

# 面向对象程序设计 ( Java )

主讲教师：申雪萍

shenxueping@buaa.edu.cn



2025/9/10

Xueping Shen



北京航空航天大学  
COLLEGE OF SOFTWARE  
BEIHANG UNIVERSITY 软件学院

# About the course

主讲教师：申雪萍  
[shenxueping@buaa.edu.cn](mailto:shenxueping@buaa.edu.cn)



2025/9/10

Xueping Shen



北京航空航天大学  
COLLEGE OF SOFTWARE  
BEIHANG UNIVERSITY 软件学院

- 主讲老师
  - 申雪萍
- 助教
  - 赵诗阳
  - 程海尚

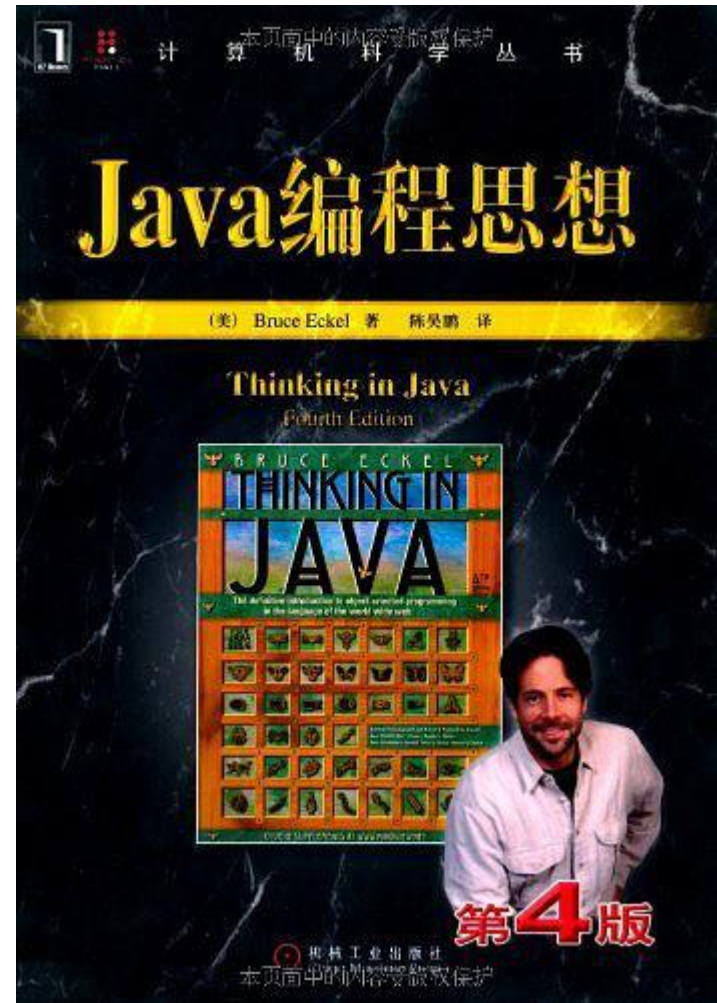
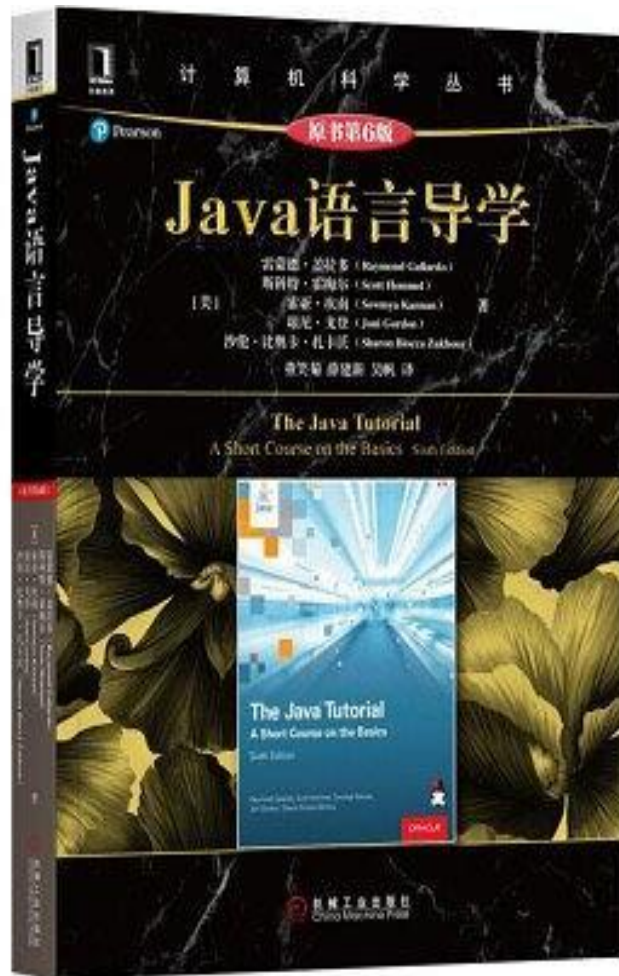
# 课程目标

- Java是软件开发领域最为重要的编程技术之一，其技术栈全面，开源技术和开源框架众多、安全稳健、简单易用，发展前景仍旧十分广阔。因此“Java程序设计”作为本科计算机和软件专业的基础课，一直被公认为是大学期间提升学生编程能力、系统能力和工程能力的最重要的课程之一。
- 本课程借助Java作为教学语言，向学生系统地介绍面向对象的程序设计方法和编程思想，培养学生的面向对象编程思维；
- 提升学生的程序设计能力、系统能力和工程能力；

# 课程主要内容

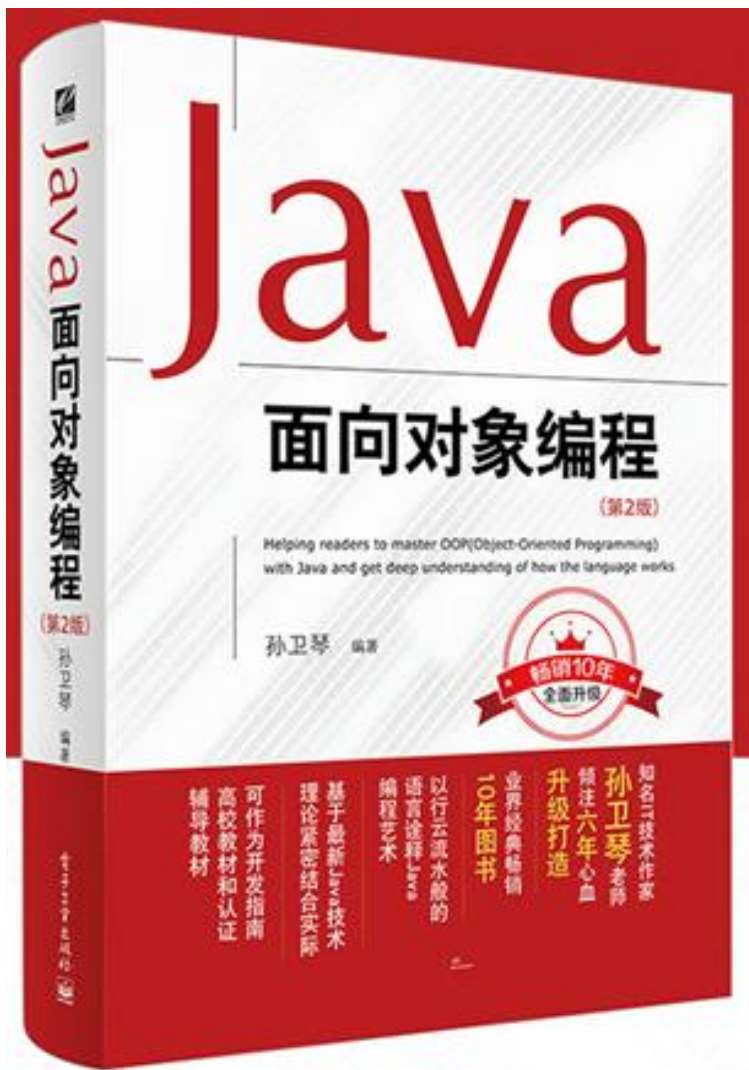
一级目录	二级目录
面向对象基础	第一章 认识对象：面向对象的编程基础
	第二章 认识对象：封装数据为类
	第三章 面向对象的三大特称之一：封装
	第四章 面向对象的三大特征之二：继承
	第五章 面向对象的三大特称之三：多态
设计原则和设计模式	第六章 完善类的设计
数据和算法	第七章 面向对象和数据结构
增强软件的鲁棒性	第八章 异常处理
软件交互	第九章 图形界面程序设计
输入输出	第十章 文件与输入输出
并发处理	第十一章 多线程程序设计
Socket编程	第十二章 网络程序设计

# Textbook and References





## Textbook and References: 中文教程



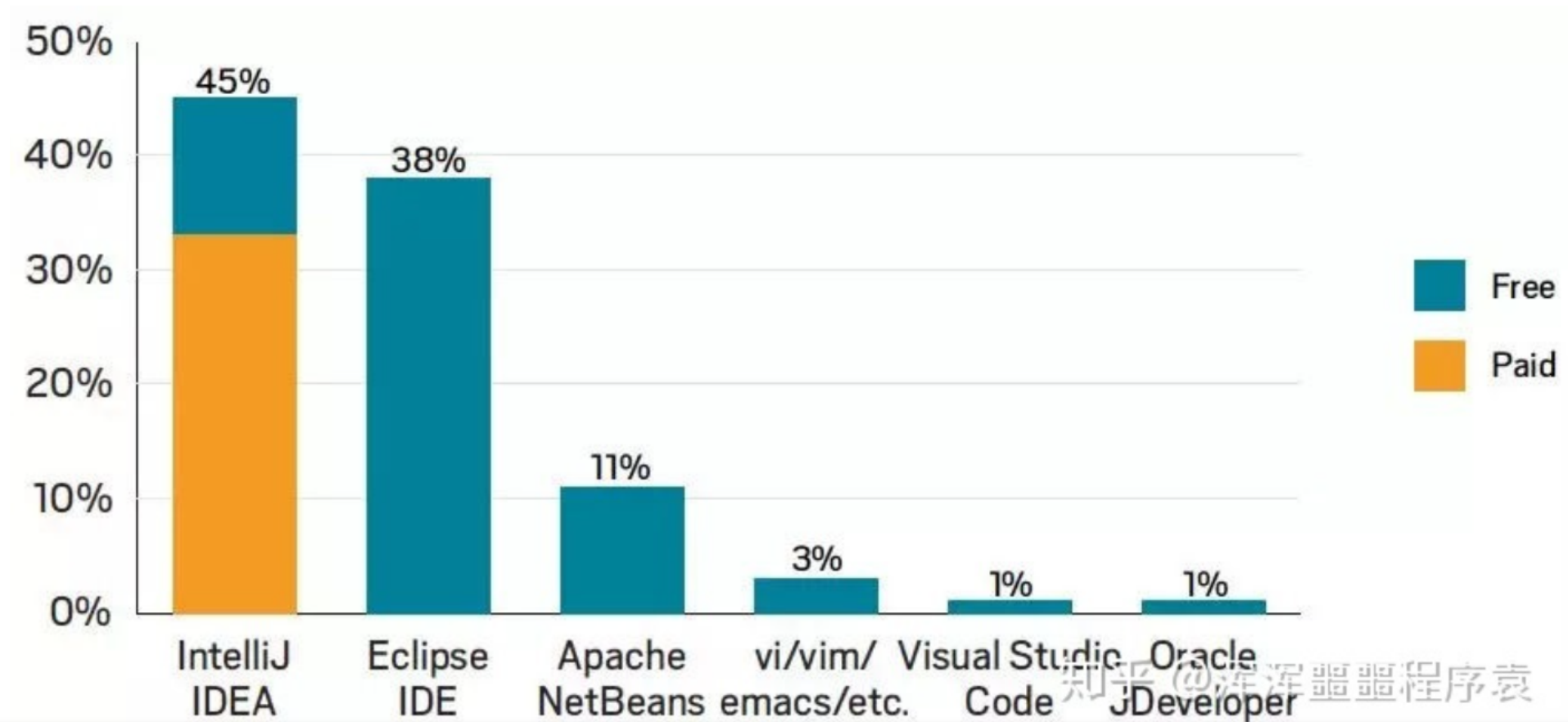
- Java2实用教程 耿祥义 教学视频
- Java面向对象编程 孙卫琴 教学视频
  - 网址: <http://www.javathinker.net/javabook.jsp>



# Lab Exercises environment

- **Windows、MAC、Linux**
- **JDK 8及以上**
- **IDE**
  - **Eclipse (开源)**
  - **IntelliJ IDEA (免费试用1年)**
  - **JCreator**
  - **Jbuilder**
  - **NetBeans**
  - **VS Code 等**

调查报告显示，有 45% 的开发者以 IntelliJ IDEA 为开发 Java 项目的 IDE。其次分别有 38% 的开发者使用免费的 Eclipse，以及有 11% 的受访者使用 NetBeans。





















http://www.eclipse.org/downloads/ Eclipse 搜索 邮

Eclipse Downloads

Packages Developer Builds Projects

Compare Packages Older Versions Eclipse Helios (3.6.1) Packages for Windows

	<b>Eclipse IDE for Java Developers</b> , 99 MB Downloaded 1,626,683 Times <a href="#">Details</a>	 Windows 32 Bit Windows 64 Bit
	<b>Eclipse IDE for Java EE Developers</b> , 206 MB Downloaded 1,116,763 Times <a href="#">Details</a>	 Windows 32 Bit Windows 64 Bit
	<b>Eclipse Classic 3.6.1</b> , 170 MB Downloaded 520,504 Times <a href="#">Details</a> <a href="#">Other Downloads</a>	 Windows 32 Bit Windows 64 Bit
	<b>Eclipse IDE for C/C++ Developers</b> , 88 MB Downloaded 440,289 Times <a href="#">Details</a>	 Windows 32 Bit Windows 64 Bit
	<b>Eclipse for PHP Developers</b> , 141 MB Downloaded 240,131 Times <a href="#">Details</a>	 Windows 32 Bit Windows 64 Bit
	<b>Eclipse IDE for JavaScript Web Developers</b> , 108 MB Downloaded 88,526 Times <a href="#">Details</a>	 Windows 32 Bit Windows 64 Bit
	<b>Eclipse Modeling Tools (includes Incubating components)</b> , 249 MB Downloaded 65,588 Times <a href="#">Details</a>	 Windows 32 Bit Windows 64 Bit
	<b>Pulsar for Mobile Developers</b> , 122 MB Downloaded 57,656 Times <a href="#">Details</a>	 Windows 32 Bit Windows 64 Bit
	<b>Eclipse IDE for Java and Report Developers</b> , 241 MB Downloaded 56,499 Times <a href="#">Details</a>	 Windows 32 Bit Windows 64 Bit

# 实验设置及评判机制

- 随堂LAB： 8个
- 迭代实验（两次迭代）： 1个
- 学期大作业（Team Work）

- 3~4个人组成一个学习小组
  - 项目经理（组长）
  - 软件工程师
- 日常学习中相互切磋，解决问题，共同进步
- 运用面向对象的设计思路，协作完成一个功能相对完整的小型软件

- 软件学院云平台（请使用VPN登录）
  - 登录网址：<https://cloud-beihangsoft-cn.e1.buaa.edu.cn/>
  - 或：<https://cloud-beihangsoft-cn.e2.buaa.edu.cn/>
- 实验：实验除了有代码之外，必须**提交实验报告（采用下发的模板编写）**，最后我们都要存档
  - 请大家关注**deadline**
  - **杜绝拷贝和抄袭（一旦发现，零分处理）**

# Grading

1. 考勤+自学（10%）（实验课每次签到）
2. 实验（和课堂知识点相结合LAB 8个）（16%）
3. 迭代项目（指定题目，分两次下发）（14%）
4. 学期大作业（自拟题目、创新）（20%）
  - a) 注意：学生不能选择和小学期相同的项目为题，或者简单变更原有题目后提交作为学期大作业者都会重重扣分。
5. 期末考试（机考，40%）



- 理论是基础
- 实践是检验真理的唯一标准（正确的时间做正确的事）
  - 做好近期和远期规划（内驱力）
  - 勤奋，奋斗不息（情商、态度）
  - 多写，多练（行动）

- QQ大班群：
  - 资源下载和公告发布
  - 姓名\_学号

- 微信群：
  - 日常沟通



群聊：2025 秋季 OOP 课程群



该二维码7天内(9月17日前)有效，重新进入将更新



25年秋季OOP课程群

群号: 696614991 [复制](#)

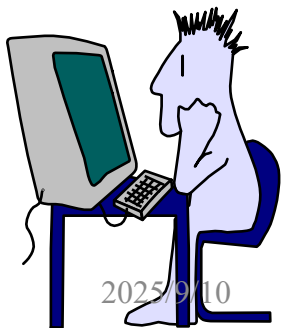


# Lesson1

## 认识对象：面向对象的编程基础

主讲老师：申雪萍

[shenxueping@buaa.edu.cn](mailto:shenxueping@buaa.edu.cn)



2025/9/10

Xueping Shen



北京航空航天大学  
COLLEGE OF SOFTWARE  
BEIHANG UNIVERSITY 软件学院

# 主要内容

- 程序设计语言现状
- 按面向对象开发的范例
- 面向过程设计思想
- 面向对象设计思想
- 认识Java

# 程序设计语言现状



2025/9/10

Xueping Shen



北京航空航天大学  
COLLEGE OF SOFTWARE  
BEIHANG UNIVERSITY 软件学院

# 编程语言的几个发展阶段

- 面向机器语言
- 面向过程语言
- 面向对象语言
- ;
- 还有一种脚本语言

# 机器语言—low level language

- 使用高低电压来表示0和1两种状态，计算机使用二进制数来存储和处理信息
- 机器语言直接由计算机的指令组成，可以被计算机直接执行



- 计算 $1+1$ ?

```
10111000
01000001
00000001
00000101
01000001
00000001
```

- **汇编语言**

- 使用一些简单的容易记忆的符号来代替二进制指令，比机器语言更容易读懂，更容易编写
- 汇编语言抽象层次很低，程序员需要考虑大量的机器细节。

机器语言与汇编语言都是面向机器的语言，需要针对不同的硬件环境编写不同的代码，因此这两种语言被称为低级语言

# 汇编语言

**MOV AX, 1**

**ADD AX, 1**

**10111000**

**01000001**

**00000001**

**00000101**

**01000001**

**00000001**

# 过程语言—high level language

- 随着计算机硬件功能的提高，在20世纪60年代出现了**面向过程的程序设计语言**，如**C语言和Fortran语言**
- **面向过程的语言**比低级语言更加接近人类的自然语言，因此**被称为高级语言**
- 使用面向过程的程序设计语言，不用再考虑机器指令的细节，只需要按照语言的语法规则去编写代码

```
int max(int a, int b)
{
    return a>b?a:b;
}
```

# 过程语言—high level language

- 面向过程的程序由**若干个过程或函数模块组成，每个模块的目标就是完成某一个任务。**

# 面向过程（C语言）

```
#include <stdio.h>
#include <stdlib.h>
void swap(int *x, int *y)
{
    int temp;
    temp = *x;
    *x = *y;
    *y = temp;
}
int main()
{
    int a = 10, b = 20;
    printf("交换前: \n a = %d, b = %d\n", a, b);
    swap(&a, &b);
    printf("交换后: \n a = %d, b = %d", a, b);
    return 0;
}
```

简单数据类型

复杂数据类型

```
typedef struct    //公历日期和时间结构体
{
    vu16  year;

    vu8   month;

    vu8   date;

    vu8   hour;

    vu8   min;

    vu8   sec;

}_calendar_obj;

_calendar_obj calendar; //定义结构体变量
```

# 面向对象编程—high level language

- 随着计算机硬件设备功能的进一步提高，使得面向对象的编程成为可能。
- 面向对象更加符合人的思维模式，编写的程序更加健壮和强大，也能够解决更加复杂的问题，并且面向对象编程鼓励创造性的程序设计。
- 面向对象的基本思想是，从现实世界中客观存在的事物出发来构造软件系统，并在系统的构造中尽可能运用人类的自然思维方式。



# 面向对象（Java）（Grandma类）

类

```
/** Grandma.java */
public class Grandma{
    private String name; //名字属性
    private int age=80; //年龄属性
    private static final String
sex="female";
    /** 不带参数的构造方法 */
    public Grandma(){
        name="未知";
    }
    /** 带参数的构造方法 */
    public Grandma(String name,int
age){
        this.name=name;
        this.age=age;
    }
}
```

```
/** 模拟吃饭的行为 */
public void eat(String food){
    System.out.println(food+"真好吃！");
}
/** 模拟做算术的行为 */
public void add(int a,int b){
    int sum=a+b;
    System.out.println("运算： "
        +a+"+"+b+"="+sum);
}
/** 显示奶奶的信息*/
public void printInfo(){
    System.out.println("姓名:"+name
        +"年龄:"+age);
}
}
```

# 一般程序执行

```
a = 4  
b = 3  
c = 2  
d = a*b+c
```



```
load 0 a  
load 1 b  
mult 0 1  
load 1 c  
add 0 1  
save 0 d
```



```
00000001000000001000  
00000001000100001010  
00000101000000000001  
00000001000100001100  
00000100000000000001  
000000100000000001110
```



**编写、编译、上载、运行**



# 一般程序执行

- 因为计算机只能识别0、1代码，因此使用高级语言编写的程序**需要通过某种机制将其转变成机器码**，计算机才能识别；
- 将高级语言编写的源文件转化成机器码通常有两种方式：**编译方式和解释方式**；

- (1) 编译方式 (Compilation)

- 针对当前的处理器，将源文件全部翻译成机器指令，称为目标程序，再将目标程序交给计算机执行

- (2) 解释方式 (Interpretation)

- 此种方式并不产生整个的目标程序，而是根据当前的处理器，边解释边执行，解释一句，执行一句

# 一般程序执行

- 无论哪种高级语言，都必须提供相应的**编译器或者解释器**
- Java语言编写的程序使用的是**编译与解释相结合**的方式来运行的
- JVM中提供的JIT（即时编译方式）将字节码直接转化成高性能的本地机器码，即JIT使得Java程序既能跨平台又能高速运行

# 程序设计语言的多样化特点

- 计算机语言呈现出多样化的特点：
  - 面向过程：C、Fortran等
  - 面向对象：C++、Java、C#等
  - 函数式编程语言
    - Haskell、Scala
  - 脚本语言
    - JavaScript、Python、Ruby、PHP、Asp .Net、Perl

各大编程语言特性一览

语言	优点	缺点	适用场景
Java	跨平台 面向对象 生态丰富	启动慢 内存占用大 语法冗长	企业级应用 Android开发 大型系统
C	高性能 底层控制 高移植性	手动内存管理 缺少现代特性 开发周期长	操作系统 嵌入式系统 性能关键应用
Python	简洁易读 快速开发 丰富的库	执行速度慢 GIL限制并发 动态类型不安全	数据分析 AI/机器学习 Web开发
C++	高性能 面向对象 硬件控制能力强	复杂难学 内存管理困难 错误难调试	游戏开发 系统软件 实时系统
JavaScript	前端必备 生态系统庞大 异步处理优秀	类型系统弱 历史包袱重 兼容性问题	Web前端 Node.js后端 桌面/移动应用
R	统计分析优秀 可视化能力强 专业数据处理库	性能较差 语法不一致 内存管理不佳	统计分析 数据可视化 学术研究
PHP	Web开发简便 部署简单 学习曲线平缓	安全问题 代码质量参差不齐 一致性欠佳	Web开发 CMS系统 简单Web应用
Go	并发支持优秀 编译速度快 语法简洁	缺少泛型(旧版本) 错误处理冗长 生态系统年轻	云服务 微服务 并发系统
Rust	内存安全无GC 性能高 并发安全	学习曲线陡峭 编译时间长 生态发展中	系统编程 WebAssembly 替代C/C++
SQL	数据查询标准 声明式语言 数据完整性保障	非完整编程语言 跨数据库差异 处理层次数据困难	数据库操作 数据分析 报表生成
C#	强类型现代 语言特性 .NET框架支持	主要限Windows平台 启动速度慢 框架体积大	Windows应用 游戏开发(Unity) 企业级应用

@知晓栈

@知晓栈



# 未来展望

- 计算机语言的发展呈现以下几个趋势：
  - 继续向更高层次的抽象化方向发展，以简化编程过程和提高开发效率；
  - **加强与自然语言处理技术的融合，实现更加自然的人机交互方式；**
  - 推动跨平台、跨语言的互操作性发展，以适应日益复杂的软件系统需求；
  - 注重安全性和可靠性设计，确保软件系统的稳定运行和数据安全。

# 特色化示范性软件学院建设与发展

- 实现**关键软件技术突破**，构建领域软件产业生态；
- 培养**特色化的软件人才**，构建领域软件人才生态；
- 关键软件五大领域分别是：
  - **关键基础软件**
  - 大型工业软件
  - 行业应用软件
  - 新型平台软件
  - 嵌入式软件

# 选择语言？



这么多语言，我该学哪个呢？

# 选择哪种语言？

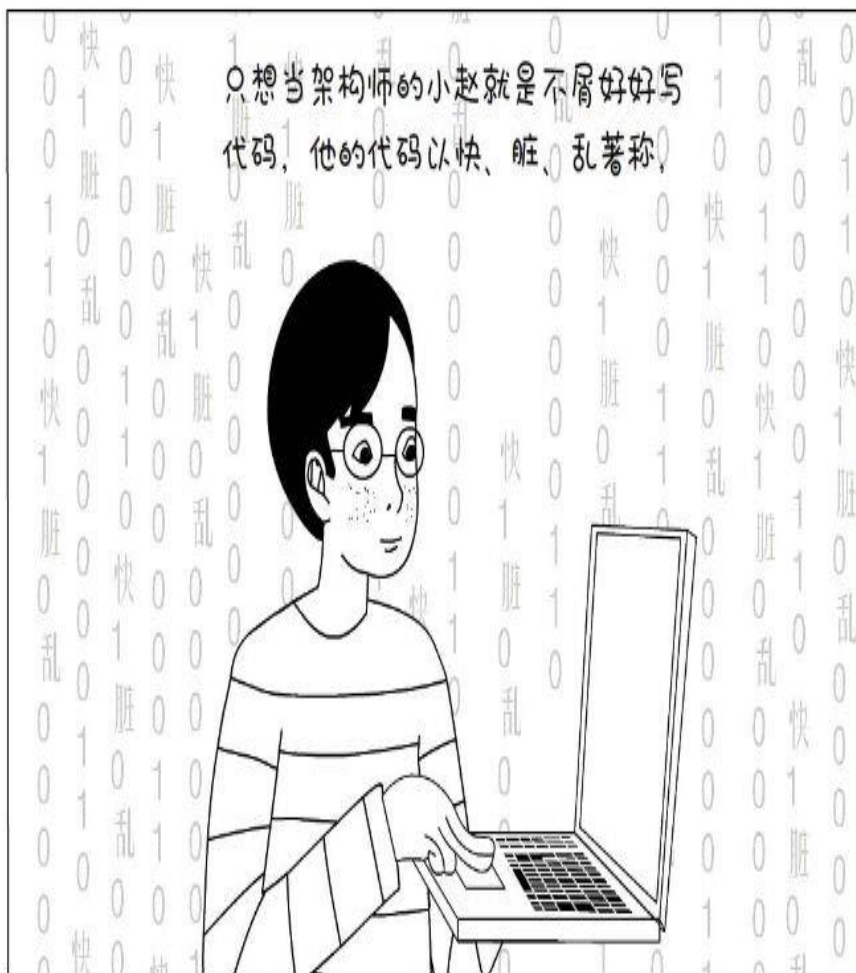
- C/C++、Java、.Net(C#)以及其它语言
  - **C/C++**：程序性能高、**支持底层应用**
    - 主要用于系统级软件、资源受限环境软件
    - 典型应用：通用软件、主机游戏、与硬件相关底层应用等
  - **Java**：跨平台，更好的互操作性
    - 主要用于大规模**企业级应用软件**，随着硬件的发展，其跨平台的特性使得其在嵌入式领域(如手机软件)应用也日益广泛
    - 典型应用：电信、银行等行业管理信息系统、Android应用、人工智能、大数据分析
  - .Net：开发、部署效率高，成本低
    - 主要用于中小规模企业级应用软件
    - 典型应用：部门级信息系统、桌面应用软件
  - ios平台上的语言：Objective-C、Swift
  - 其它脚本语言也有其应用环境，如PHP、**Python**、JS等
- **建议**
  - **至少精通一种语言，深入理解该语言的各种特性**
  - **进入科研、工作场景，用什么学什么！**



# 送给准备从事软件行业的您

- 软件技术更新非常迅速
  - 学习过程中不应拘泥特定的环境或工具，更要**掌握基本的原理和思想**
  - **重要的是培养学习能力、迅速掌握新技术的能力**
  - 及时关注技术更新，但又不盲目追崇新技术
- 信奉
  - (在软件行业，)没有不会的，只有不去学的！
- 忠告
  - 大学期间，至少精通一门语言，并了解各种流行语言

# 希望大家厚积薄发，成为业界追逐的人才



# 主要内容

- 计算机语言与程序
- 程序设计语言现状
- 按面向对象开发的范例
- 面向过程设计思想
- 面向对象设计思想
- 认识Java

# 按面向对象开发的范例



2025/9/10

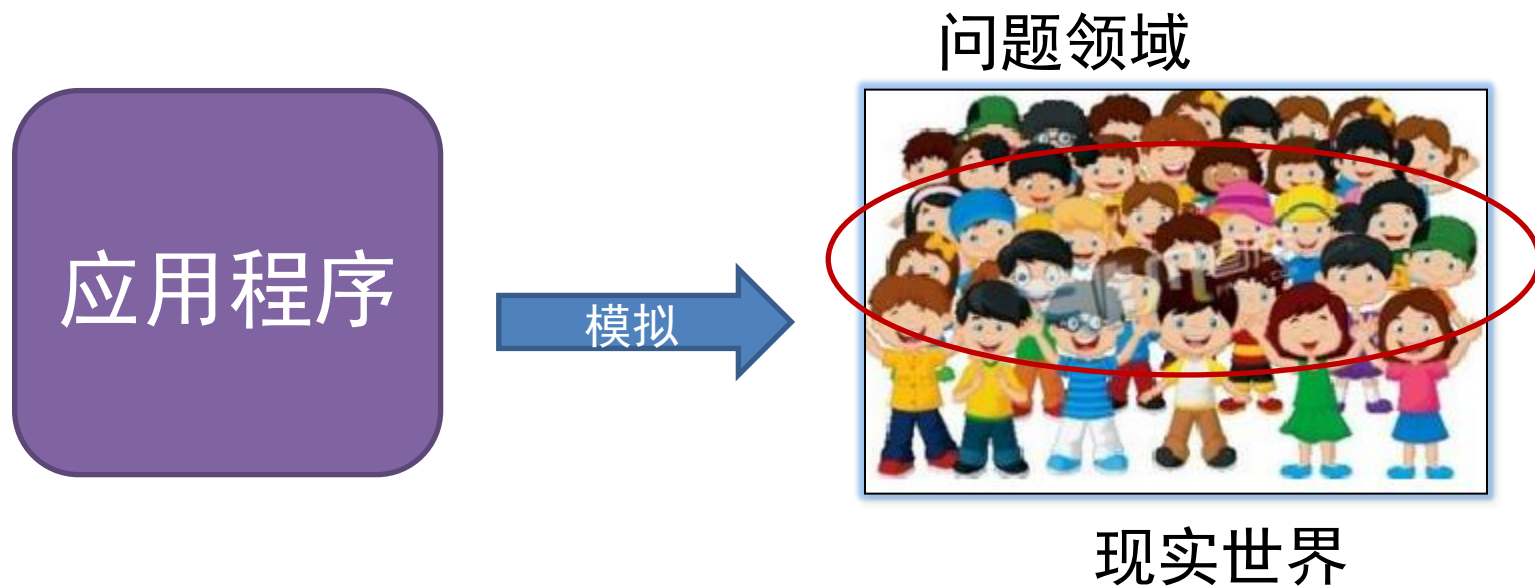
Xueping Shen



北京航空航天大学  
COLLEGE OF SOFTWARE  
BEIHANG UNIVERSITY 软件学院



# 应用程序模拟现实世界



# 问题领域：敬老院的奶奶们(对象们--objects)



101岁的会做算术的王奶奶



88岁的会做算术的张奶奶



92岁的会做算术的孙奶奶

# 对现实世界建立对象模型

类是对象共性的描述，  
包括属性和行为

奶奶类（名字，年龄，  
吃饭，做算术）

**Grandma类**

name属性

age属性

eat()方法

add()方法

实例化

抽象

**Grandma对象**

王奶奶  
101岁

会吃饭，做算术

张奶奶  
88岁

会吃饭，做算术

孙奶奶  
92岁

会吃饭，做算术

# 定义Grandma类

```
/** Grandma.java */
public class Grandma{
    private String name; //名字属性
    private int age=-1; //年龄属性
    /** 不带参数的构造方法 */
    public Grandma(){
        name="未知";
    }
    /** 带参数的构造方法 */
    public Grandma(String name,int
age){
        this.name=name;
        this.age=age;
    }
}
```

```
/** 模拟吃饭的行为 */
public void eat(String food){
    System.out.println(food+"真好吃！");
}
/** 模拟做算术的行为 */
public void add(int a,int b){
    int sum=a+b;
    System.out.println("运算： "
        +a+"+"+b+"="+sum);
}
/** 显示奶奶的信息*/
public void printInfo(){
    System.out.println("姓名:"+name
        +"年龄:"+age);
}
}
```

# 用new语句创建Grandma对象—实例化的过程

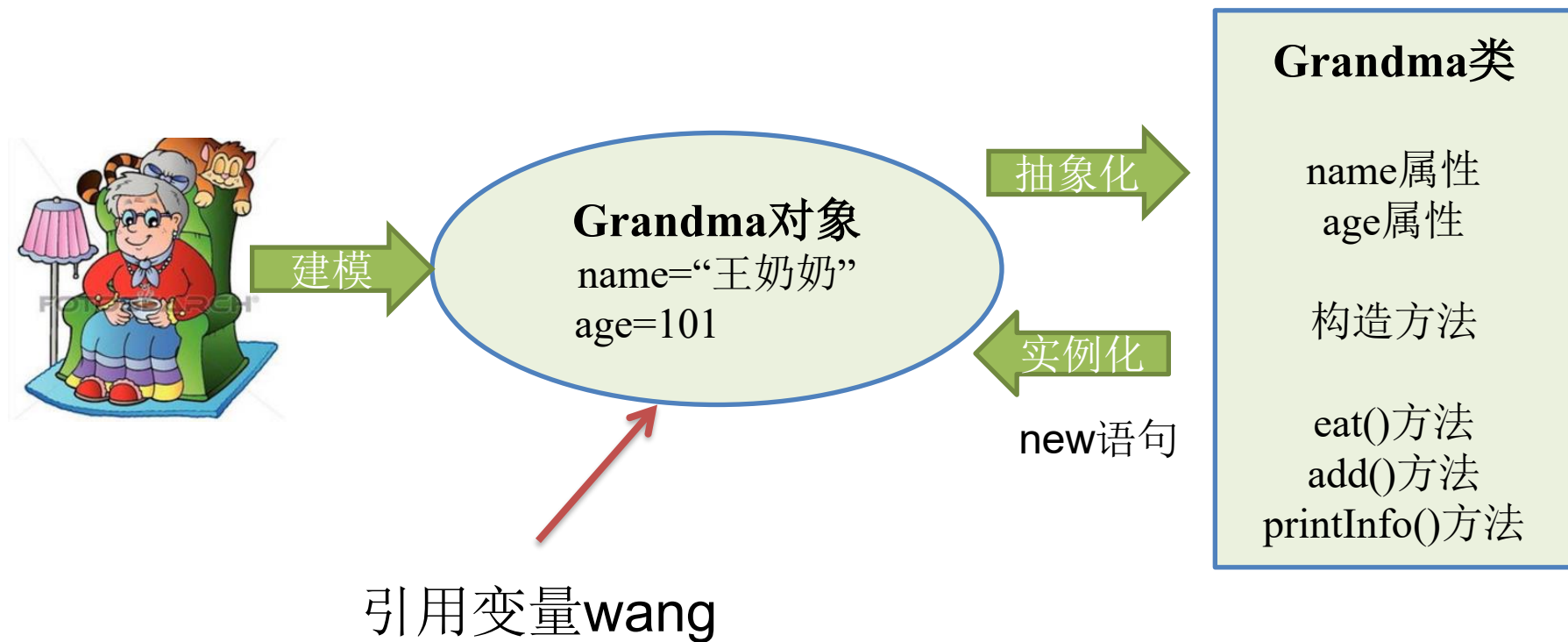
```
Grandma wang=new Grandma("王奶奶",101);  
Grandma zhang=new Grandma("张奶奶",88);  
Grandma sun=new Grandma("孙奶奶",92);
```



## 定义GrandmaTester类

```
/** GrandmaTester.java */  
public class GrandmaTester{  
    public static void main(String args[]){  
        Grandma wang=new Grandma("王奶奶",101);  
        wang.printInfo();  
        wang.eat("豆浆");  
        wang.add(7,8);  
        Grandma zhang=new Grandma("张奶奶",88);  
        zhang.printInfo();  
        zhang.eat("苹果");  
        zhang.add(90,98);  
    }  
}
```

# 小结





# 类与对象的关系





# 抽象和实例化

- 抽象：设计制作模具的过程（类的构建）
- 实例化对象/创建对象：是使用模具制造饼干的过程

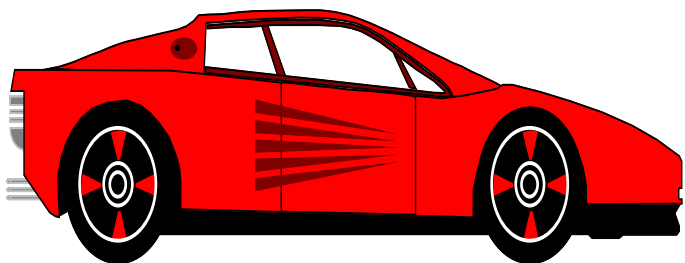


# 类是一种新的数据类型

- 制作完模具，我们就可以量产了，模具可以重复使用



## 范例二：现实生活中的小汽车

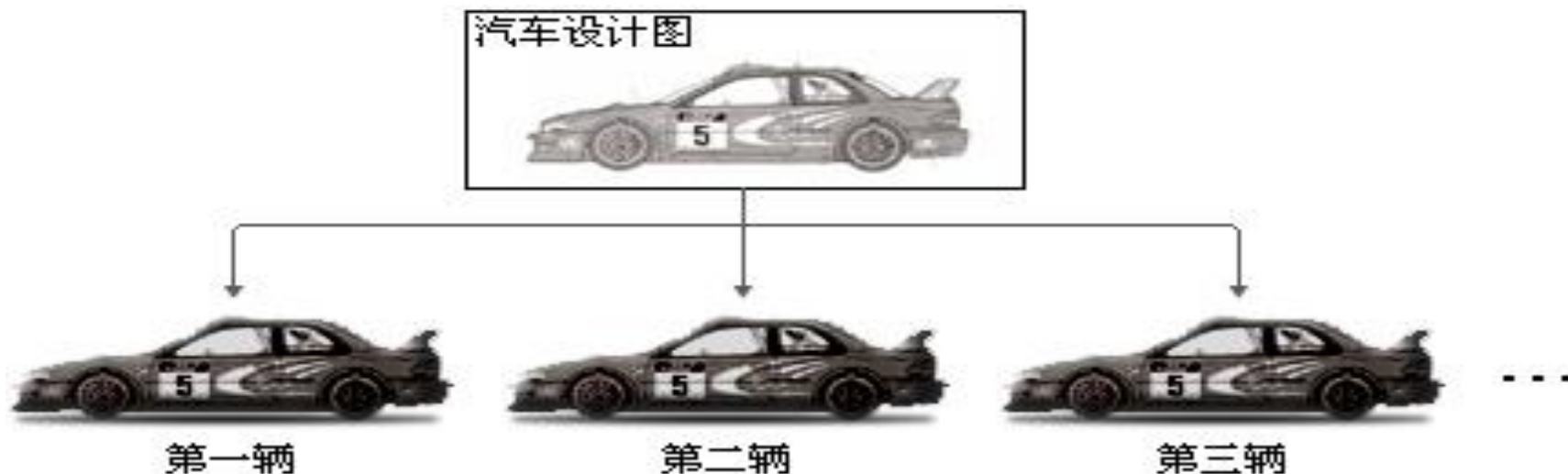


现实生活中的  
对象

计算机中  
的对象的**原型**

```
class Car {  
    String color;  
    String brand;  
    String model;  
    int door_number;  
    int speed;  
  
    void brake() { ... }  
    void speedUp() {...};  
    void slowDown() { ... }  
}
```

# 面向对象程序设计



如果将对象比作汽车，那么类就是汽车的设计图纸。  
**所以面向对象程序设计的重点是类的设计，而不是对象的设计。**

# 主要内容

- 计算机语言与程序
- 程序设计语言现状
- 按面向对象开发的范例
- 面向过程设计思想
- 面向对象设计思想
- 认识Java

# 面向对象与面向过程设计思想



2025/9/10

Xueping Shen



北京航空航天大学  
COLLEGE OF SOFTWARE  
BEIHANG UNIVERSITY 软件学院

# 制作一桌丰盛的年夜饭：两种不同的方案哦

- 第一种方案：
  - 做什么？怎么做？准备材料，按步骤由自己做
  - 嗯，累死。。



## 烤羊肉串的做法

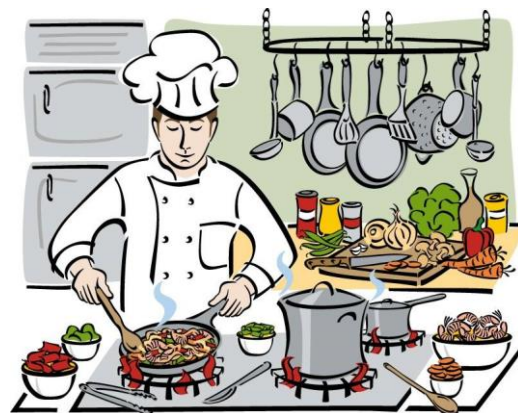
- 1 羊腿肉洗干净,切成大小均匀的小块
- 2 放在1个大碗中,加入生抽,胡椒粉,油,盐淹(最好淹1晚上,能充分入味)
- 3 把淹好的羊肉串在竹签子上
- 4 依次放在烤网上,底下垫烤盘上铺锡纸
- 5 烤箱220度预热10分钟烤15~20分钟
- 6 拿出撒上孜然粒和辣椒粉即可~



# 制作一桌丰盛的年夜饭：两种不同的方案哦

- 第二种方案：
  - 采购员（姑姑）买肉
  - 采购员（妈妈）买菜
  - 采购员（你）买鸡蛋和作料
  - 厨师（爸爸）烹饪
  - 厨师（婶婶）烤甜品

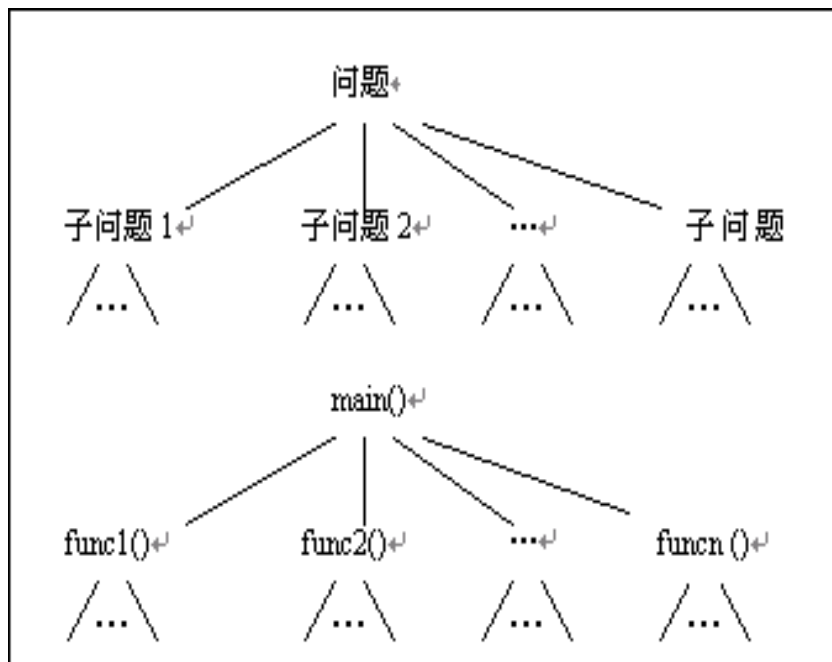
哈，对于大型软件系统，  
第二种方案似乎更合理





- 面向过程的思维：
  - 确定第一步先做什么，第二步做什么，第三步做什么。。。
  - 问题的解决方法：**自顶向下的功能分解法**
- 面向过程程序设计
  - 程序由多个**模块**构成
    - 程序组成单位：**过程或函数**

# 面向过程程序设计



**存在问题：**把数据和处理数据的过程分离。可复用性差、开发和维护都很困难。

- ①要解决的问题被分解成若干个子问题，每个子问题又被划分成若干个子子问题。
- ②这种自顶向下的功能分解一直持续下去，直到子问题足够简单，可以在相应的子过程中解决

# 面向过程与面向对象对比 (对复杂数据的封装)

- **面向过程 (C)**

- 在一个结构体中定义窗口的大小, 位置, 颜色, 背景等属性, 对窗口操作的函数与窗口本身的定义没有任何关系, 如HideWindow, MoveWindow, MinimizeWindow, 这些函数都需要接受一个代表要被操作的窗口参数, 是一种**谓语句与宾语**的关系。

- **面向对象 (Java and C++) (Window.java)**

- 定义窗口时, **除了要指定在面向过程中规定的那些属性**, 如大小, 位置, 颜色, 背景等外, **还要指定该窗口可能具有的动作**, 如隐藏, 移动, 最小化等。这些函数被调用时, 都是以某个窗口要隐藏, 某个窗口要移动的语法格式来使用的, 这是一种**主语与谓语句**的关系。

# Window.java（对象和函数之间是**主语与谓语**的关系）

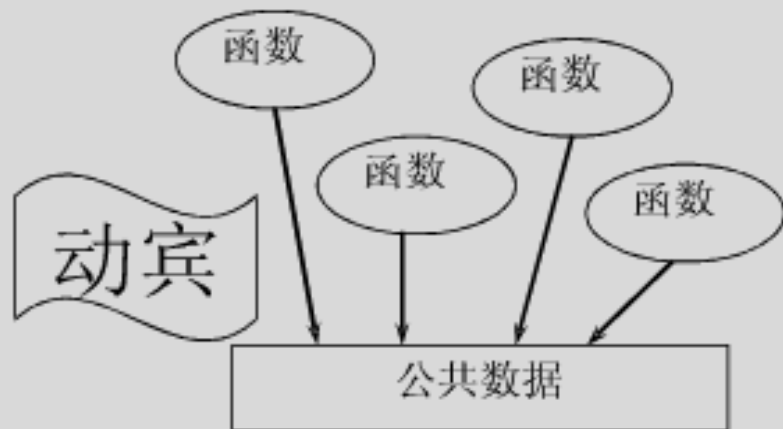
```
public class Window {  
    //定义窗口属性：大小，位置，颜色，背景  
    private int size;  
    private int x;  
    private int y;  
    private String color;  
    private String background;  
    //定义窗口方法  
    public void windowMove(){  
        System.out.println("moving");  
    }  
    public void windowHide(){  
        System.out.println("hideing");  
    }  
    public void windowMinimization(){  
        System.out.println("Minimizing");  
    }  
}
```

```
public static void main(String[] args) {  
    // TODO Auto-generated method stub  
    Window andrewWindow=new Window();  
    Window macWindow=new Window();  
    andrewWindow.windowMove();  
    andrewWindow.windowMinimization();  
    andrewWindow.windowHide();  
    macWindow.windowMove();  
    macWindow.windowMinimization();  
    macWindow.windowHide();  
}
```

Java中所有数据和  
函数都隶属于类

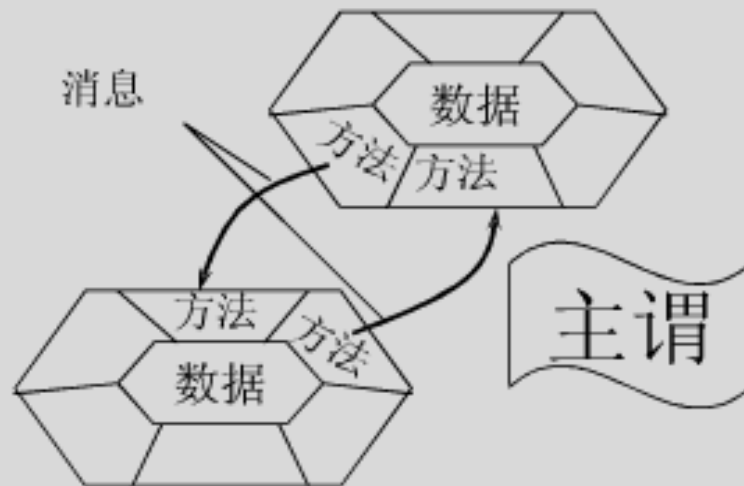
# 面向过程 VS 面向对象

## 面向过程程序设计



算法 + 数据结构

## 面向对象程序设计



对象 + 消息

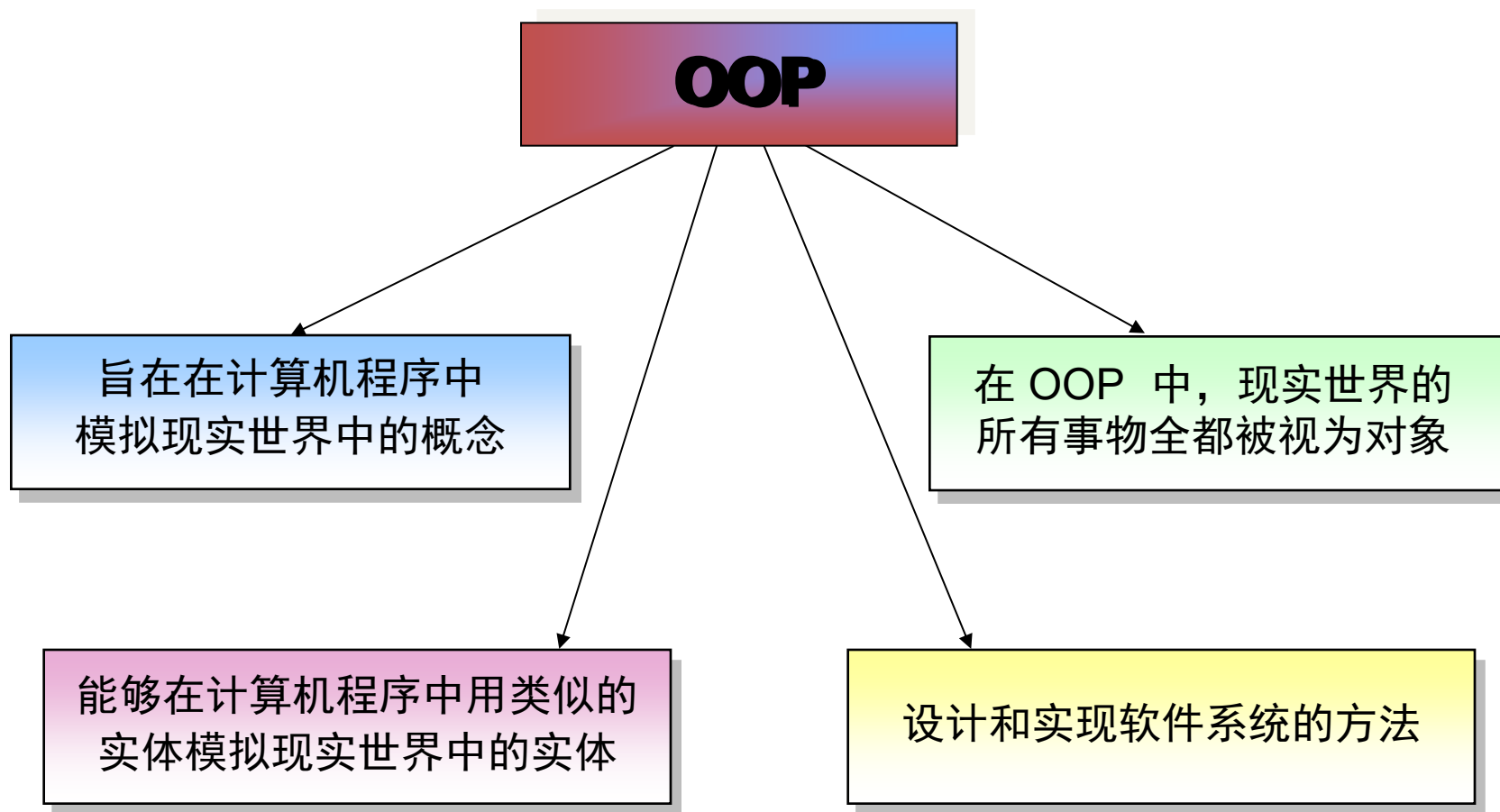
# C结构体，C++结构体 和 C++类的区别及联系

- 区别：
  - C的结构体不能添加成员函数，不能继承和派生；
  - C++的结构体和类，都能添加成员函数，能够继承和派生，具有多态性；
  - C++的结构体和类的区别仅仅是默认情况下外部对类成员的访问权限的不同，结构体默认是public，C++中类默认是private；
- 联系：
  - **结构体也可以被认为是一种特殊的类，它不存在任何函数，构造和析构函数也没有，而且是一个公共的类**

# C语言、C++和Java对比：

- C是面向过程的语言。C++和Java都是面向对象的。在C中没有类或者对象的概念，C中对包含复杂属性的主体采用struct封装；
- Java和C++都是面向对象语言，C++兼容C, 而且对C中的结构体做了扩展，但是Java中彻底抛弃了struct的概念；
- Java和C++程序的共同点都需要抽象出类，都具有封装、继承和多态的三大特征；
- C和C++主要用在在中间件和底层，Java主要用在应用层，Java更侧重于面向对象的思想（为了完全面向对象，可以适当的牺牲效率），而C++更侧重于效率（为了更好的效率，可以适当的牺牲面向对象），所以C++相比Java在语法上更复杂一些。

# 面向对象的设计思想





- 设计应用程序，模拟现实世界中，**动物园里饲养员喂养动物**的场景

- Animal类
- Animal子类Dog类
- Animal子类Cat类
- 食物Food类
- Food类子类Bone
- Food类子类**Fish**
- 饲养员Feeder类
- 测试类

**Fish的特殊性在哪里？**

- ① 第一步：**明确问题域，分析这个问题里面有哪些类和对象**
- ② 第二步：**分析这些类和对象应该具有哪些属性和方法。**
- ③ 第三步：**分析类和类之间具体有什么关系。**

# 面向对象的设计思想

- 人在思考的时候，首先眼睛里看到的是一个一个的对象。
- 面向对象的基本思想是：从现实世界中客观存在的事物出发来构造软件系统，并在系统的构造中尽可能**运用人类的自然思维方式**。
- 面向对象更加强调运用人类在日常生活的逻辑思维中经常采用的思想方法与原则，如**抽象、分类，继承、聚合、多态**等。

# 面向对象设计思想

- 面向对象其实是现实世界模型的自然延伸。现实世界中任何实体都可以看作是对象,都归结为某一类事物,都是某一类事件的实例。万物皆对象
- 程序由类组成: 对相同类型的对象进行分类、抽象后,得出共同的特性而形成了类。例如Student, Teacher和Person类。
- 将数据及对数据的操作行为放在一起,作为一个相互依存、不可分割的整体——对象。对象是细粒度的。
- 对象之间通过消息(方法)相互作用,完成系统功能。

# 面向过程的想法和面向对象的想法

- 面向对象和面向过程的想法有着本质上的区别
- 面向过程的思维：
  - 确定第一步先做什么，第二步做什么，第三步做什么。。。 自顶向下的功能分解法
- 面向对象的思维：
  - 第一步：明确问题域，分析这个问题里面有哪些类和对象
  - 第二步：分析这些类和对象应该具有哪些属性和方法。
  - 第三步：分析类和类之间具体有什么关系。

# 面向过程的想法和面向对象的想法

- “面向过程”是一种**以过程为中心**的编程思想，强调的是功能行为。
- “面向过程”只是**分析出解决问题所需要的步骤**，然后用函数把这些步骤一步一步实现，使用的时候一个一个依次调用就可以了。面向过程在这一系列工作的执行中，强调的是工作的执行。
- “面向过程”不支持丰富的“面向对象”特性（比如继承、多态），

# 面向对象编程

- 面向对象更加重视软件的可维护，可扩展，可复用等特性。
- 面向对象编程的三个特性
  - 封装
    - 面向对象编程的核心思想就是将数据和数据的操作封装在一起，通过抽象，即从具体的实例中抽取共同的性质形成一般的概念，比如类的概念
  - 继承
  - 多态
    - 一种是操作名称的多态，即方法的重载。
    - 另一种是指同一个操作被不同对象调用时可能产生不同的行为。

# 主要内容

- 计算机语言与程序
- 程序设计语言现状
- 按面向对象开发的范例
- 面向过程设计思想
- 面向对象设计思想
- 认识Java