

# 面向对象编程思想

刘新锋

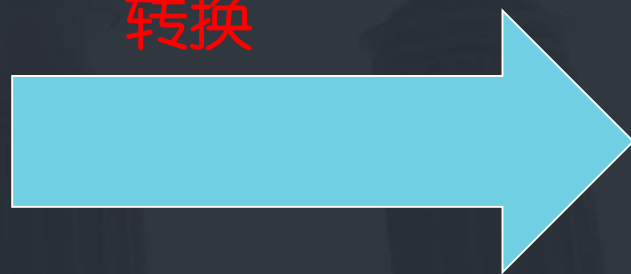


# 面向对象与面向过程程序设计

○ 大千世界



转换



0-1世界



# 面向对象编程概述

简单来说，面向对象编程（Object Oriented Programming, 简称 OOP）描述的是对象之间的相互作用。

面向对象编程主要我们从如下几点论述；

1. 面向过程和面向对象的区别
2. 类和对象：对象 $\leftrightarrow$ 类
3. 面向对象编程的基本特征
4. 面向对象的优点

# 面向对象编程与面向过程编程的区别

过程化程序设计先确定算法，再确定数据结构；面向对象编程先确定数据结构，在确定算法。

面向过程编程的程序员，习惯于建立数据结构存放数据并定义方法（函数）来操作数据；面向对象编程的程序员则构造一个对象模型，将数据与方法组织在一起。

为了更好的理解面向对象编程思想，我们可以举个生活中的例子：  
如何把衣服放入衣柜？

面向过程：为了把衣服装进衣柜，需要3个过程。

1. 把衣柜打开（得到打开的衣柜）
2. 把衣服放进去（打开门后，得到里面存放衣服的衣柜）
3. 把衣柜门关上（打开门、存好衣服、获得关好门的衣柜）

每个过程有一个阶段性的目标，依次完成这些过程，就能把衣服存进衣柜。

面向对象：

为了把衣服存入衣柜，需要做三个动作（或者叫行为）。

每个动作有一个执行者，它就是对象。

1) 衣柜，你给我把门打开

2) 衣柜，你给我把衣服装进去（或者说，衣服，你给我钻到衣柜里去）

3) 衣柜，你给我把门关上

依次做这些动作，就能把衣服存入衣柜。

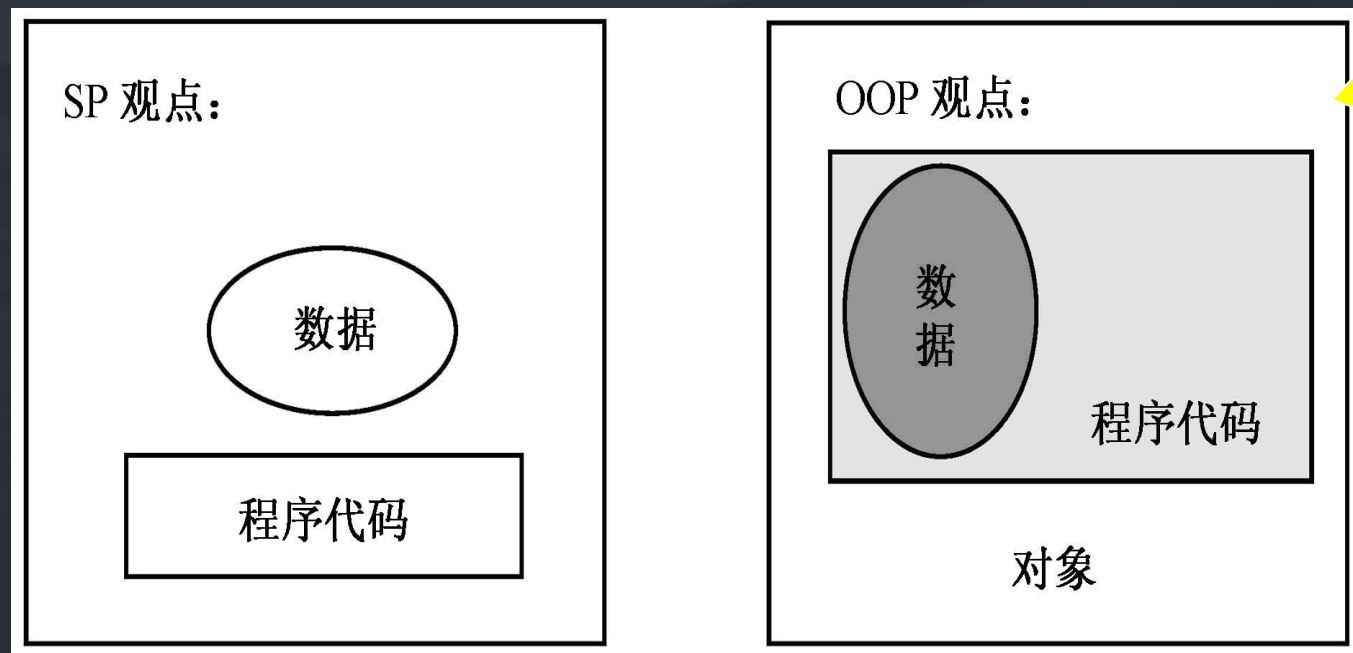
衣柜. 开门()

衣柜. 存入(衣服)

衣柜. 关门()

# 面向对象与面向过程程序设计

## 3、两种程序方法的对比



面向对象程序设计将数据和操作数据的程序代码绑在一起构成对象具有更好的可维护性，因某类对象数据的改变只会引起该类对象操作代码的改变，而与其他类型的对象无关，这就把程序代码的修改维护局限在一个很小的范围内。

# 面向对象程序设计的类和对象

## 对象

对象就是一组变量和相关方法的集合，一个对象有状态、行为等，其中变量表明对象的状态，方法表明对象所具有的行为

## 类

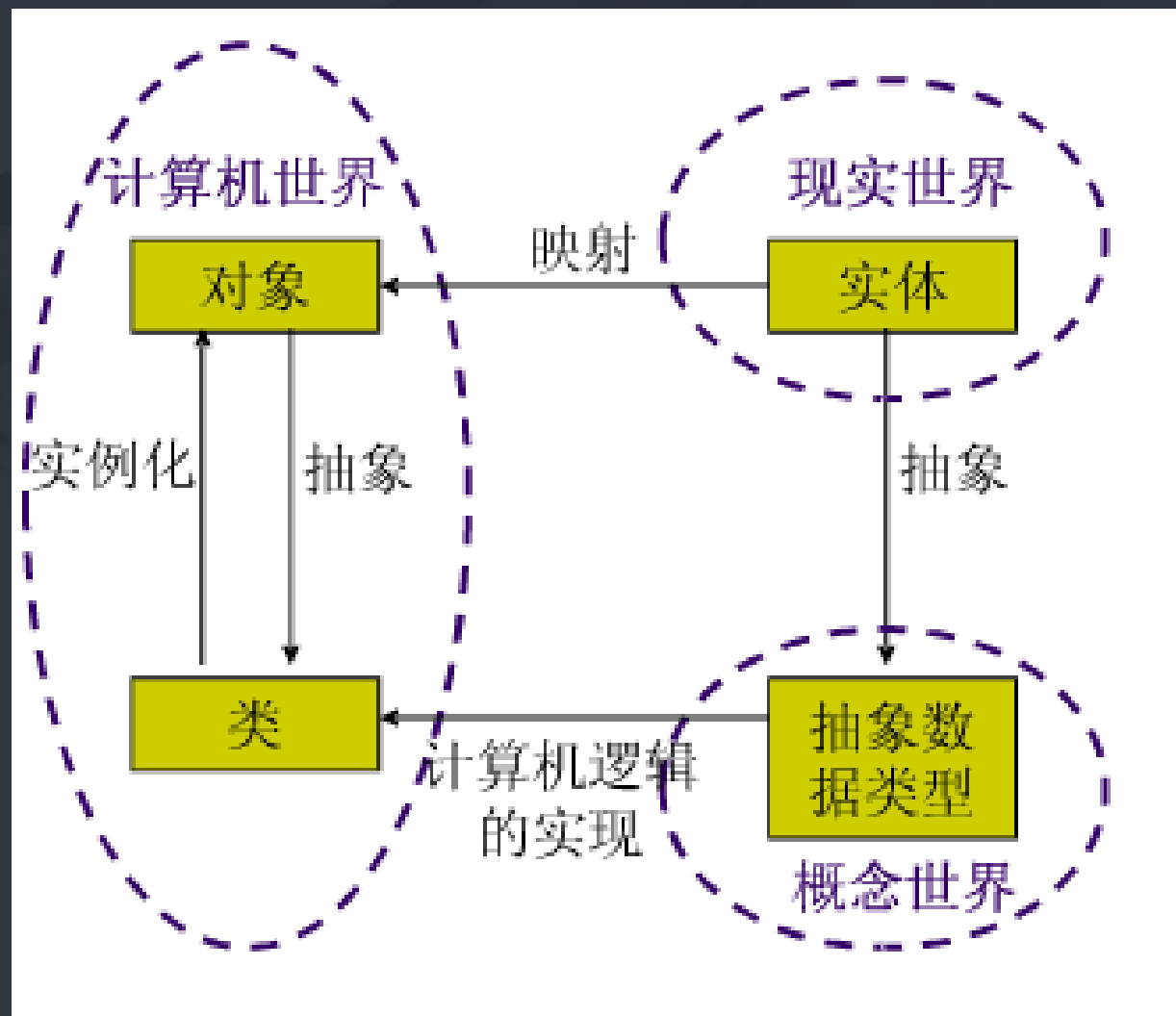
类是一组具有相同数据结构和相同操作的对象集合

类是组成Java的基本要素，它封装了一类对象的状态和方法



# 对象、实体与类之间的关系

类是具有相似属性和行为的一组具有实例（对象）的抽象集合。组成类的对象均为此类的实例，类与实例的关系可以看成是抽象与具体的关系





# 如何定义一个类

```
public class MyDate{  
    private int day,month,year;  
    public MyDate(int day,int monyh,int year){  
        this.day = day;  
        this.month = month;  
        this.year = year;  
    }  
    public int getDate(){....}  
}
```

# 对象的生成

对象的生成

通过new操作符生成一个对象

例如：

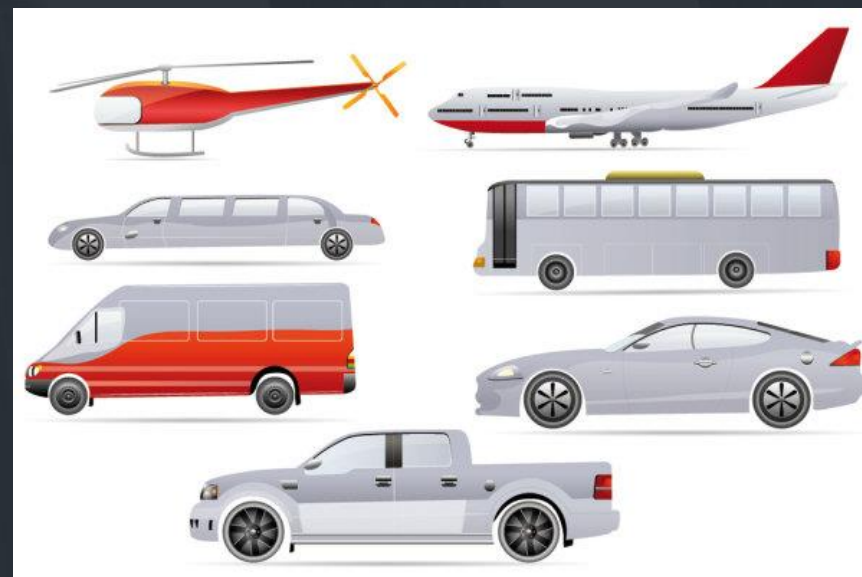
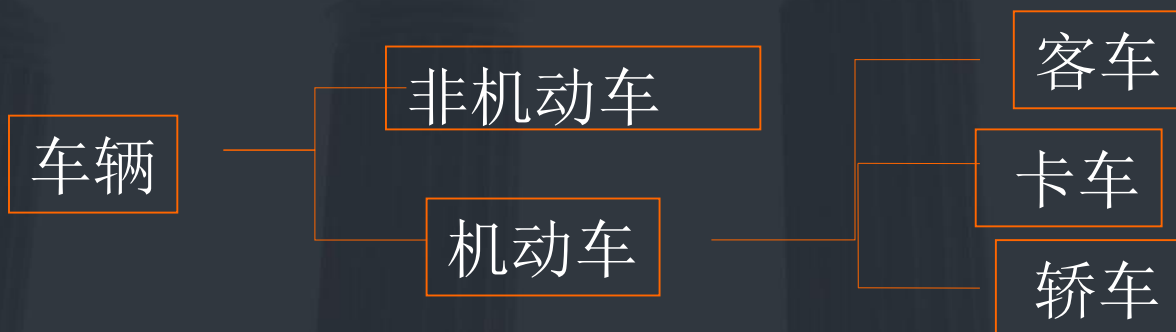
```
MyDate md = new MyDate();
```

# 面向对象的四大特点

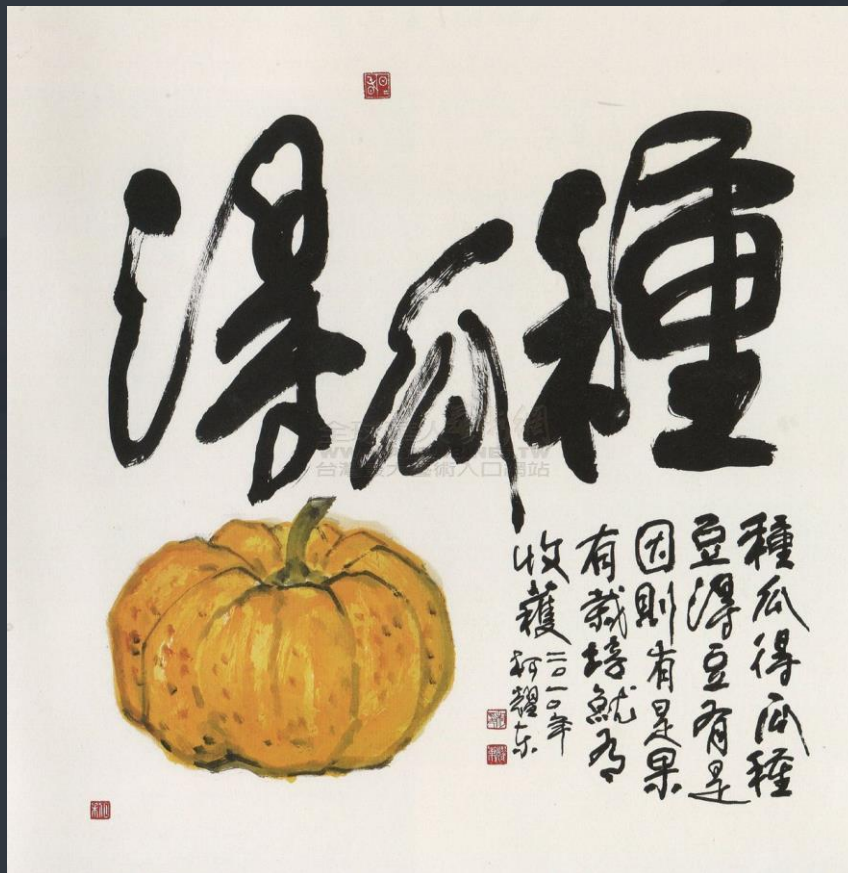


# 抽象 (abstraction)

抽象就是将一类实体的共同特性抽象出来，封装在一个新的类中，所以说抽象是面向对象语言的基础。比如狗是一个对象，但是我们在研究这个对象的时候会把它的一些同类放在一块，一起来考虑，而且抽象的时候，只考虑我们感兴趣的一些数据；某些人可能关心的数据是，狗类的叫声，狗类的蹦跑的方法；某些人可能关心的数据是狗的颜色，狗的大小等等。

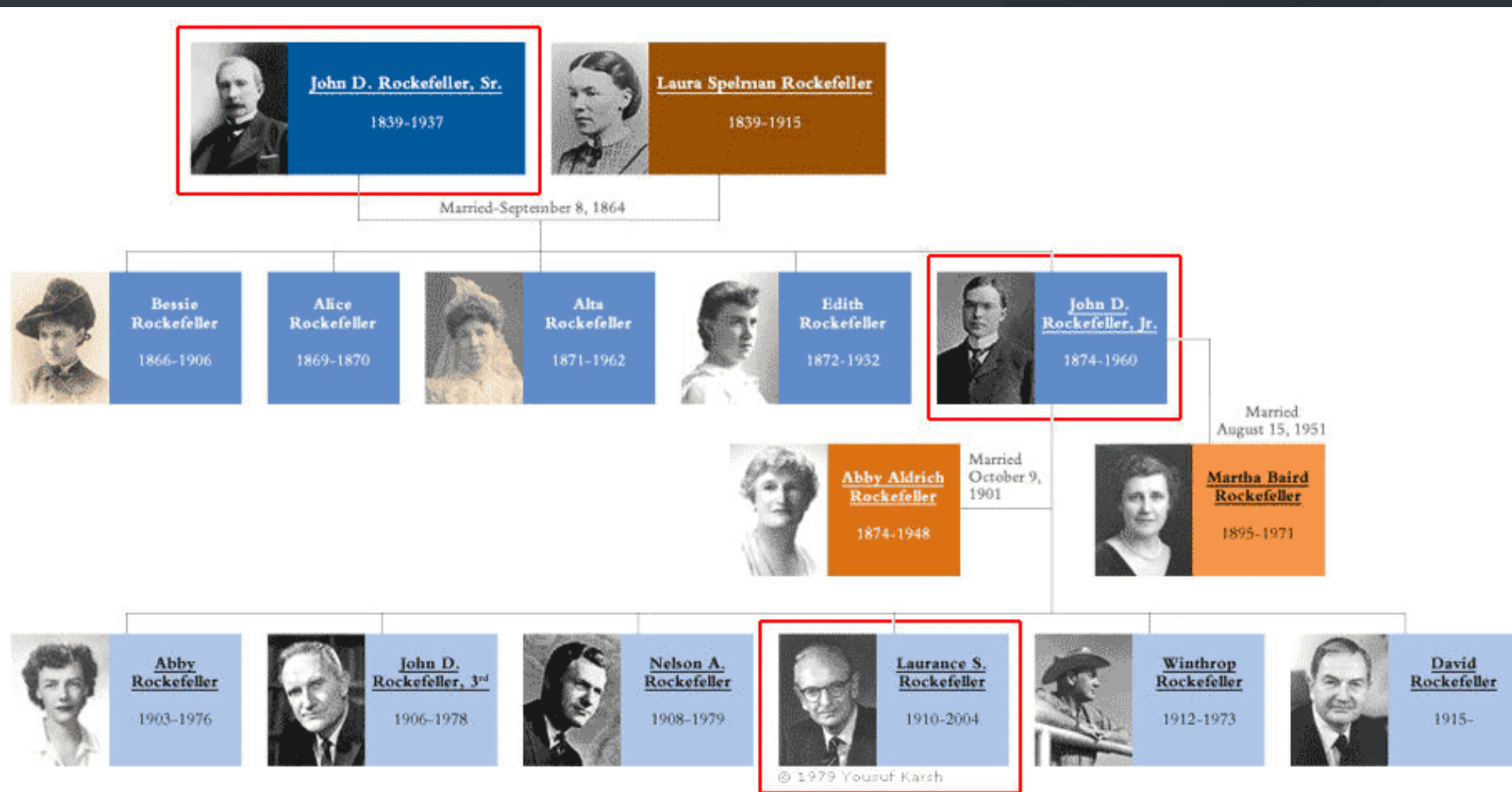


# 继承 (Inheritance)





# 继承 (Inheritance)



编者注：红色圈中者为洛克菲勒家族各代核心人物。

# 继承(Inheritance)

让一个类B去继承另一个类A，那么被继承的类A里的一些公开的可以被B看到的东西都可以被B继承过来，就不必重新开发。其实继承体现的是is-a关系。

继承是一个对象获得另一个对象的属性的过程。

例如 “人” 是一个基类，则 “男人” 是 “人的一个子类。如果’ 小李” 是一个 “男人”，也就是说 “小李” 是 “男人” 的一个对象，那么显然 “小李” 也是 “人” 这个基类的一个对象。

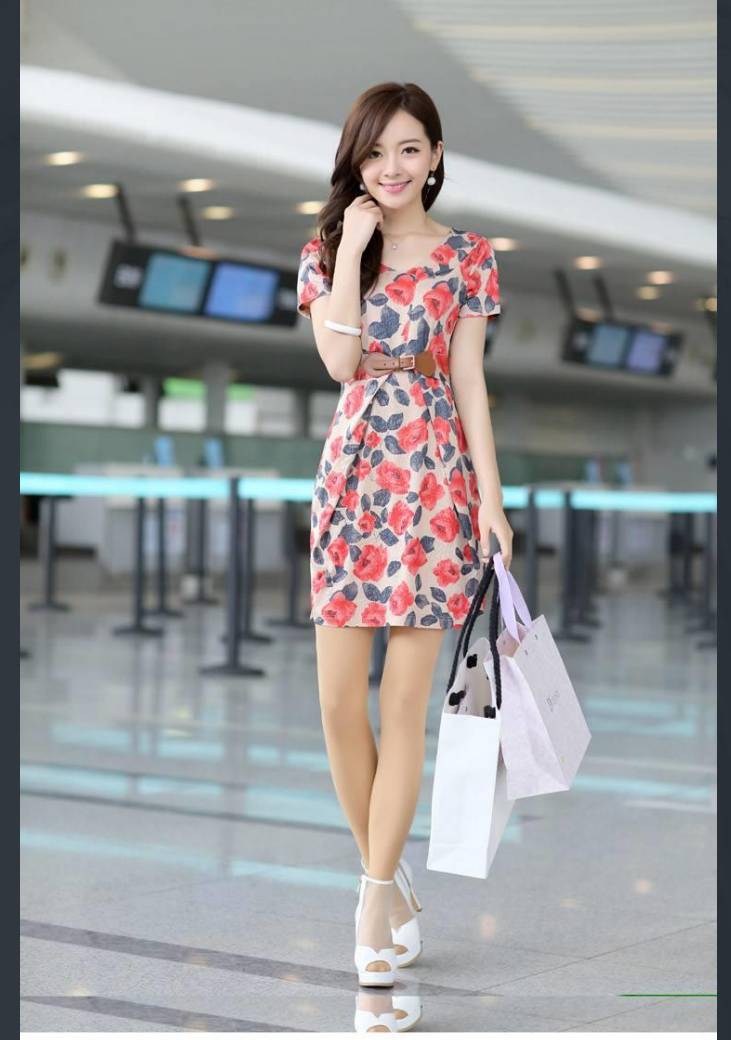
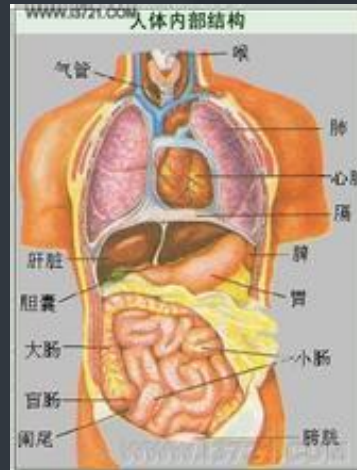
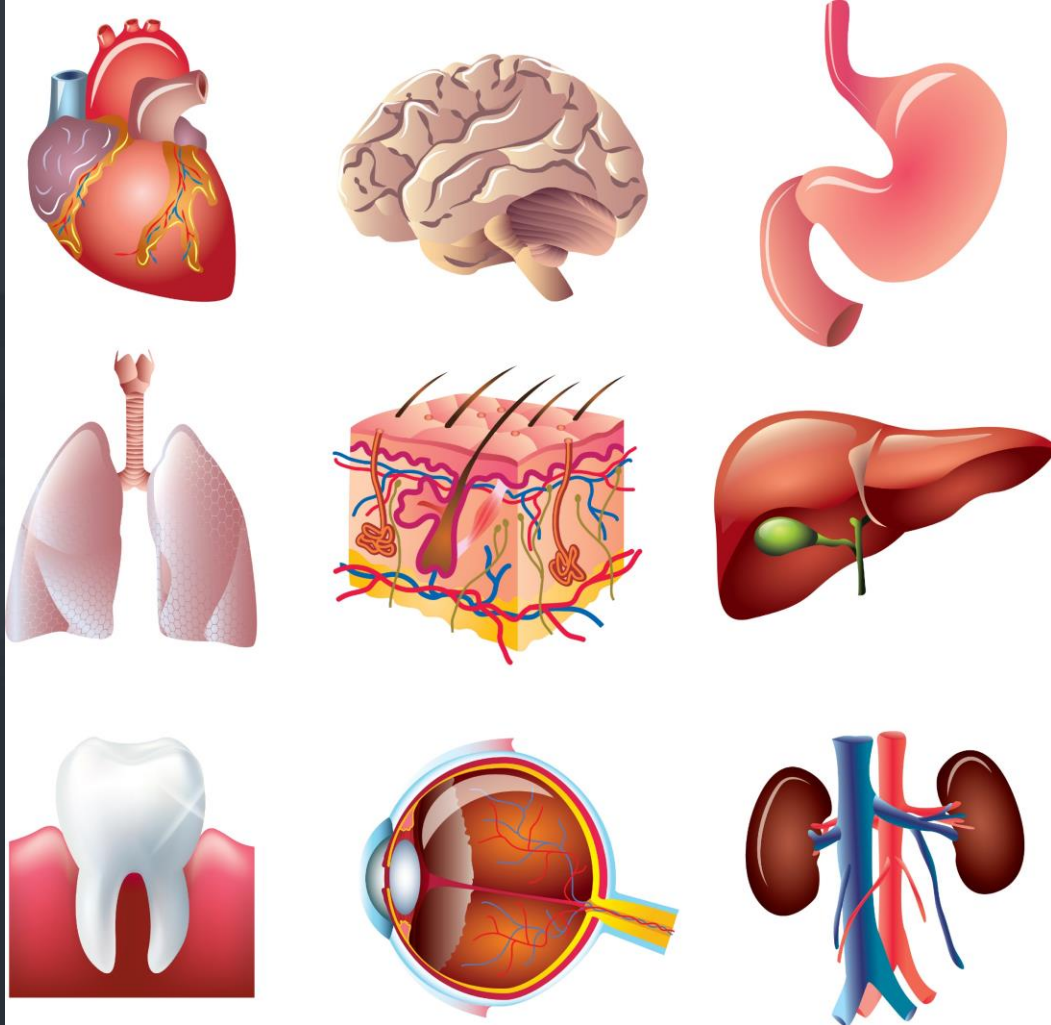




# 封装 (Encapsulation)

1. 封装 (**encapsulation**) 的含义是：类的设计者把类设计成一个黑匣子，使用者只能看见类中定义的公共方法，而看不见方法的实现细节，也不能直接对类中的数据进行操作。
2. 封装的目的有：
  - ① 隐藏类的实现细节；
  - ② 迫使用户通过接口去访问数据；
  - ③ 增强代码的可维护性。

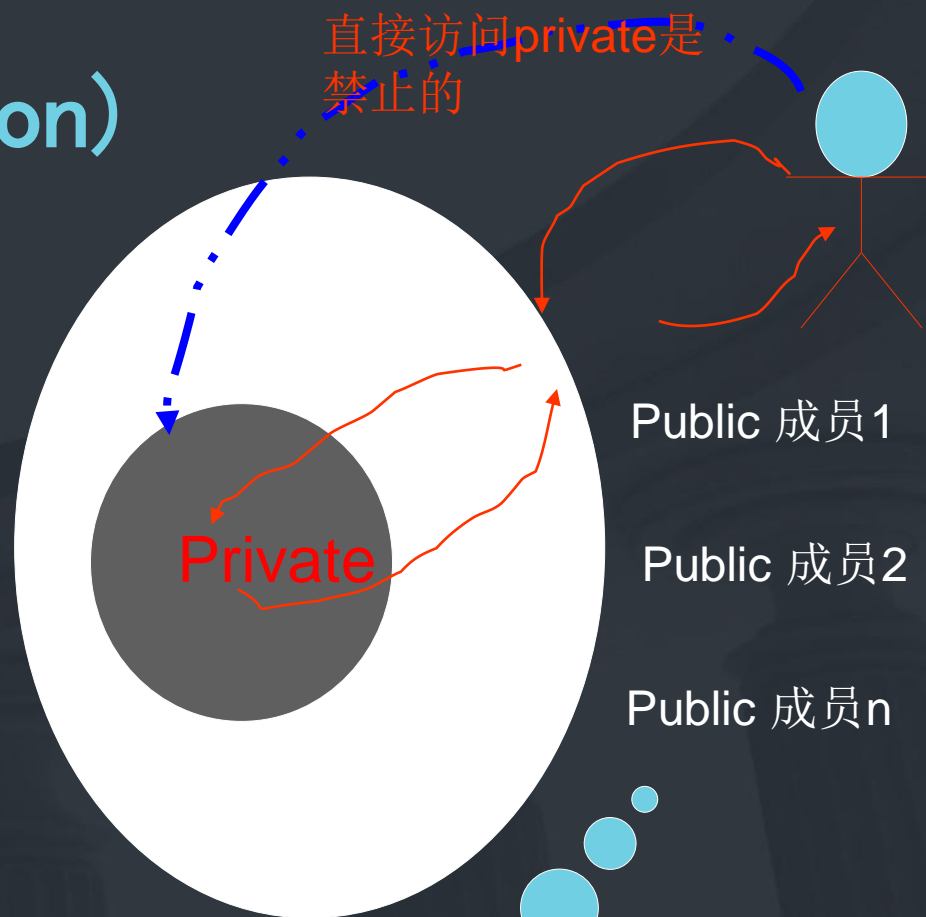
# 封装 (Encapsulation)



# 封装 (Encapsulation)

```
class classname
{
    public:
        //public members
        //friend function
    private:
        //private members
};
```

边界



一个Class定义了一种抽象的数据类型,用户只能访问Public成员,不能直接访问Private成员

# 封装 (Encapsulation)

对象是对一组变量和相关方法封装。通过对象的封装，实现了模块化和信息隐藏

用访问控制符来控制对类成员的访问权限，实现类成员的隐藏。

	同一个类	同一个包	不同包的子类	不同包的非子类
<code>private</code>	✓			
<code>default</code>	✓	✓		
<code>protected</code>	✓	✓	✓	
<code>public</code>	✓	✓	✓	✓



# 多态性(Polymorphism)

- 1.老大囚牛，喜音乐，蹲立于琴头；
- 2.老二睚眦(yá zì)，杀喜斗，刻镂于刀环、剑柄吞口；
- 3.老三嘲风，形似兽，殿台角上的走兽；
- 4.四子蒲牢，洪钟提梁的兽钮，鸣声远扬；
- 5.五子狻猊 (suān ní)，形如狮，喜烟好坐，所以形象一般出现在香炉上，随之吞烟吐雾
- 6.六子霸下，又名**赑屃**(bì xì)，是碑下龟；
- 7.七子狻狂(bì àn)，似虎好讼，狱门或官衙正堂两侧有其像；
- 8.八子负屃 (fù xì)，身似龙，雅好斯文，盘绕在石碑头顶；
- 9.老九螭吻 (chī wěn)，口润嗓粗而好吞，遂成殿脊两端的吞脊兽，取其灭火消灾。



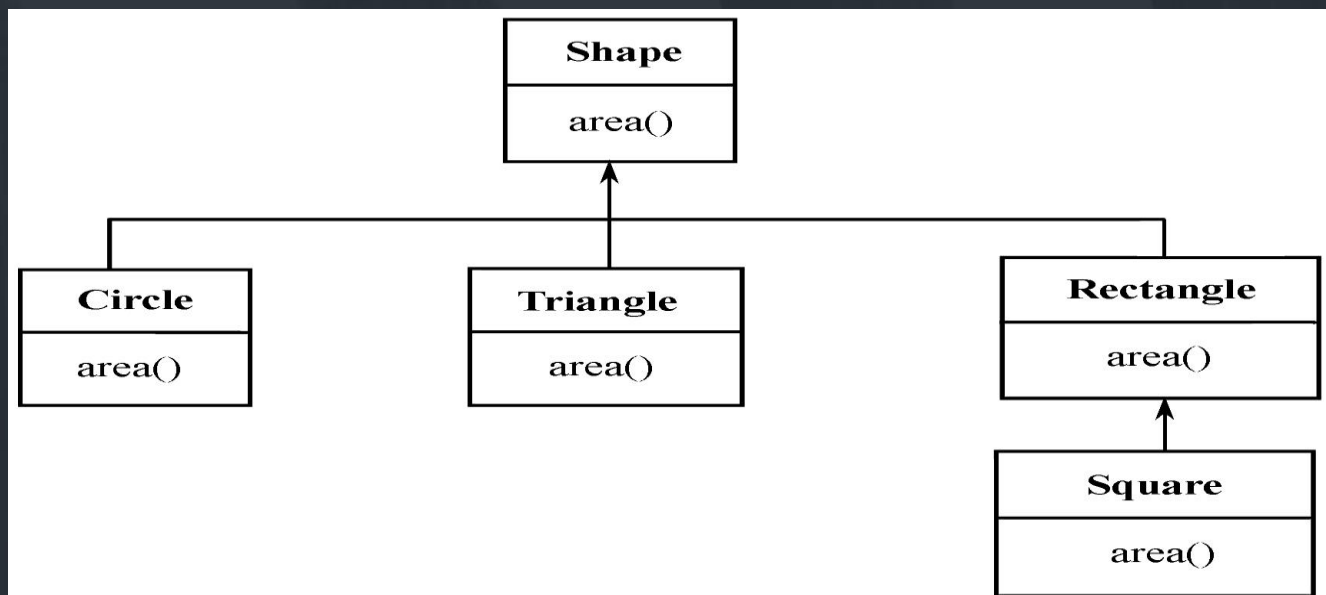
# 多态性 (Polymorphism)

Java语言的多态是指程序中同名的不同方法共存的情况，多态可为程序提供更好的可扩展性，同样也可以代码重用。

实现多态性的三种方式：

1. 通过子类对父类方法的覆盖实现多态；
2. 利用重载在同一个类中定义多个同名的不同方法来实现多态。
3. 接口指向实现类

图中同名函数 `area()` 作用在 `Circle`、`Triangle` 等不同类上时，将执行不同的操作，这就是多态。



# 面向对象思想优点

## 1. 可重用性

编写可重用模块，比如类

## 2. 可扩展性

要求应用软件能够很方便、很容易的进行扩充和修改

## 3. 可管理性

采用封装了数据和操作的类作为构建系统的部件，使项目组织更加方便合理



# 总结

面向对象程序设计的主要特点是以对象作为基本的逻辑构件，用类来描述具有相同特征的对象，整体的代表这类对象，以继承性作为共享机制，共享类中的方法和属性，对象之间则以消息传递的方式进行。与传统的程序设计开发人员不再是仅仅根据某种程序设计语言的语句和表达式来编制程序，而是要求软件开发者通过装配其他编制着提供的可重用的‘软件体’来生产软件，面向对象的程序设计最为显著的特点还是封装性、继承性、多态性。