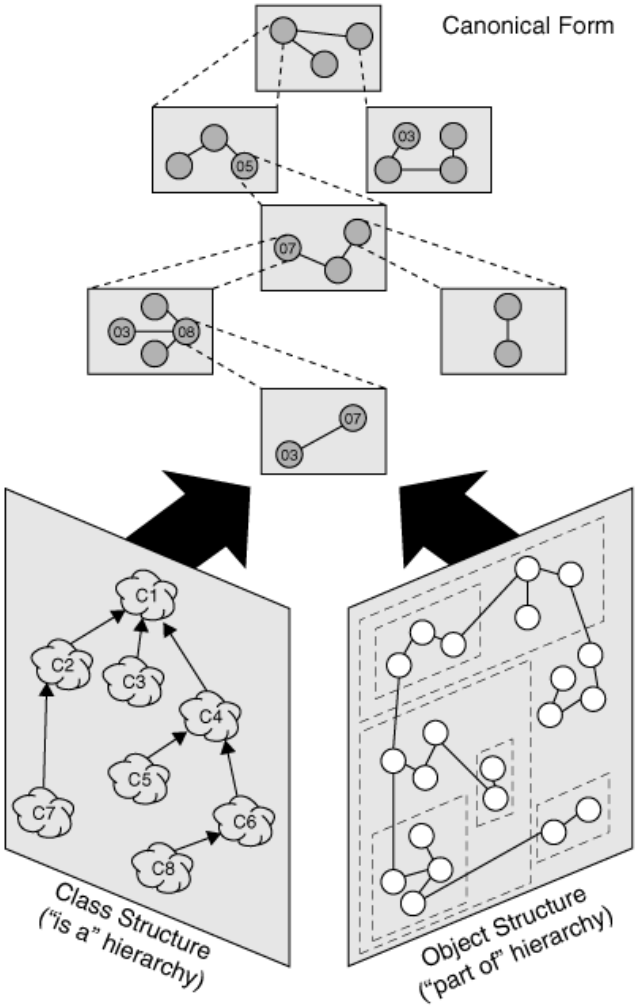
پیچیدگی سیستمها

- اغلب سیستمهای پیچیده از زیر سیستمهای کوچکتر تشکیل شده اند و یک ساختار سلسله مراتبی بین آنها برقرار است .
- برای کاهش پیچیدگی از طریق نگرش سازمان یافته به پیچیدگی باید سیستم را به اجزای کوچکتر (ساده تر) تقسیم کرد و سپس روابط بین این اجزا را شناسایی کرد .



- انواع رابطه بین اجزا در سیستمهای پیچیده :
  - ✓ ساختار کلاس (**IS-A**)
  - ✓ ساختار شیئ (**PART-OF**)
- با تجزیه سیستم در واقع برای بررسی هر قسمت از سیستم نیاز نیست کل سیستم را در نظر بگیرید و در این صورت بر پیچیدگی سیستم و محدودیت ذهن انسان غلبه میکنید .

تحلیل شیئ گرا



- اصول شیئ گرایی تا حد زیادی برای ساده کردن سیستمهای پیچیده موثر است .

- در این نگرش دنیای نرم‌افزار را به عنوان سیستم‌هایی می‌توان در نظر گرفت که شامل مجموعه‌ای از اشیاء مستقل از همدیگر بوده ولی بین آنها روابطی حاکم است که بر اساس مدلسازی این روابط و اثرات متقابل آنها، ارائه راه حلی برای یک مسأله از طریق نرم‌افزار در دنیای واقعی میسر می‌شود.

در مدل شی تعریف یک سیستم و عناصر اصلی آن بر چهار اصل زیر استوار است:

- **Abstraction** — تجرید یا چکیده سازی
- **Encapsulation** — پنهان سازی جزئیات یا محصورسازی
- **Modularity** — واحدبندی
- **Hierarchy** — سلسله مراتب

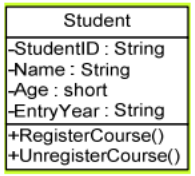
تجرید – Abstraction

- اصل تجرید به طراح و آنالیز کننده سیستم کمک میکند تا بجای در نظر گرفتن همه جزئیات یک زیر سیستم تنها به بررسی ابعاد اساسی آن از زاویه دید خود بسنده کنند .

- به عنوان مثال : برای بررسی آناتومی بدن از زاویه دید متخصص پوست استخوان بندی و ساختار رگ و قلب و ... اهمیت ندارد و یا متخصص اعصاب فقط بر روی ساختار اعصاب تمرکز دارد و به جزئیات دیگر نمی پردازد . هر یک از این مثالها به نوعی از تجرید برای یک مفهوم ثابت (بدن انسان) استفاده میکنند .

- تجرید روی اینکه یک شیئ چیست و چکار میکند تمرکز دارد و به جزئیات نحوه پیاده سازی نمی پردازد .

- **انواع تجرید :**
  - ✓ تجرید موجودیت (Entity Abstraction)
  - ✓ تجرید رفتار (Procedural Abstraction)
  - ✓ تجرید مجازی (Virtual Abstraction)



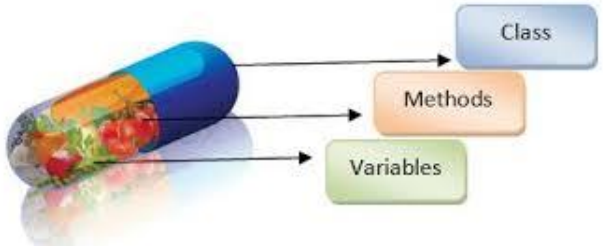
- **تعیین و تعریف یک عملیات یکسان**
- **وقتی مطرح است که لایه های مختلف هر یک برای انجام یک عمل مشخص یکسان باید وظیفه ای را به عهده گیرند و نتیجه عمل یک لایه به لایه دیگر برای اعمال عملیات مورد نظر به لایه دیگر تحویل داده میشود . مثال : TCP/IP**

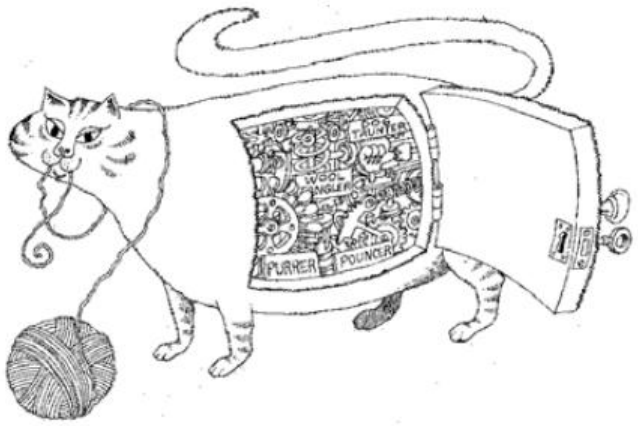
پنهان سازی - Encapsulation

- پنهان سازی یکی دیگر از روشهای ساده سازی پیچیدگی ها می باشد که با محدود کردن راههـای دسترسـی بـه یـک شـیء و محصور کردن آن جزئیات غیر ضروری را از دید استفاده کننده بیرونی پنهان می سازد .

- قواعد پنهان سازی ایجاب میکند که ارتباط اجزا از طریق یک واسط باشد و هیچ قسمتی از اجزای نرم افزاری نباید بـه جزئیـات داخلی یک قسمت دیگر وابسته باشد .

Abstraction Sample

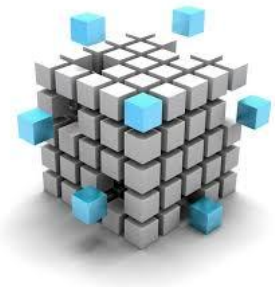




Encapsulation hides the details of the implementation of an object.

Modularity – واحد بندی

- تقسیم بندی سیستم به واحد هایی که از نظر منطقی در یک گروه قرار میگیرند (واحد بندی) میتواند در ساده سازی سیستم موثر باشد .



- ✓ قابلیت استفاده مجدد
- ✓ پیچیدگی کمتر

«سیستمی را واحدبندی شده گویند که به مجموعه‌ای از واحدهای منسجم و معنی‌دار که وابستگی بین آنها حداقل است، تجزیه شده باشد.»

Cohesion

Coupling

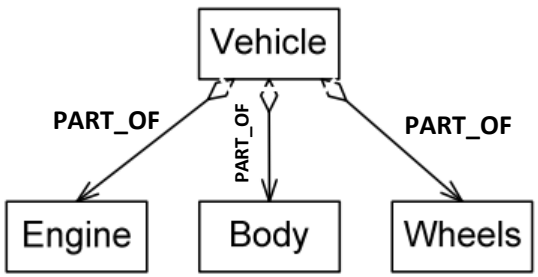
انسجام عبارتست از خاصیتی متعلق به درجه ارتباط عملکردی عناصر داخلی یک واحد نسبت به هم است. واحد A را در نظر بگیرید، اگر همه عناصر A برای رسیدن به یک هدف منسجم و واحد با هم همکاری می‌کنند پس A یک واحد کاملاً منسجم است. به هر اندازه که عناصر واحد A وظایف گوناگونی را انجام می‌دهند و به هر اندازه که ارتباط این وظایف با همدیگر ضعیف باشد، به همان اندازه درجه انسجام ضعیفتر خواهد بود.

Modularity packages abstractions into discrete units.

سلسله مراتب – Hierarchy

- کلاس نقشه نوعی و مشترک برای گروهی از اشیاء است که ویژگی‌های مشترکی داشته، و رفتارهای مشترکی از خود نشان می‌دهند . در واقع کلاس تعریف یک موجودیت است و شیئ نمونه ای از آن موجودیت .

- سلسله مراتب عبارتست از مرتب سازی تجرید ها در سطوح مختلف .رابطه سلسله مراتبی بین کلاسها IS-A و PART-OF می باشد .



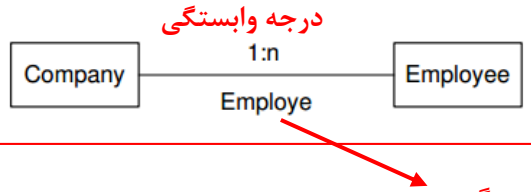
- وراثت مهمترین شکل سلسله مراتب IS-A می باشد که در آن کلاس فرزند خصوصیات و رفتار کلاس پدر را ارث می برد .

- در ارتباط PART_OF یک کلاس از یک یا چند کلاس دیگر تشکیل می شود .

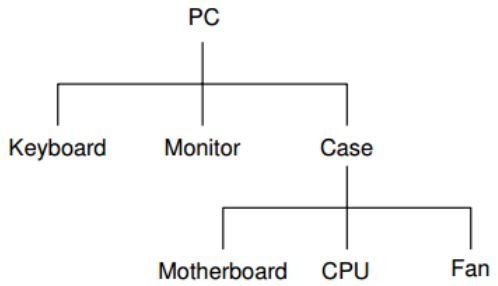
انواع ارتباط بین کلاسهای مختلف

- **رابطه انجمنی** (Association)
- **رابطه تجمعی** (Aggregation)
- **رابطه وراثت** (Inheritance)

- **رابطه انجمنی :** نوعی وابستگی معنایی بین کلاس‌های متفاوت که با حذف وابستگی عملاً هیچ ارتباط بین دو کلاس وجود نخواهد داشت. مثال:



- **رابطه تجمعی** : زمانی که یک شی از تعدادی اشیاء دیگر تشکیل می‌گردد، این رابطه را تجمعی(جزئی-از) گویند



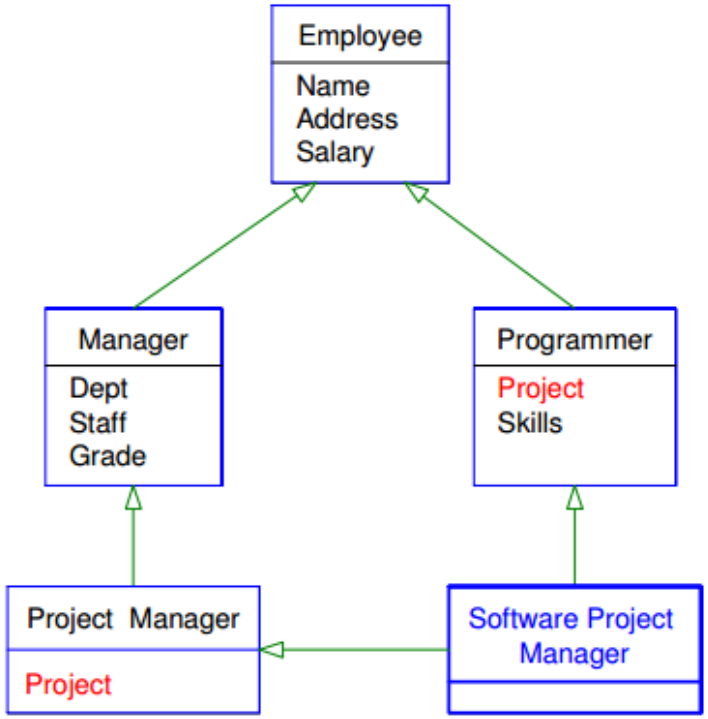
- **رابطه وراثت :** وراثت عبارت از رابطه بین چند کلاس که در آن یک کلاس در ساختار و رفتار یا هر دو با یک کلاس(وراثت یگانه) یا چند کلاس (وراثت چندگانه) دیگر شرکت دارد.

Specialization

در واقع کلاس فرزند یک تخصیص از کلاس پدر را نمایش داده و همزمان کلاس پدر یک تعمیم از کلاس فرزند به حساب می‌آید.

Generalization

مثال : رابطه وراثت (Inheritance)



سوال : اگر موجودیت پروژه (project) را بطور جداگانه در نظر بگیریم ارتباط مدیر پروژه و برنامه نویس با پروژه از چه نوعی می باشد ؟

سوال :کدامیک از روابط زیر رابطه وراثت چندگانه می باشد ؟

طبقه بندی کلاسها

به طور کلی طبقه بندی کلاسها مزایای زیادی دارد که یکی از آنها کاهش پیچیدگی سیستم می باشد . نظم یک سیستم ارتباط زیادی با طبقه بندی یا دسته بندی اجزای آن دارد .

Classification

استفاده از طبقه‌بندی در شناسایی و بهبود کلاس‌ها

منابع تشخیص کلاسها :

- فضای مسئله
- فضای راه حل

کلاسهایی که برای حل مسئله بهتر است ایجاد کنیم

کلاسهایی که باید ایجاد کنیم

شناسایی کلاسها

شناسایی کلاسها بر اساس دو روش انجام میگیرد :

- **مبتنی بر داده**

فرآیند تعیین کلاس‌ها با پرسیدن دو سوال صورت می گیرد:

۱- ساختار هر کلاس چیست؟

۲- چه عملیاتی بوسیله هر کلاس انجام می گیرد؟

- **مبتنی بر وظیفه**

فرآیند تعیین کلاس‌ها با پرسیدن دو سوال صورت می گیرد:

۱- هر کلاس چه مسئولیتی دارد؟ (چه عملیاتی بوسیله این کلاس انجام می گیرد؟)

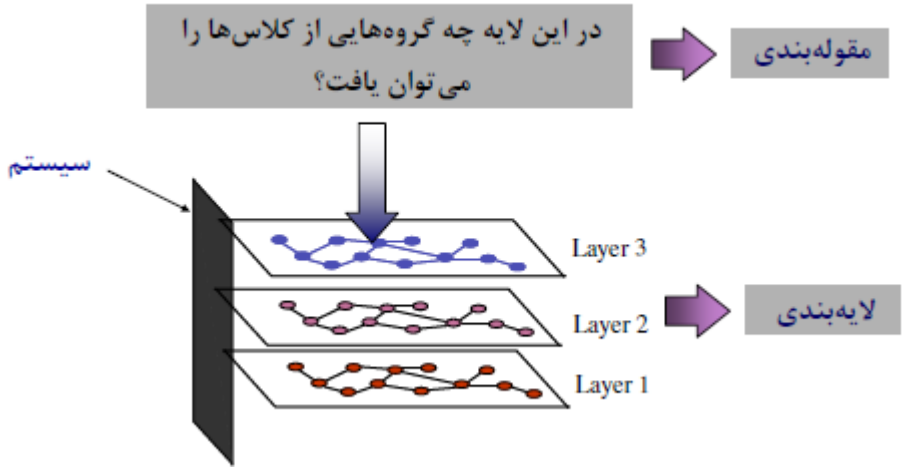
۲- این شی، چه اطلاعاتی با بقیه اشیاء به اشتراک می گذارد؟

در روش مبتنی بر داده نگاه و اولویت شما برای دسته بندی یا طبقه بندی کلاسها ساختار داده ای و تشابه ساختار داده ای می باشد . اما در روش مبتنی بر وظیفه نگاه و اولویت شما عملکرد کلاس می باشد .

لایه بندی

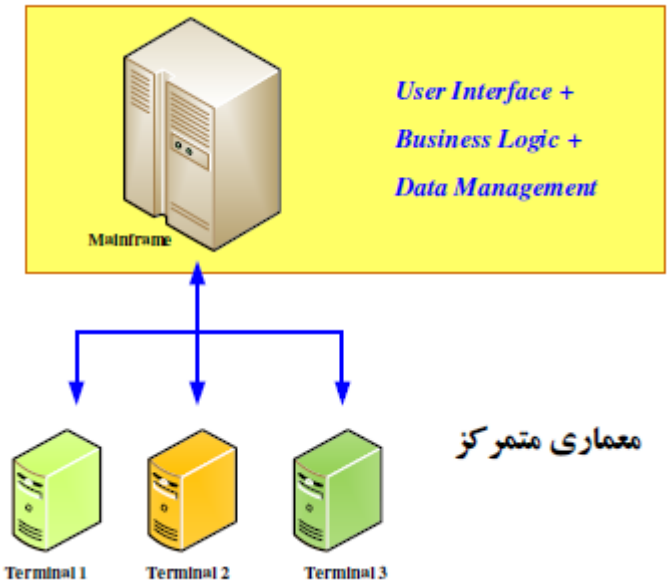
در لایه‌بندی یک سیستم نرم‌افزاری به صورت تعدادی از لایه‌ها تقسیم‌بندی می‌گردد. هر لایه از تعدادی مولفه تشکیل شده که همکاری گروهی این مولفه‌ها بوجود آورنده رفتار لایه می‌باشد. استفاده از لایه‌بندی وابستگی‌ها را کاهش می‌دهد بطوریکه لایه‌های پایین تر از جزئیات و واسط‌های لایه‌های بالاتر اطلاعی ندارند.

همچنین این روش می‌تواند به شناسائی بخش‌های قابل استفاده مجدد و تصمیم گیری در مورد مولفه‌های قابل خریداری و مولفه‌های قابل ساخت کمک نماید. می‌توان به معماری‌های متمرکز، معماری Client/Server و معماری 3-Tier بعنوان نمونه‌های از لایه‌بندی در دنیای نرم‌افزار اشاره نمود.

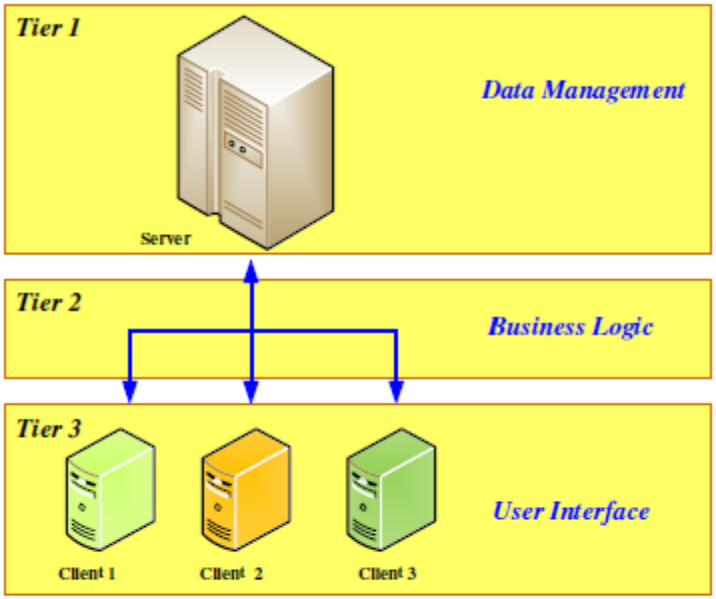


مقوله بندی در واقع گروه بندی کلاسها می باشد .

لایه بندی



معماری متمرکز



معماری 3-Tier

نمونه ای از لایه بندی در معماری ۳ لایه