

面向对象的编程 和编程语言

- 程序设计范型
- 面向对象的编程语言
- 为实现OOD模型
选择编程语言
- 几种典型的面向
对象的编程语言
简介
- 用非OO编程语
言实现OOD模型

三、为实现OOD模型选择编程语言

在OOD完成之后，选择什么编程语言实现OOD模型？

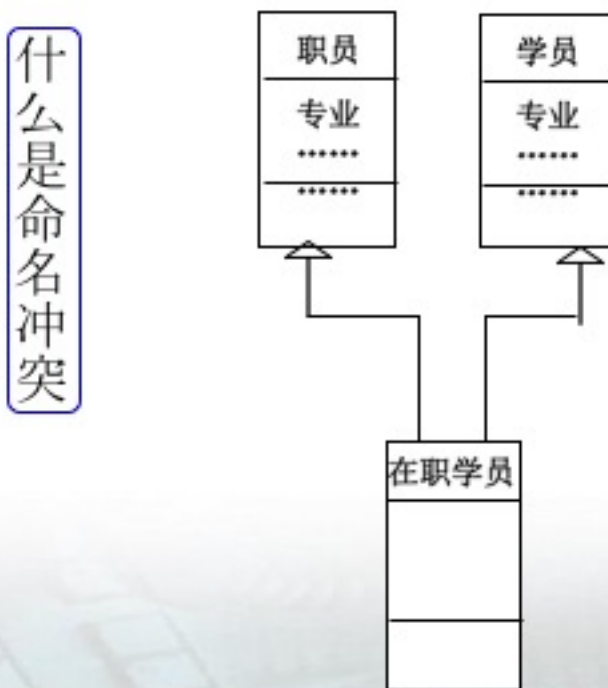
1、一般原则

- * 基本原则——语言的选择完全从实际出发
主要考虑成本、进度、效率等实际因素
- * OOPL是实现OOD的理想语言
它使源程序能很好的对应OOD模型。
- * 带有类库、编程环境、权限管理的OOPL更好。
- * 用非OO语言也能实现OOD模型
缺乏OO机制的保证和支持，
但若自觉遵循一定的原则，可以保持某些OO风格。



2、编程语言的评价标准

- (1) 能否描述类和对象
是否提供封装机制？
对封装有无可见性控制？
- (2) 能否实现一般-特殊结构
支持多继承、单继承还是不支持继承？
支持多继承时，是否能解决命名冲突？
是否支持多态？



- 职员类的“专业”是该职员从事的专业；
- 学员类的“专业”是该学员学习的专业。

问题：

“在职学员”类同时继承了两个“专业”属性，引用时无法判断指的是哪一个。



北京大学

面向对象的编程和编程语言

- 程序设计范型
- 面向对象的编程语言
- 为实现OOD模型选择编程语言
- 几种典型的面向对象的编程语言简介
- 用非OO编程语言实现OOD模型



面向对象的编程 和编程语言

- 程序设计范型
- 面向对象的编程语言
- 为实现OOD模型
选择编程语言
- 几种典型的面向
对象的编程语言
简介
- 用非OO编程语
言实现OOD模型

- (3) 如何实现整体-部分结构
用什么实现？如何表示多重性？
- (4) 如何实现属性和操作
用什么表示属性？用什么描述操作？
有无可见性控制？
能否描述约束？
是否支持动态绑定(dynamic binding)？
- (5) 如何实现关联和消息通讯
用什么实现关联？如何表示多重性？
如何实现消息通讯？
- (6) 其它可考虑的因素（反映于具体的语言版本）
是否带有可视化编程环境
是否带有类库
能否支持对象的永久存储





面向对象的编程 和编程语言

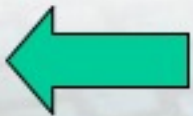
- 程序设计范型
- 面向对象的编程语言
- 为实现OOD模型选择编程语言
- 几种典型的面向对象的编程语言简介
- 用非OO编程语言实现OOD模型

注释：

绑定：一个对象（或事物）与其某种属性建立某种联系的过程。
如：一个变量与其类型或值建立联系，一个进程与一个处理器建立联系等。——《计算机科学技术百科全书(第二版)》

在计算机语言中有两种主要的绑定方式：

- **静态绑定**发生于数据结构和数据结构间，程序执行之前。静态绑定发生于编译期，因此不能利用任何运行期的信息。它针对函数调用与函数的主体，或变量与内存中的区块。
- **动态绑定**则针对运行期产生的访问请求，只用到运行期的可用信息。在面向对象的代码中，动态绑定意味着决定哪个方法被调用或哪个属性被访问，将基于这个类本身而不基于访问范围。





面向对象的编程 和编程语言

- 程序设计范型
- 面向对象的编程语言
- 为实现OOD模型
选择编程语言
- 几种典型的面向
对象的编程语言
简介
- 用非OO编程语
言实现OOD模型

四、几种典型的面向对象的编程语言简介

C++——Visual C++
Object Pascal——Delphi
Smalltalk
Objective-C
Eiffel
Java

**重点： C++, Smalltalk, Eiffel,
Java**

面向对象的编程 和编程语言

- 程序设计范型
- 面向对象的编程语言
- 为实现OOD模型选择编程语言
- 几种典型的面向对象的编程语言简介
- 用非OO编程语言实现OOD模型

1、C++

由AT&T的Bell试验室开发，1988年推出产品

是在C语言的基础上扩充OO特征而得到的
是C语言的超集

是一种混合型的OOPL

保持C语言的高效率、可移植，与C兼容
使广大程序员容易接受

采用强类型机制

支持动态绑定

是目前使用最广的OOPL



北京大学



面向对象的编程 和编程语言

- 程序设计范型
- 面向对象的编程语言
- 为实现OOD模型选择编程语言
- 几种典型的面向对象的编程语言简介
- 用非OO编程语言实现OOD模型

(1) 类和对象

类 \Rightarrow class

对象 \Rightarrow object

封装机制：有

可见性控制：private、protected、public、friend

对象的创建和删除：

提供构造函数（constructor）—— 类名（）

（C++规定，每个类必须有默认的构造函数，没有构造函数就不能创建对象。）

和析构函数（destructor）—— ~类名（）

（析构函数的定义：析构函数也是特殊的类成员函数，它没有返回类型，没有参数，不能随意调用，也没有重载，只有在类对象的生命期结束的时候，由系统自动调用。）

静态对象（从程序开始执行到退出）

创建：类名 对象名

删除：程序退出时

动态对象（显式地创建和删除）

创建：对象指针 = new 类名（参数）

删除：delete 对象指针



北京大学



面向对象的编程 和编程语言

- 程序设计范型
- 面向对象的编程语言
- 为实现OOD模型选择编程语言
- 几种典型的面向对象的编程语言简介
- 用非OO编程语言实现OOD模型

(2) 一般-特殊结构

基类-派生类 (base-derived class)

支持单继承和多继承

用命名空间 (namespace) 解决命名冲突

支持多态和重载

在继承的同时可定义被继承内容的可见性

(3) 整体-部分结构 (聚合)

用指针或嵌套对象(**nested object**)实现聚合, 用指针数组或指针链表等实现多重性。



北京大学



面向对象的编程 和编程语言

- 程序设计范型
- 面向对象的编程语言
- 为实现OOD模型选择编程语言
- 几种典型的面向对象的编程语言简介
- 用非OO编程语言实现OOD模型

(4) 属性、操作

属性⇒成员变量

可见性: **private**, **protected**, **public**, **friend**
类中的变量可说明为静态的(即为其所有的对象共享)

操作⇒成员函数

构造函数: 与类名相同, 可以有多个

析构函数: ~类名

可见性: 同属性

没有显式的约束

动态绑定: 声明为**virtual**的函数是动态绑定的,
在基类中定义的**virtual**的函数, 可以在派生类中改写

(5) 关联和消息

用指针实现关联

多重关联可以用指针数组

消息: 用函数调用实现, 采用强类型的参数





术语对照

OOA和OOD

C++

对象 object

对象 object

类 class

类 class

属性 attribute

成员变量 member variable

操作 operation

成员函数 member function

一般/特殊

基类/派生类

generalization/specialization

base class/derived class

整体/部分 whole/part

嵌套对象 nested object

嵌入指针 embedded pointer

消息 message

函数调用 function call

关联 association

对象指针 object pointer



北京大学

面向对象的编程 和编程语言

- 程序设计范型
- 面向对象的编程语言
- 为实现OOD模型选择编程语言
- 几种典型的面向对象的编程语言简介
- 用非OO编程语言实现OOD模型

面向对象的编程 和编程语言

- 程序设计范型
- 面向对象的编程语言
- 为实现OOD模型选择编程语言
- 几种典型的面向对象的编程语言简介
- 用非OO编程语言实现OOD模型

2、Java

- ❖ Java是由Sun公司于1995年推出的，适合于分布式环境，独立于平台。
- ❖ Java是纯面向对象，语法与C++基本一致，但去掉了C++的非面向对象成分。
- ❖ Java不使用指针，解释执行。
- ❖ Java对分布式和客户/服务器结构的支持，提供了丰富的类库和方便有效的开发环境，并提供了语言级的多线程、同步原语和并发控制机制。





面向对象的编程 和编程语言

- 程序设计范型
- 面向对象的编程语言
- 为实现OOD模型选择编程语言
- 几种典型的面向对象的编程语言简介
- 用非OO编程语言实现OOD模型

(1) 类和对象

定义类的关键字: `class`

封装机制: 有

对象的创建和删除:

构造函数: 类名 ()

终止函数: `finalize()`

对象的静态声明和动态创建类似于C++

(2) 属性和操作

属性: 实例变量、类变量

可见性: `private`、`protected`、`public`和`friendly`

操作: 实例方法、类方法

可见性: 同属性

(3) 继承

超类/子类

支持重载和多态

(4) 关联、聚合

可用对象引用实现关联或聚合



北京大学

面向对象的编程 和编程语言

- 程序设计范型
- 面向对象的编程语言
- 为实现OOD模型选择编程语言
- 几种典型的面向对象的编程语言简介
- 用非OO编程语言实现OOD模型

五、用非OO编程语言实现OOD模型

1、过程语言——以C为例

(1) 类, 对象 \Rightarrow 无

可以用结构定义对象, 可通过指针说明应该有哪些函数(不封装)

(2) 一般-特殊 \Rightarrow 无

可把一般结构嵌入特殊结构

(3) 整体-部分 \Rightarrow 指针、嵌套的结构

(4) 属性与操作 \Rightarrow 变量、函数

(5) 关联 \Rightarrow 指针; 消息 \Rightarrow 函数调用

