

طراحی سیستم های شی گرا



M.Madadyar

<http://www.madadyar.ir>

info@madadyar.ir

سیستم چیست ؟

- مجموعه ای از اجزای هماهنگ که با هم هدف خاصی داشته باشند .
- در مهندسی نرم افزار با سیستمهای اطلاعاتی روبرو هستیم .
- سیستمهای اطلاعاتی دارای پنج مولفه هستند.
 - مجموعه ای از ورودیها
 - مجموعه ای از خروجیها
 - مجموعه ای از فایلهای اطلاعاتی که اطلاعات سیستم در آنجا ذخیره می شود.
 - مجموعه ای از پروسسها که بیانگر عملیات سیستم می باشد.
 - هدف

نرم افزار

- فرایند تولید نرم افزار چگونگی تحلیل ، طراحی و پیاده سازی سیستمهای اطلاعاتی را توصیف می کند.

Methodologies –

- فرایند دقیق و چند مرحله ایی جهت تولید نرم افزار

Techniques –

- مجموعه ایی از فنون مدلسازی که امکان ایجاد محصولات هر مرحله از فرایند را فراهم می کند

Tools –

- نرم افزارها و برنامه های کامپیوتری که فرایند مدلسازی و بهره گیری از تکنیکها را امکان پذیر می کند

نرم افزار

- مهندسی به معنی

– انجام عملیاتی به منظور تولید محصولی با تبعیت از مجموعه قواعد مشخصی. به عبارت دیگر انجام سیستماتیک یک عملیات.

مهندسی نرم افزار

– طراحی و تولید نرم افزار با تبعیت از یک قواعد و ضوابط مشخصی.

مراحل تولید نرم افزار

مطالعه امکان سنجی (Feasibility Study)

تحلیل سیستم (System Analysis)

طراحی سیستم (System Design)

پیاده سازی سیستم (System Implementation)

آزمون و اصلاح (System Test & Integration)

اعتبارسنجی سیستم (System Validation)

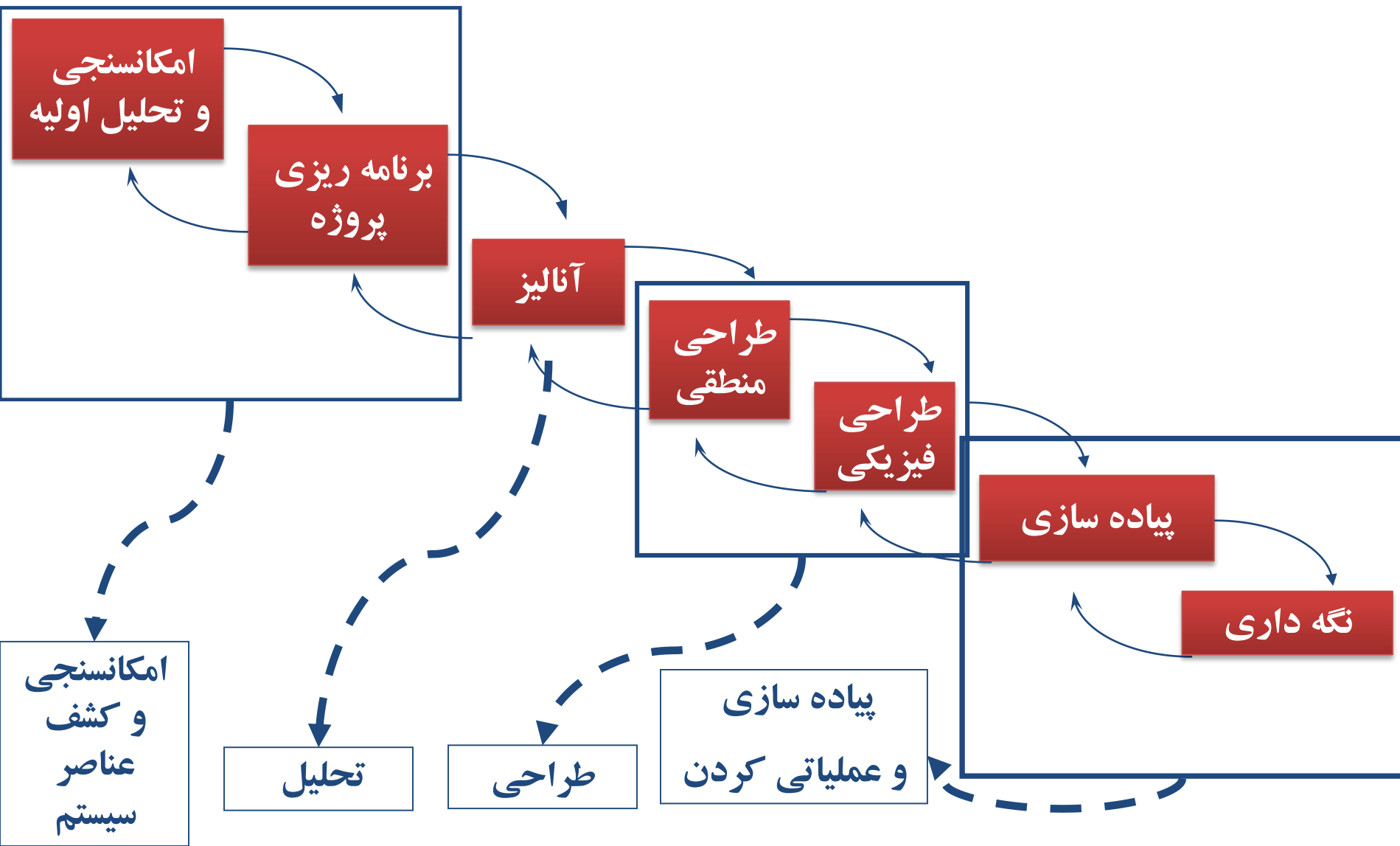
نگه داری سیستم (System maintenance)

نگه داری سیستم

فرایند تولید نرم افزار

- تولید نرم افزار فرایندی مرحله به مرحله و گام به گام است.
- مجموعهٔ مراحل و گامهای مورد نیاز ، محصولات هر مرحله ، قواعد و دستوراتی که باید در ایجاد محصولات میانی رعایت گردد و چگونگی بدست آوردن هر یک از آنها را اصطلاحاً چرخهٔ حیات نرم افزار (SDLC) می نامند.
- Software Development Life Cycle = **SDLC**

مراحل تولید نرم افزار در قالب SDLC



رویکرد ها



شی گدا

ساخت
یافته



ایده

- اگر مسئله بزرگی داشته باشیم و بخواهیم راه حلی برای آن پیدا کنیم . این مسئله را به زیرمسائل کوچکتر بشکنیم تا به زیرمسائل حل شده برسیم بعد این زیرمسائل حل شده را بهم بچسبانیم تا جواب نهایی را پیدا کنیم.

کاربرد

- استفاده از این تکنولوژی در زبانهای ساخت یافته . مانند : پاسکال و...



ایده

- همه چیز را در قالب اشیا و روابط بین آنها دیدن.

کاربرد

- زبانهای برنامه نویسی گرا و پایگاه داده شی گرا و مهندسی نرم افزار شی گرا.

شیء گرایی بیش از آنکه یک متدلوژی مشخص برای توسعه سیستم ها باشد، یک رهیافت و یا فراتر از آن یک **قالب فکری** برای مدلسازی و ساخت سیستم هاست.

مقایسه

❖ روش ساخت یافته ریشه در **مهندسی سیستم** دارد.

❖ روش شیء گرا ریشه در **مهندسی نرم افزار** دارد.

❖ هر دو روش اگر بدرستی بکار گرفته شوند می توانند مفید واقع شوند.

❖ تفاوت در تمرکز این دو روش است،

❖ **تحلیل ساخت یافته روی فرایندها و داده ها متمرکز میشود،**

❖ **تحلیل شیء گرا به موجودیتهای و ارتباطات بین آنها می پردازد.**

❖ کدام روش بهتر است؟

❖ بستگی به نوع مسئله، مهارت های تیم، و ابزارهای در دسترس دارد.

❖ **برای مسائل مهندسی نرم افزار روش شیء گرا مناسب است.**



طراحی سیستم های شی گرا



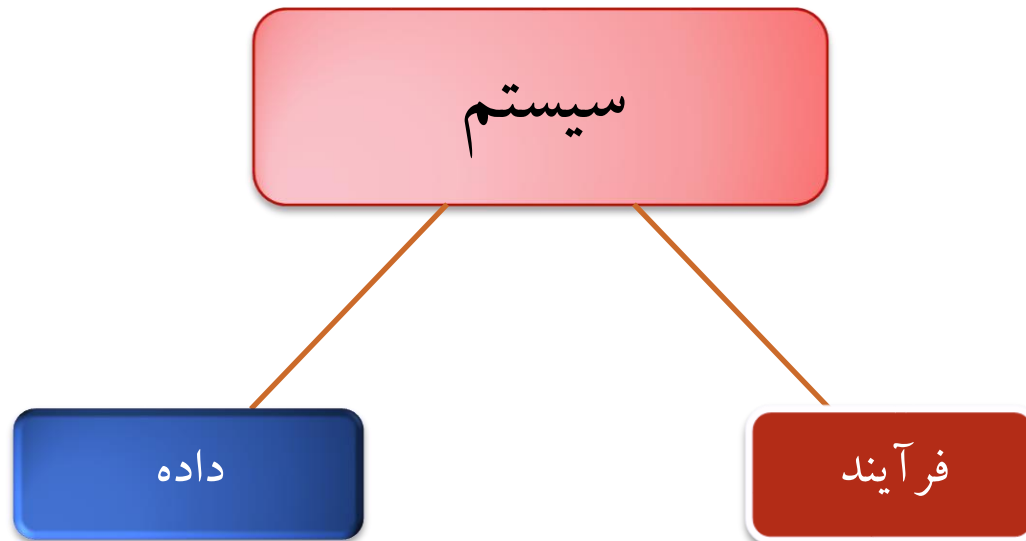
M.Madadyar

<http://www.students.madadyar.com>

info@madadyar.com

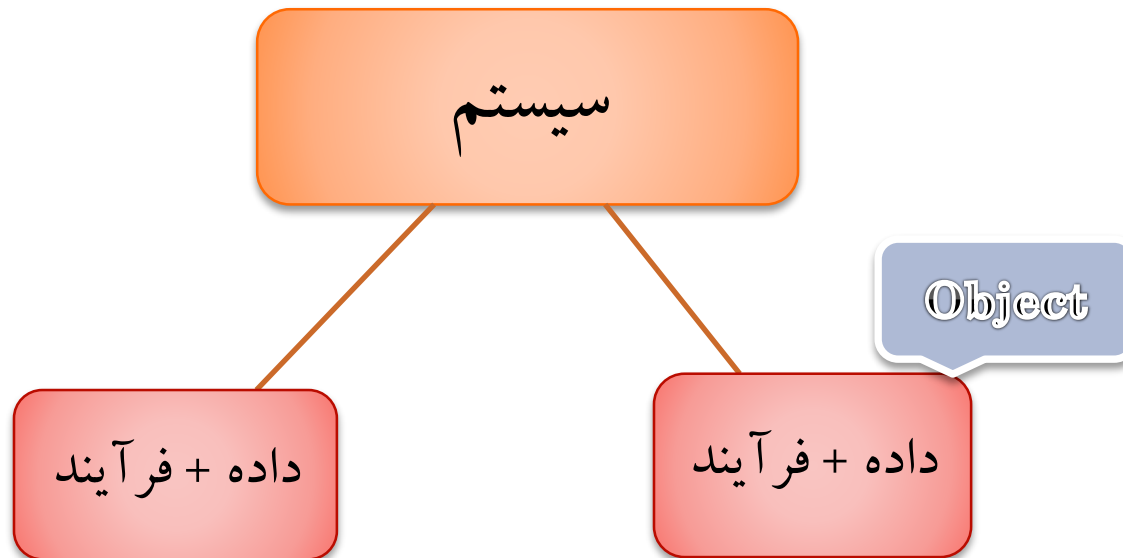
ساخت یافته

- در الگوی ساخت یافته، در اولین تقسیم، سیستم به دو وجه "داده" و "فرآیند" تفکیک می شود.
- سپس طی روندی سلسله مراتبی و مطابق با روش از بالا به پایین، هریک از این وجوه خود به اجزاء و مؤلفه های فرعی تجزیه می شوند.
- این روند تا به جایی ادامه پیدا می کند که عناصر قابل پیاده سازی (ساختار پرونده های فیزیکی، واحدهای برنامه ای) مشخص شوند.



شی گرا

- در رهیافت شیء گرا تفکیک سیستم به عناصر داده و فرآیند صورت نمی گیرد.
- بلکه به واحدهایی تفکیک می شوند که هریک از آنها خود شامل عناصر داده و فرآیند هستند.
- این واحدها را بسته به سطح تجزیه آنها، می توان **شیء** یا **کلاس** نامید.



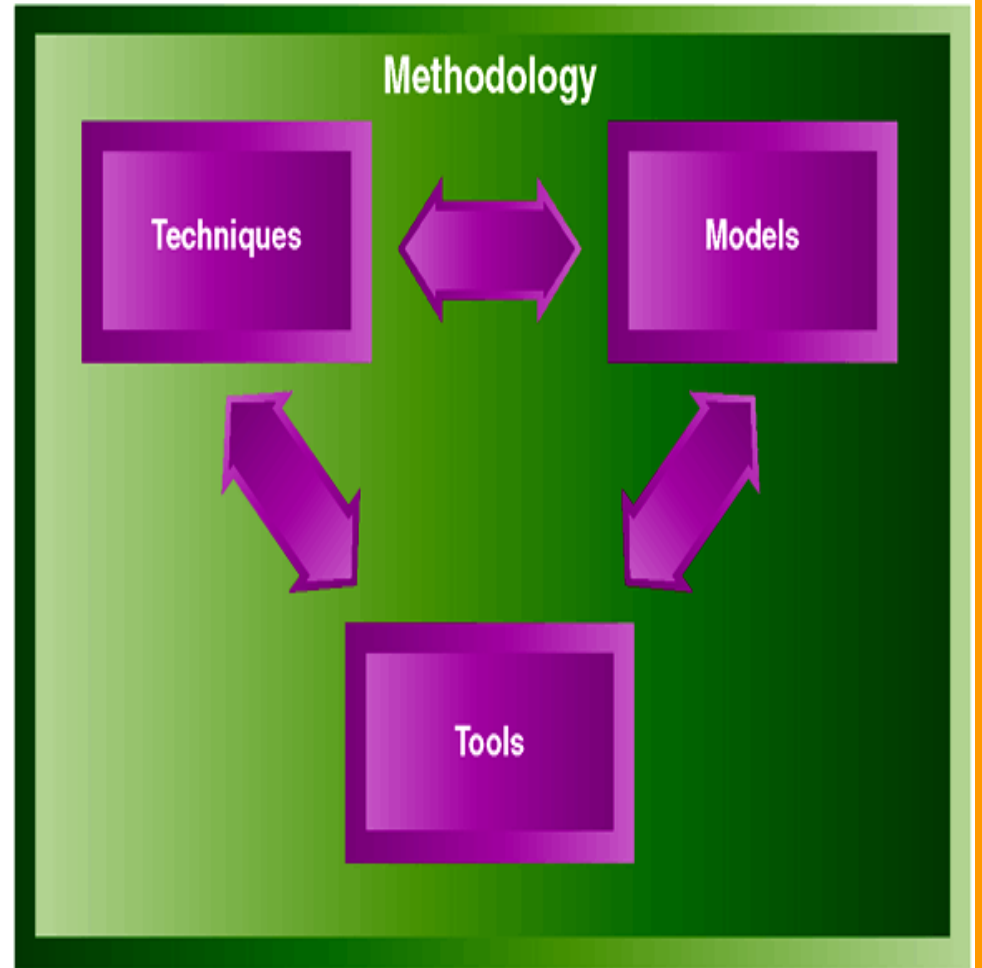
متدولوژی (METHODOLOGY)

- فرا روش یا متدولوژی به بیان مراحل انجام یک کار با هدف مشخص پرداخته و محصولات و فرآورده های ایجاد شونده در هر مرحله را بیان می نماید.
- یک متدولوژی ، مرکب است از مرحله ای که هر یک به نوبه خود از مراحل فرعی تشکیل شده اند.
- با کمک این مراحل ، توسعه دهندگان سیستم می توانند در هر مرحله ابزارها و روش های مناسب آن مرحله را انتخاب کرده و پروژه های توسعه سیستم ها ی اطلاعاتی را برنامه ریزی ، مدیریت ، کنترل و ارزیابی می کنند.
- فرایند دقیق و چند مرحله ایی جهت تولید نرم افزار.



متدلوژی ها و مدل ها

ارتباط میان اجزای
متدلوژی



متدولوژیهای ساخت یافته

Structured
Methodologies

متدلوژی ساخت یافته

نگرش فرایند گرا

○ شاخص های این دسته از متدلوژی ها ، عبارتست از :

- اطلاعات و فعالیت ها سیستم مورد نظر، بصورت مجزا، مورد بررسی قرار می گیرد .
- فرآوردهای هر مرحله بصورت مستقل از کاربران ، تهیه می شود و کاربر، در حین کار، دخالت داده نمی گیرد .
- در ایجاد فرآورده های مبتنی بر متدولوژی های ساخت یافته ، از نوشتار ، بیشتر از نمایش بوسیله اشکال و دیاگرام ها استفاده می شود.
- برنامه نویسی بالا به پایین (که برنامه های پیچیده را به سلسله ماژول ها تقسیم می کرد).



○ متدولوژی های ساختیافته معروف، عبارتست از:

- SSADM (System Structured Analysis and Design Methodology)
- IE (Information Engendering)
- CDM (Custom Development Method)



متدولوژی های شیء گرا

Object Oriented
Methodologies

○ شاخص های این دسته از متدولوژی ها عبارتند از:

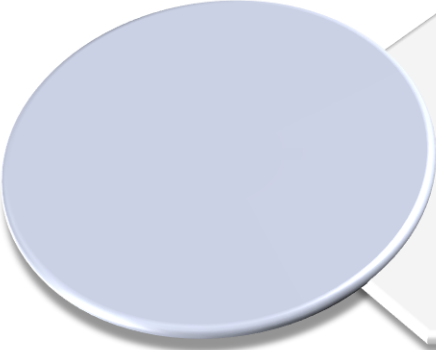
- اطلاعات و فعالیتهای مرتبط، در قالب یک ساختار، که با عنوان کلاس، بیان می شود.
- اشکال در تمام متدولوژی های این دسته، به صورت عام و در قالب UML، استفاده می شود.
- روش انجام کار به صورت تکرار، با هدف توسعه تدریجی تهیه فرآورده ها در مراحل مختلف در نظر گرفته می شود.
- مراحل مختلف تا حصول به پذیرش توسط کاربر می باشد.
- بعلت پشتیبانی معمارهای چند لایه توسط این دسته از متدولوژی ها، در توسعه نرم افزارهای چند لایه، این دسته از متدولوژی ها پیشنهاد می گردد.



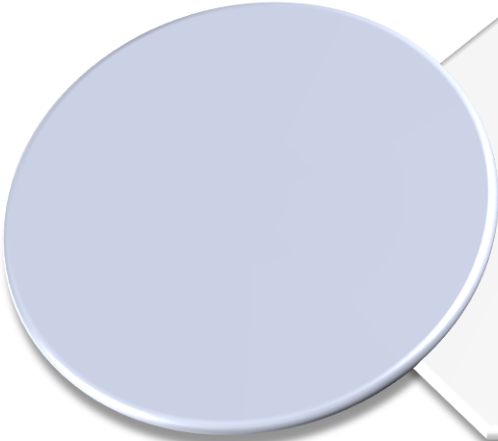
○ متدولوژی های شی گرای معروف عبارتست از :

- RUP (Rational Unified process)
- CBD (Component Based Development)
- OOADA (Object Oriented Analysis & Design Methodology)
- OOSE (Object Oriented Software Engineering)
- OMT (Object Modeling Technique)





آنالیز ساخت یافته به
طراحی ساخت یافته و
متعاقبا برنامه نویسی
ساخت یافته



آنالیز شی گرا منجر به
طراحی شی گرا و در
نهایت برنامه نویسی
شی گرا می شود.



مروری بر یک متدولوژی ساخت یافته

SSADM ○

- نمودار LDS
○ برای نمایش ساختار منطقی داده ها
- نمودار DFD
○ نمایش جریان اطلاعات در سیستم.
- نمودار ELH (به روش جکسون)
○ نمایش تغییرات موجودیت ها در سیستم.

مدلسازی در SSADM

○ مدل سازی فرایندی ، مدل سازی داده ای و مدل سازی رفتاری از جمله تکنیک های مورد استفاده در هر دو نگرش هستند که نوع عملکرد این تکنیک ها و پیاده سازی آنها در هر یک از این دو متفاوت است .

○ تکنیک های ساخت یافته عبارتند از:

○ مدل سازی فرایندی (DFD) ،

○ مدل سازی داده ای (LDS) و

○ مدل سازی رفتاری (ELH) .



- یک DFD حرکت داده ها را در سرتاسر یک سیستم اطلاعاتی نشان می دهد ولی منطق برنامه یا مراحل پردازش را نشان نمی دهد .
- مجموعه ای از DFD ها یک مدل منطقی را بوجود می آورند که نشان می دهد سیستم چه کاری انجام می دهد (نه چگونه آن را انجام می دهد!).

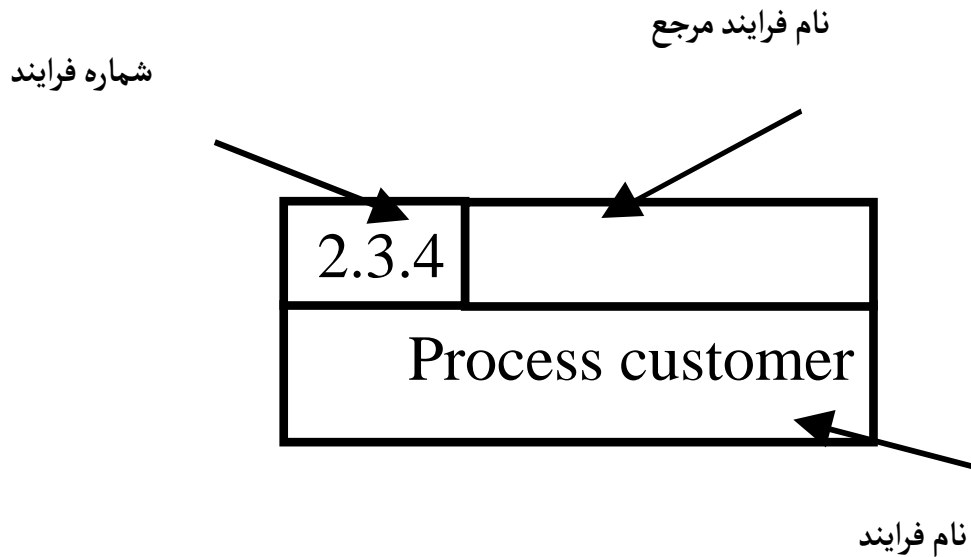
○ اجزای DFD

- DFD ها از چهار علامت پایه ای استفاده می کنند
- برای نمایش فرایندها ،
- جریان داده ها ،
- گروه های داده ای
- منبع / گیرنده خارجی



فرایند Process

- داده ورودی را دریافت می کند و خروجی را تولید می کند که در شکل یا محتویات ویا هر دو فرق می کند .



○ جریان داده ها Data flow

- یک یا چند آیتم داده ای را نشان می دهد که به شکل زیر است :



- پیکان یک جهته در مواردی است که داده ای را فقط وارد یا خارج کنیم.

- و پیکان دو طرفه برای مواردی است که هم داده را می خوانیم و هم مجدداً بازنویسی می کنیم .

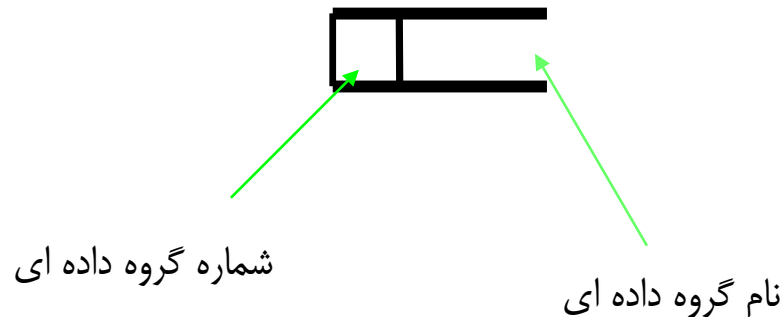


○ گروه های داده ای Data store

○ داده هایی را نشان می دهد که توسط سیستم ذخیره شده است .

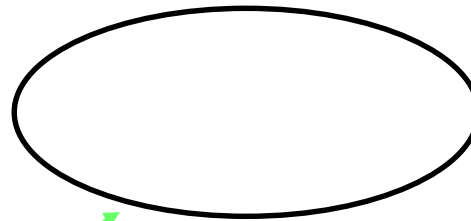
• مشخصات فیزیکی یک گروه داده ای مهم نیست زیرا ما با یک مدل منطقی سرو کار داریم.

یک گروه داده ای به شکل زیر نمایش داده می شود :



○ منبع / گیرنده خارجی

- با یک بیضی نمایش داده می شود که نام موجودیت در داخل آن می آید و از آن به عنوان یک منبع و یا گیرنده خارجی استفاده می شود .



نام



○ قواعد DFD ها

- داده ها به طور مستقیم بین فرایندها جریان پیدا نمی کنند .
- داده ها به طور مستقیم بین گروه های داده ای جریان پیدا نمی کنند.
- داده ها نمی توانند به طور مستقیم از Data store به موجودیت ها یا بر عکس انتقال یابند .



LDS(LOGICAL DATA STRUCTURE)

Entity موجودیت

به تمام چیزهای قابل تشخیص در محیط کاری کاربر گفته می شود. در پایگاه های داده ها هر جدول یک Entity است.

نمایش موجودیت



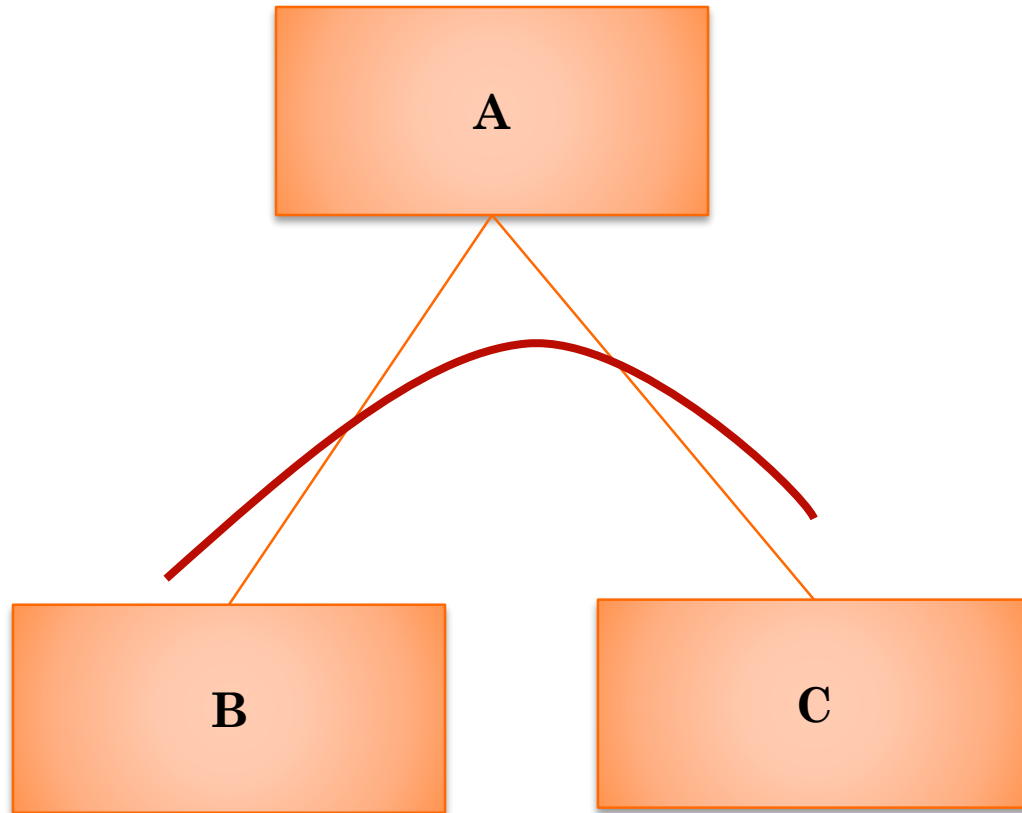
○ ارتباط (Relationship)

- راهی است که نشان می دهد موجودیت ها چگونه با هم تعامل دارند . در پایگاه داده ها ارتباط اشاره دارد به اتصالات بین جداول از طریق کلید خارجی .

○ درجه ارتباط (Cardinality)

- تعداد ارتباط های یک موجودیت با دیگر موجودیت ها است .

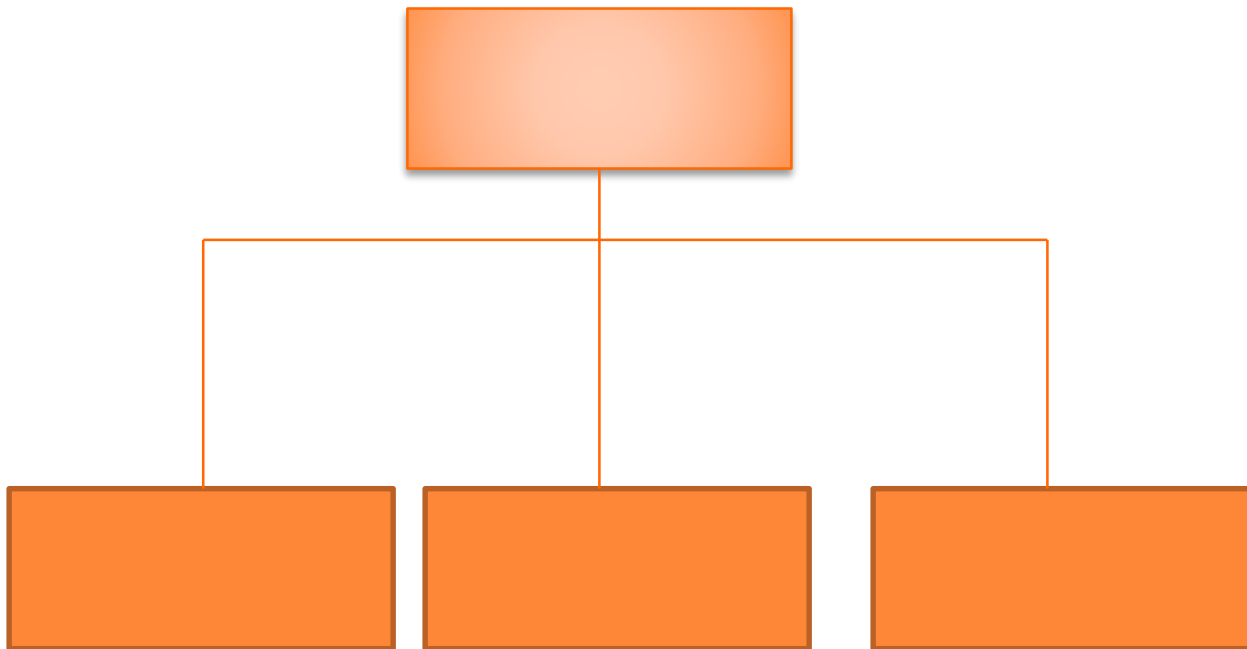




مدل سازی رفتاری ELH

○ سومین تکنیک مدل سازی ، ELH ها می باشد.

○ در این تکنیک تغییراتی را که یک موجودیت در کل سیستم پذیرفته است را توصیف می کند.



مفاهیم شی گرای



Object Oriented Concepts

تاریخچه شی گرای

اولین بار در سال ۱۹۶۶ در
قالب زبانی به نام simula
که به منظور پیاده سازی
شبیه سازی کامپیوتری تهیه
شده بود، ارائه شد.

در دهه ۷۰ زبان small
talk بوجود آمد

• (توسط آلن کی Alan Kay
طراحی شده).

از دهه ۹۰ تا کنون بشدت
در حوزه زبانهای برنامه
نویسی و مهندسی نرم افزار
رشد کرده است.



مزایای شی گرای

○ مدل سازی دنیای واقعی را تسهیل می کند.

○ شی گرایی امکان Reuse کردن برنامه های کامپیوتری را فراهم می کند.

○ Reusability

• یعنی یک قطعه برنامه ای را یکبار نوشتن و به دفعات متعدد در برنامه های مختلف از آن استفاده کردن.

• مزایا

○ الف - صرفه جویی در زمان

○ ب - صرفه جویی در هزینه



مزایای شی گرای (ادامه)

- شی گرای امکانات مناسبی برای ساخت سریع GUI فراهم می کند.
- امکان اجرای موازی (Parallel) نرم افزار را روی چندین CPU فراهم می کند.
- امکان Proto typing سریع فراهم می کند.
- بهبود ساختار برنامه. (امکان تغییر و توسعه سریع)
- امکان توسعه کم هزینه تر و سریع تر و راحت تر نرم افزار را فراهم می کند.



شی گرای (OBJECT ORIENTED)

راه سازمان دهی نرم افزار به عنوان مجموعه ای از
اشیا مجزا که در آن ساختمان داده ها و رفتارها به
هم می پیوندند.

- مثلاً در کلاس شخص، من و شما دارای صفات سن - قد و وزن هستیم.
همچنین عملیات زیر را انجام می دهیم : خوردن - خوابیدن - مطالعه کردن و
....

○ آنالیز شی گرا (OOA)

- برای تجزیه و تحلیل فرایند ساخت نرم افزار.
- محور آنالیز سیستم
 - هر چیز در فرایند پیاده سازی نرم افزار به منزله کلاس در نظر گرفته خواهد شد.
 - تاکید و سرو کار داشتن با سوالاتی که با **What** شروع می شوند.

○ طراحی شی گرا (OOD)

- تاکید و سرو کار داشتن با سوالاتی که با **How** شروع می شوند.
- در طراحی شی گرا تاکید بر پیاده سازی صفات و متدهای کلاسها است که به منزله هسته یک کلاس مطرح می گردند.





Goals of Object-Oriented Design

Robustness

- Complex programs should operation correctly
- Should deal with improper inputs and conditions (Therac-25)

Adaptability

- Software grows over a long lifetime
- May run on different generations and makes of hardware

Reusability

- Building from reusable pieces avoids “reinventing the wheel” if done right (Therac 25)

صفات بنیادی شی گرا

Design patterns

- کلاس Class
 - صفات کلاس Attributes
 - عملیات کلاس Operations
- تجرید Abstraction
- ارث بری Inheritance
- نمونه سازی Instantiation
- چندشکلی Polymorphism
- مسئولیت (Responsibility)
- پنهان سازی Encapsulation
- ارسال پیام Message Sending
- تناظر یا ارتباط Associations
 - تجمع Aggregation
 - ترکیب composition



مفاهیم شی گرا

کلاس (Class)

- مفهومی انتزاعی برای دسته بندی اشیا.
- یک کلاس یک الگو (template) است که صفات و رفتارهای یک شی واقعی را تعریف می کند.

شی (Object)

- به هر مفهوم یا هرچیز قابل درکی که بشود با خصوصیات و رفتار مستقل آن را از یک محیط باز شناسی کرد.
- نمونه ایجاد شده از یک کلاس.

صفت (Attribute)

- هر شی یکسری خصوصیات دارد که به آنها صفت گفته می شود که در واقع یک مقدار یا ارزش مشخصی برای آن به ازای هر شی می تواند وجود داشته باشد.

عملیات (Operation)

- توابع و رویه هایی که بر روی داده های کلاس عملیاتی انجام می دهند.
- نشان دهنده رفتار کلاس است.



نمونه سازی (INSTANTIATION)

- به معنای ایجاد یک شی از یک کلاس است یعنی نمونه ای از آن کلاس که همان شی می باشد تولید می گردد.
- سازنده کلاس (Constructor)
- مخرب کلاس (Destructor)

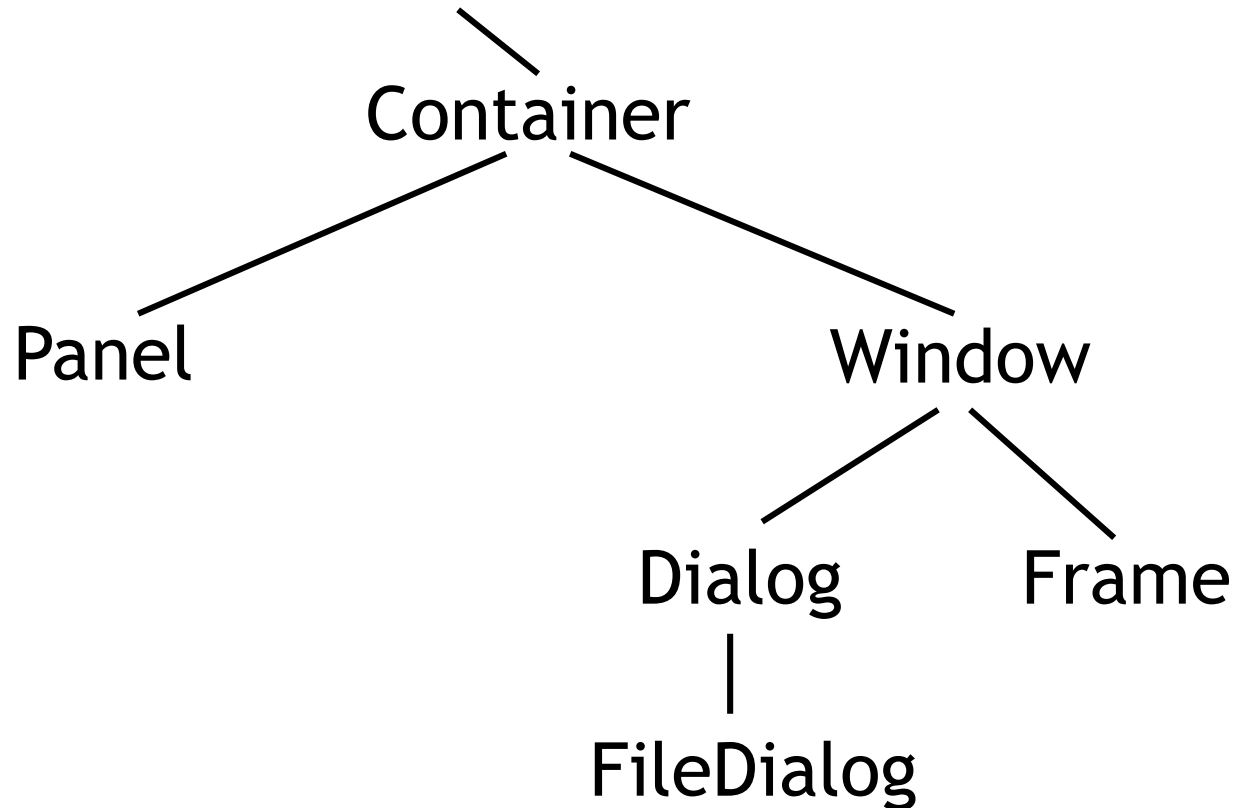


ارث بری (INHERITANCE)

- ارث بری یکی از جنبه های مهم در مفاهیم شی گرای است.
- در صورت وجود خصوصیات و عملیات مشترک مابین کلاس ها ، می تواند ارث بری صورت گیرد.
- هر کلاس می تواند از کلاس یا کلاسهای دیگری مشتق شده باشد.
 - (خواص یا رویدادهای کلاسهای دیگر را به ارث ببرد).
- Subclass, Superclass



EXAMPLE OF (PART OF) A HIERARCHY



A **FileDialog** is a **Dialog** is a **Window** is a **Container**



EXAMPLE OF INHERITANCE

```
class Person {  
    String name;  
    int age;  
    void birthday () {  
        age = age + 1;  
    }  
}
```

```
class Employee  
    extends Person  
{  
    double salary;  
    void pay () { ...}  
}
```

Every **Employee** has **name** and **age** fields and **birthday** method *as well as* a **salary** field and a **pay** method.



تجريد (ABSTRACTION)

- صفات و عملیات یک شی را آنقدر از صافی بگذرانیم تا مجموعه ای که مورد نیاز است باقی بماند .
- این یک دید کلی بدون نگاه کردن به جزئیات می باشد.
- به این معنا است که بخشی از یک موجودیت یک مفهوم را که میان همه وارثان مشترک است، از پیش تعریف کنیم.

- **Abstract Class**

- **No Objects!**

- **(Interface)**

- **Sealed Class**

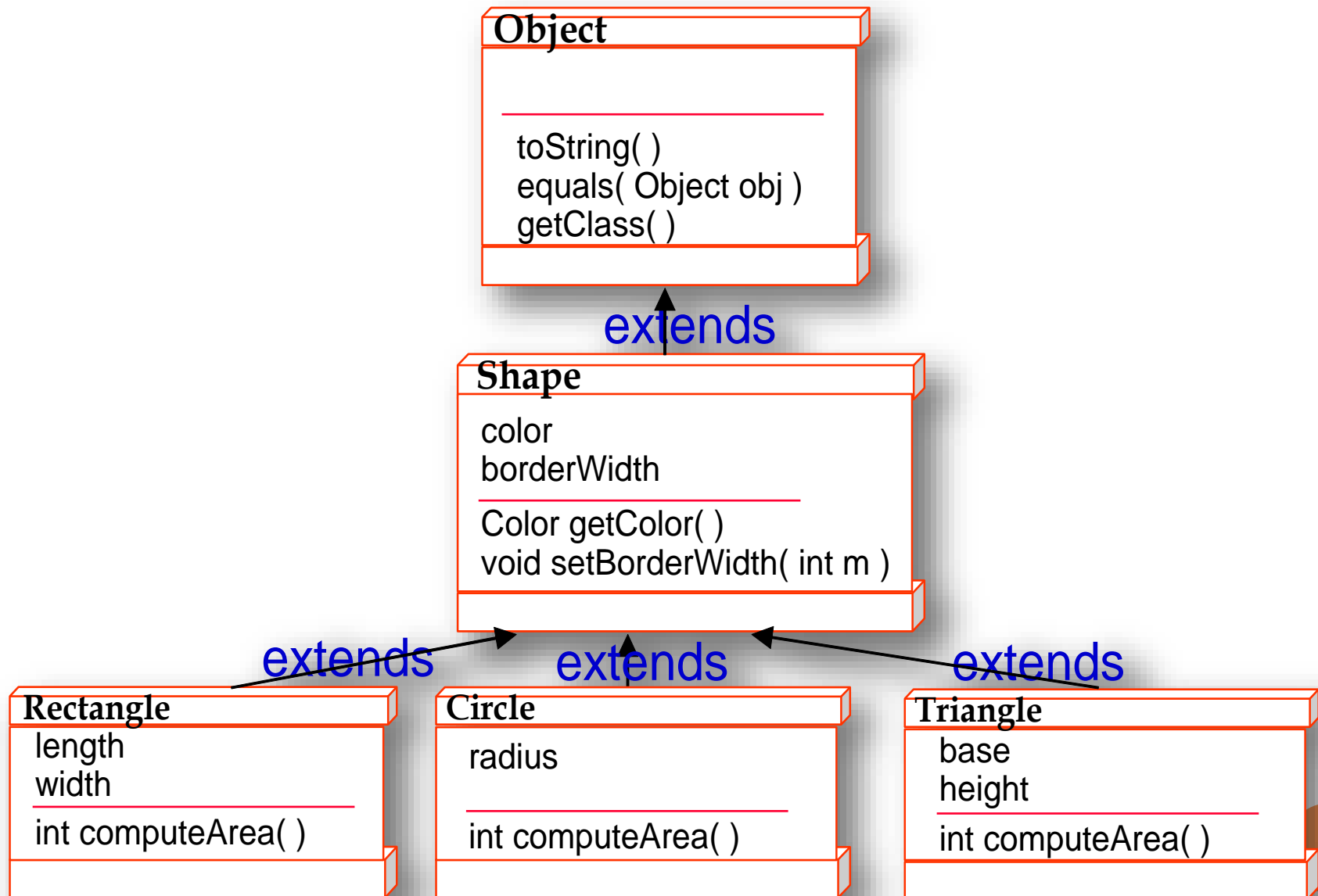


چند ریختی (POLYMORPHISM)

- بطور کلی می توان گفت که چند شکلی به معنای یک چیز بودن و چند شکل داشتن است.
- چند ریختی به این معنا است که اشیاء می توانند در موقعیت های مختلف، رفتارهای متفاوتی بروز دهند.
- عملیات ها می توانند نام مشابهی در کلاس های متفاوت داشته باشند و هر کلاس عملیات را به شکلی مختلف انجام دهد.
- به عنوان مثال در کلاس چند ضلعی تابع محاسبه مساحت ،
تخفیف متفاوت برای انواع کالا و



POLYMORPHISM



پنهان سازی (ENCAPSULATION)

- اشیا برخی عملیات و خصوصیات داخلی خود را از دید خارج از دنیای خود پنهان می سازند.
- سطوح دسترسی به اعضای یک کلاس (Access Modifier)
- **public**, **private**, **protected**.
- عمل مخفی سازی باعث می شود که اشیاء بدون آنکه از چگونگی کارکرد یکدیگر اطلاع داشته باشند با هم کار کنند.
- ◉ مثال: رانندگی بدون اطلاع از تنظیمات و کارکرد درونی خودرو.



○ مسئولیت (Responsibility)

- مسئولیت چیزی است که به شی اختصاص داده می شود و وظیفه ایفای نقشی در سیستم را دارد.
- فراهم کردن سرویس های مورد انتظار.
- *An object must contain the data (attributes) and code (methods) necessary to perform any and all services that are required by the object.*



```
class Employee {  
    // Fields  
    private String name;    //Can get but not change  
    private double salary; // Cannot get.  
    // Constructor  
    Employee(String n, double s) {  
        name = n; salary = s;  
    }  
    // Methods  
    void pay () {  
        console.write("Pay to the order of " +  
                        name + " $" + salary);  
    }  
    public String getName() { return name; } // getter  
}
```

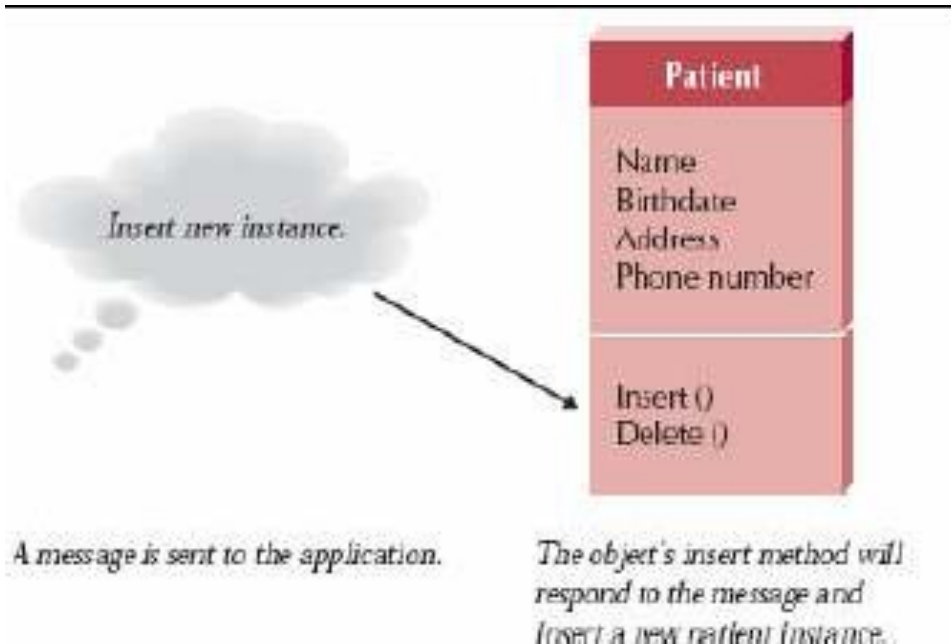


ارسال پیام (MESSAGE SENDING)

○ در یک سیستم ، اشیا با هم تعامل دارند و توسط متدهای همدیگر با هم ارتباط برقرار می کنند.

○ فرستادن پیام به یکدیگر.

- یک شی پیامی را برای اجرای عملیات به شی دیگر می فرستد، شی گیرنده پیام آن عملیات را اجرایی کند.



تناظر یا ارتباط ASSOCIATIONS

- اشیا در بعضی حالات به یکدیگر وابسته هستند.
- هر گونه ارتباط مابین اشیایی از کلاس ها تناظر است (ساده ترین شکل رابطه).
- یک ارتباط یه یک کلاس امکان می دهد تا درباره صفات و رفتارهای کلاس دیگر بداند.
- از طریق ارسال پیام ها و دریافت پاسخها.



تجمع (AGGREGATION)

- تجمع نوع دیگر و قویتر از تناظر یا ارتباط بین اشیا می باشد.
- رابطه تجمع، یک رابطه بین یک واحد کل و جزء است. در رابطه تجمع یک کلاس می تواند شامل نمونه های از کلاس های دیگر نیز باشد.
- در رابطه تجمع، شی جز به شی کل وابسته نیست.
- مثال : تیم پروژه و کارمندان ، ماشین ، موتور و لاستیک و



ترکیب (COMPOSITION)

- ترکیب نوع دیگری از تجمع است که درگیر یک ارتباط قوی بین شی تجمع با اشیای جزاش می باشد.
- رابطه ترکیب، شبیه رابطه تجمع می باشد اما با یک تفاوت:
- در رابطه ترکیب، چرخه حیات جزء نمی تواند بیش از چرخه حیات کل باشد.
- به عبارت دیگر شی جزء هیچ وقت نمی تواند بدون شی کل وجود داشته باشد، شی جزء همزمان با شی کل بوجود می آید و همزمان با شی کل از بین می رود.
- cascading deletion
- مثال: پنجره در سیستم عامل ویندوز

