



# 第一章对象导论

## 1.1 抽象过程

- 问题空间——解空间
  - 编程是翻译的过程
- 编程语言
  - 硬件
  - 汇编
  - C
  - C++/Java

# “纯粹”的面向对象程序设计方法是什么样的？

- 万物皆对象。

- 对象存储数据,接收请求消息,执行操作。

- 程序是一组对象,通过消息传递来告知彼此该做什么。

- 要请求调用一个对象的方法,你需要向该对象发送消息。

- 每个对象都有自己的存储空间。

- 通过封装现有对象,可制作出新型对象。

- 每个对象都有一种类型。

- 每个对象都是某个“类”的一个“实例”。其中,“类”(Class)是“类型”(Type)的同义词。一个类最重要的特征就是“能将什么消息发给它?”。

- 同一类所有对象都能接收相同的消息。

- 类型“圆”(Circle)的对象也是类型“形状”(Shape)的一个对象,所以一个圆完全能接收发送给“形状”的消息。这一特性称为对象的“可替换性”,是OOP最重要的概念之一。

# 什么是对象

- 状态——属性
- 行为——方法
- 标识——名称

## 1.2 接口

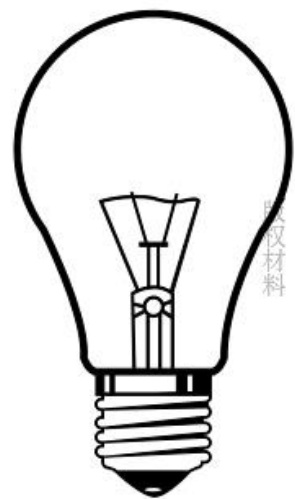
- 定义类型 (class)
  - 标识-名称
  - 状态-属性
  - 行为-方法
- 生成对象 (Object)
  - 可根据类型生成多个

Type Name

Light

Interface

on ()  
off ()  
brighten ()  
dim ()



```
Light lt = new Light();  
lt.on();
```

- 接口确定 “对某个对象能发出的请求”
- 类型完成 “具体实现”



## 1.3 每个对象都提供服务

- 对象, 服务提供者
- 每个对象都很好的完成一件任务
- 不同对象之间通过请求调用传递信息
- 高内聚、低耦合

## 1.4 面向对象三大特征——封装

- 访问控制

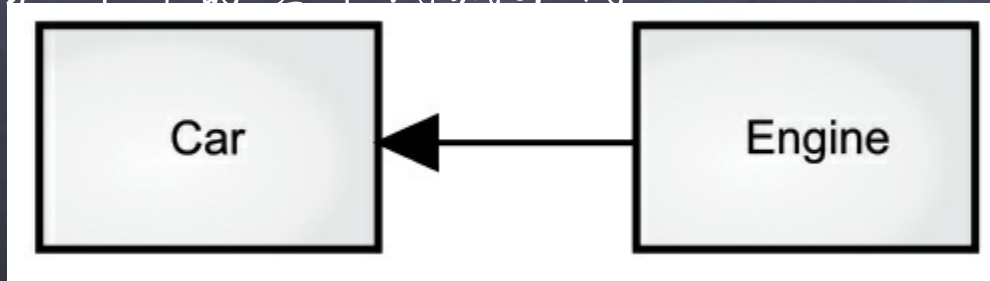
- public protected private friendly

- 客户程序员: 不该碰的不能碰, 减少耦合

- 类创建者: 允许持续性的改进工作, 类内部自由修改

## 1.5 复用

- 组合: 使用现有的类合成新的类
  - 新的类可以由任意数量和任意类型的其他对象构成
  - 内部部件和整件具有相同的生命周期
- 聚合: 动态整合在一起
  - 仍然具有整件和部件的关系
  - 整件和部件的生命周期不同





## 1.6 面向对象三大特征——继承

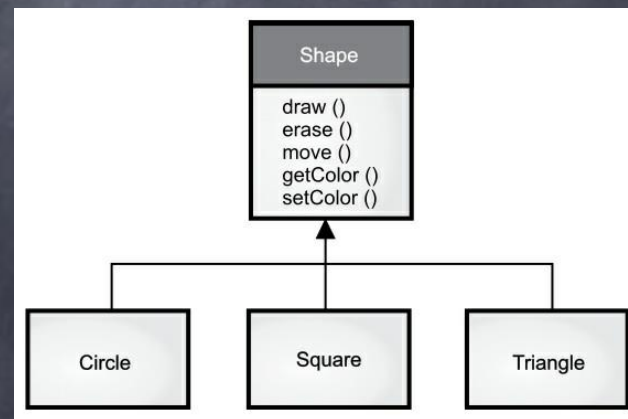
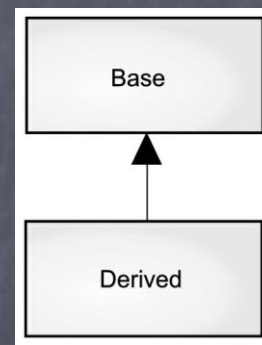
- 基类

- 具有基本功能,规定标准接口

- 导出类

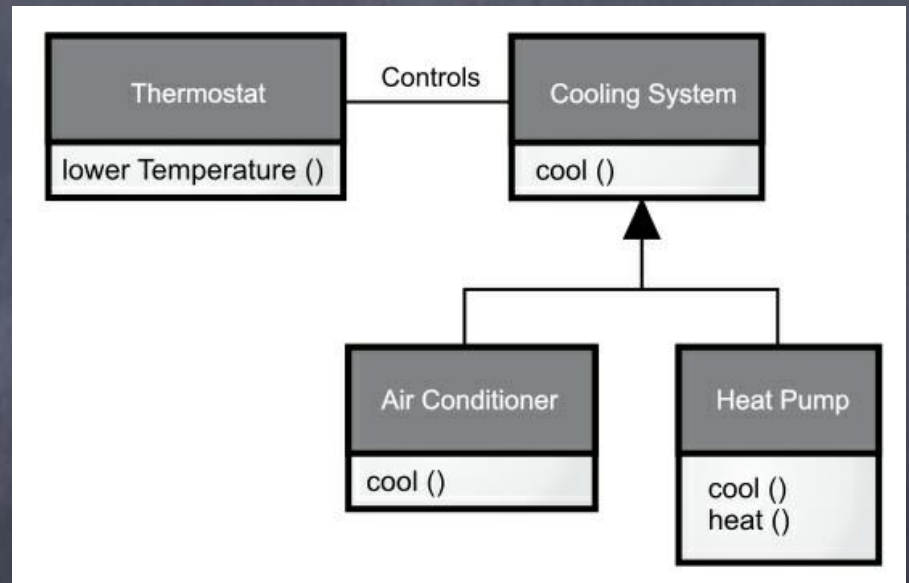
- 扩展功能,在基类的基础上增加新功能
- 替换功能,也叫“覆盖”,  
Override,更新基类某些方法的版本

- 父类更通用,子类更具体。



## “is-a” 与 “is-like-a”

- “is-a” 关系
  - 导出类只覆盖基类
- “is-like-a” 关系
  - 导出类扩展了基类

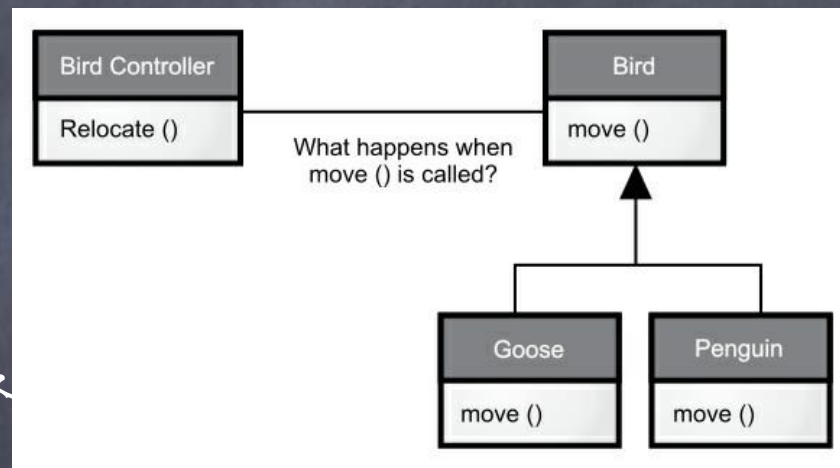


## 1.7 面向对象三大特征——多态

- 同一行为具有多个不同表现形式或形态的能力。

- 多态的必要条件

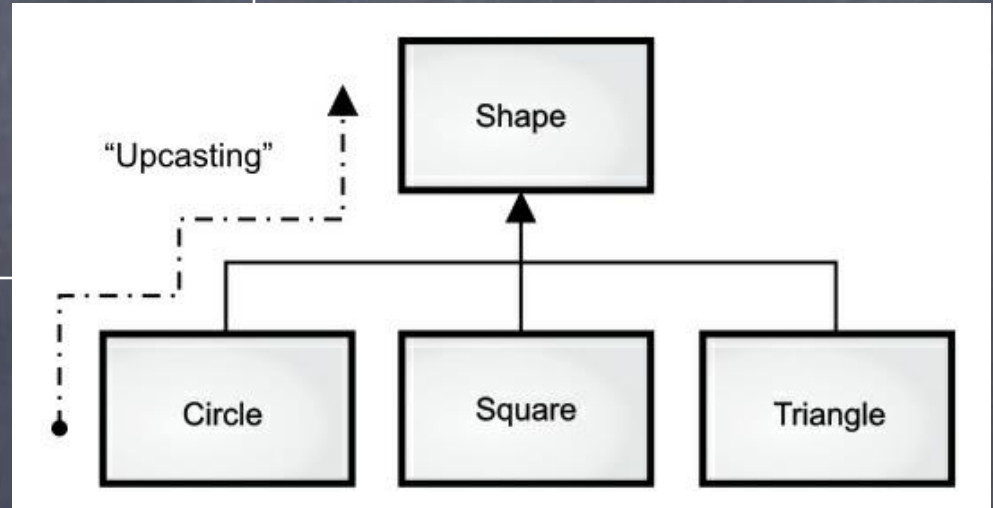
- 继承
- 覆盖
- 把导出类对象当做基类对象



- 优点

- 扩展性
- 可替换
- 解耦合

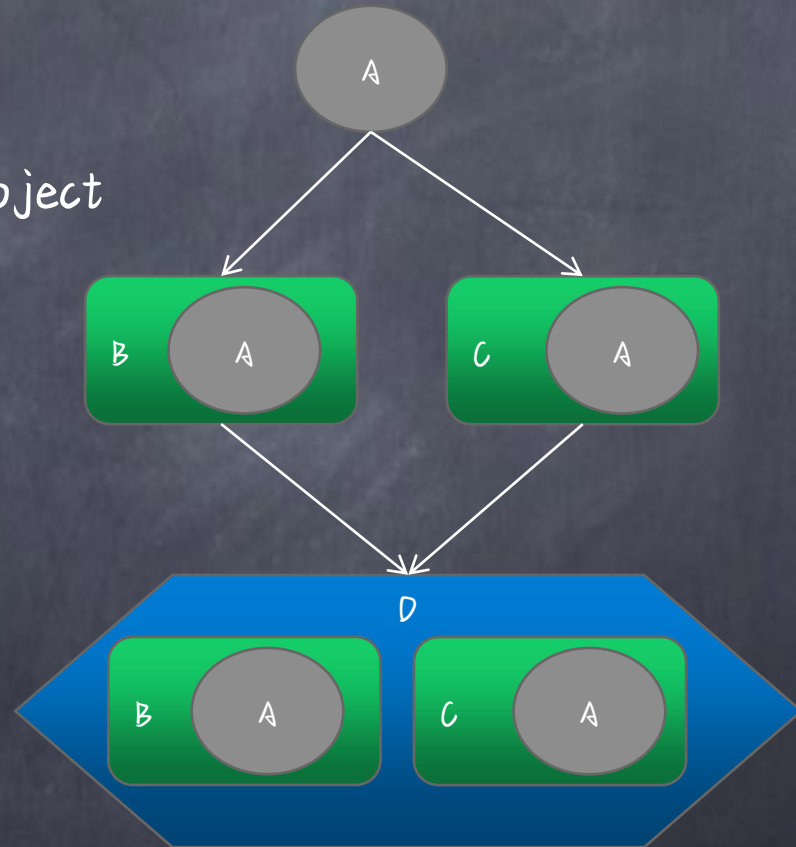
```
void doSomething(Shape shape) {  
    shape.erase();  
  
    // ...  
  
    shape.draw();  
}
```



```
Circle circle = new Circle();  
Triangle triangle = new Triangle();  
Line line = new Line();  
doSomething(circle);  
doSomething(triangle);  
doSomething(line);
```

## 1.8 单根继承

- C++的多重继承
  - 结构复杂
- Java的所有类型的共同父类, Object
- 优点
  - 统一的接口
  - 更高的兼容性
  - 便于垃圾收集
  - 灵活性





## 1.9 容器

- 能容纳不定数量的相同类型对象的集合
  - 数组、链表、字典等
- Java提供的容器
  - List
  - Set
  - Map
- 泛型(参数化类型机制)
  - `ArrayList<Shape> shapes = new ArrayList<>();`
- 特点
  - 不同容器提供了不同的接口与行为
  - 不同容器具有不同的效率

# 1.10 对象创建和生命周期

对象所占用的资源

内存、文件句柄、网络  
端口、数据库服务等



Java的对象控制策略

效率与安全性的折中

- | 对象在堆中动态创建
- | 垃圾回收,但注意仅回收内存

C++的对象控制策

略  
追求效率

- | 对象可以在堆、栈、静态区等创建
- | 更加危险,由程序员来控制创建的位置与回收的时间

## 1.11 异常处理

### C语言

- | 没有异常处理机制
- | 错误检查是一种编程技巧



### C++语言

- | 有异常处理机制
- | 可忽略

### Java语言

- | 内置异常处理机制
- | 强制使用
- | 提高鲁棒性

# 1.12 并发编程

## 进程切换

- | 操作系统支持
- | 切换代价高

02

## 中断服务程序

- | 效率高
- | 编写困难
- | 移植性差

01

03

## 线程切换

- | 操作系统支持
- | 切换代价低
- | Java语言内置

## 1.13 Java 与 Internet

- 古老又现代的客户端/服务器计算
  - 终端、分布式、云计算
- 服务器端编程技术
  - CGI、Perl、C++、Python
  - Java Servlet/JSP、.NET
- 客户端编程技术
  - 浏览器插件
  - 脚本语言, JavaScript, VBScript
  - JavaApplet, Flash, HTML5



# 总结

- 面向过程的编程技术
  - 数据+函数
  - 更加偏重计算机
- 面向对象的编程技术
  - 对象+消息
  - 更加偏重问题本身

## 我的第一个Java应用程序

- 1 // filename : HelloWorld.java
- 2 /\* Hello World, the first Java application \*/
- 3 class HelloWorld {
- 4     public static void main (String args[ ]) {
- 5         System.out.println("Hello World!");
- 6     }
- 7 }
- 将上述内容保存为 “HelloWorld.java” 大小写
- 区分

# Java的编译与运行

- Javac: java编译器, 编译Java源文件
  - Javac HelloWorld.java
- Java: 启动Java虚拟机, 运行字节码
  - Java HelloWorld



# 源代码分析

- 第 1、2 行为注释行
- 第 3 行为类说明
  - Java 的源文件就是类说明文件
  - 且其文件名必须与类名相同
- 第 4 行相当于 C 中的 main 函数, 是整个程序的入口点。其定义形式必须如此。其中:
  - public -- 指明 main 方法可被任何类引用
  - static -- 指明 main 方法是静态的
  - void -- 指明 main 方法不返回任何值
  - String args[] -- main 的参数表, 是个字符串数组
- 第 5 行相当于 C 中的 printf();

## “Hello, World”的编译与运行

- 编译命令行: `javac HelloWorld.java`
- 编译通过, 产生 `HelloWorld.class` 文件
- 运行命令行: `java HelloWorld`
- 相关的环境变量
  - `JAVA_HOME`: JDK的安装目录
  - `PATH`: `d:\java\bin`
  - `CLASSPATH`: Java解释器寻找.class文件的路径, 如 “`.;d:\myJava\`”



## Java编码风格

- 类名的首字母大写
- 如果类名由几个单词组成,就把他们并在一起(不要用下划线),其中每个单词的首字母大写
- 方法、成员变量、对象引用等几乎所有其它内容,其风格与类名一样,除了首字母小写。
- 花括号的摆放方式
- 对齐方式:“tab”

# 链接

- Java编程思想
- 链接:<https://pan.baidu.com/s/194QL1e-CMvG7VUvO2pfYRw>
- 提取码:5b8y
  
- JDK7
- 链接:<https://pan.baidu.com/s/1EWQJ3SLaZdCVG4z06VdDmw>
- 提取码:761k
  
- eclipse 64位
- 链接:<https://pan.baidu.com/s/1isL6b983yXJ4t004rHWZoA>
- 提取码:4y1f



# 作业

- 安装/熟悉系统,学习命令行操作
- 编写一个Application,该程序需要输出你的姓名,学号,班级。这个程序只需要一个类,类中只需要一个方法(“main”方法,在程序启动时被执行)。注意类名与文件名,并且大小写相关
  - 记住要把main方法设成static,并指定参数列表。
  - 用javac编译,用java运行
  - 熟悉Java编程风格
- 学习eclipse操作



提问