

《面向对象程序设计（Java）》课程教学大纲

课程编号：B3I212240

课程中文名称：面向对象程序设计（Java）

课程英文名称：Object-Oriented Programming (Java)

开课学期：春季

学分/学时：3/32+32

先修课程：高级语言程序设计、算法分析与设计

建议后续课程：Java EE 架构、软件系统分析与设计

适用专业/开课对象：软件工程专业，二年级本科生

团队负责人：申雪萍

责任教授：

执笔人：申雪萍

核准院长：张莉

一、课程的性质、培养目标

面向对象程序设计(Java)是面向软件工程专业二年级本科生的专业基础课程，其目标是向学生系统地介绍面向对象的程序设计方法和思想。该课程借助 Java 作为教学语言，讲授面向对象编程的基础理论和基本思想，培养学生的面向对象思维。主要包括类和对象、封装、继承、多态等面向对象的编程思想，以及 Java 异常处理、用户界面程序设计、文件与 Java I/O、多线程程序设计、网络程序设计的编程思想。

该课程的特点是既有理论基础，又需要大量实践。因此在教学中应理论联系实际，加强实践环节的教学。在教学方法上应尽量采用基于问题的方法、案例式方法、讨论式方法等多种教学方法。通过具体的案例分析和实践训练，培养学生掌握高质量的编程方法，具备处理实际问题的思维方法与基本能力，为进一步学习其它面向对象语言以及 Java 高级编程打下坚实的基础。

通过本课程的学习，学生应该达到以下目标：

- 1) 理解并掌握面向对象程序设计的基本思想、原理和方法。
- 2) 掌握使用 Java 语言进行面向对象程序设计的方法，培养学生初步具备模块化思维能力。
- 3) 培养学生应用 Java 解决和处理实际问题的思维方法和基本能力，能够利用 Java 编写面向对象的程序，构建面向对象的业务系统。
- 4) 培养以团队的方式开展需求、设计和编码等各阶段实践的能力。
- 5) 掌握软件开发过程中的各类建模、开发工具的使用。

本课程的教学目标和毕业要求指标点的对应关系如下表：

课程目标	支撑的培养方案中的毕业要求观测点
------	------------------

目标 1	2.3 掌握软件系统需求分析、设计、验证、实现、确认、应用和维护所需的基础知识，并用于解决复杂软件工程问题；
目标 2	4.1 能针对特定需求设计和实现软件模块，测试验证模块的正确性，并进行性能优化；
目标 3	4.2 能针对特定需求设计并实现功能完整的软件系统，包括软件系统结构设计、各模块及其交互协议实现或选择、正确性验证、部署、运行和维护等；
目标 4	9.1 理解软件开发过程中团队所涉及的角色和不同角色所承担的职责，以及不同角色之间合作的重要性。能够在团队中独立或合作开展工作；
目标 5	6.1 能够使用常用软件工程相关工具，如建模/仿真工具、开发工具、测试工具、配置管理工具和项目管理工具等；更根据实际需要选择合理的工具；

二、 课程内容、基本要求及学时分配

第一章 认识对象：面向对象的编程基础（2 学时）

- 1.1 计算机语言与程序
- 1.2 程序设计语言现状
- 1.3 按面向对象开发的范例
- 1.4 面向过程设计思想
- 1.5 面向对象设计思想
- 1.6 了解 Java

主要支持教学目标1、5

重点：理解面向对象的基本思想

难点：对象和类的区别和联系

第二章 认识对象：封装数据为类（4 学时）

- 2.1 类的概念
- 2.2 对象的概念
- 2.3 类与对象的关系
- 2.4 如何理解抽象
- 2.5 如何理解消息 ？
- 2.6 面向对象的核心
- 2.7 使用面向对象思维实际问题（案例分析）
- 2.8 对象的组合与复用
- 2.9 类的设计原则
- 2.11 类的定义
- 2.12 对象的生成、使用和清除
- 2.13 匿名对象概念和使用
- 2.14 进一步理解引用类型变量
- 2.15 实例变量和实例方法
- 2.16 类变量和类方法

主要支持教学目标2、3

重点：类的抽象和封装、类的结构、对象的创建、使用和销毁、类变量和类方法

难点：类变量和类方法、匿名对象、对象的组合与复用

第三章 面向对象的三大特称之一：封装（2学时）

- 3.1 信息隐藏的必要性
- 3.2 私有成员（变量和方法）的理解和使用
- 3.3 共有成员的理解和使用
- 3.4 保护成员的理解和使用
- 3.5 使用不加任何权限修饰符的成员
- 3.6 封装的好处
- 3.7 深刻理解封装（从系统的角度）
- 3.8 案例分析（单实例、has a双向关联）

主要支持教学目标2、3

重点：在正确理解权限修饰符的作用的基础上掌握其使用

难点：protected关键字的理解、单实例、has a双向关联

第四章 面向对象的三大特征之二：继承（2学时）

- 4.1 继承的必要性
- 4.2 单继承和多继承
- 4.3 基类与派生类
- 4.4 super和this
- 4.5 方法重载和方法覆盖
- 4.6 深化理解private、protected和public
- 4.7 复合与继承的区别
- 4.8 继承的利弊和使用原则
- 4.9 案例分析

主要支持教学目标2、3

重点：extends关键字的使用，方法重载和方法覆盖

难点：深化理解private、protected和public

第五章 面向对象的三大特称之三：多态（4学时）

- 5.1 多态的必要性
- 5.2 静多态和动多态
- 5.3 方法重载，方法覆盖
- 5.4 抽象方法
- 5.5 抽象类
- 5.7 接口的必要性（将接口用作API）
- 5.8 定义接口
- 5.9 实现接口
- 5.10 将接口用作类型、接口回调（使用接口）
- 5.11 接口的进化（通过接口的继承完成）
- 5.12 面向接口的编程
- 5.13 案例分析

主要支持教学目标2、3

重点：在正确理解抽象方法、抽象类、接口的基础上，灵活使用

难点：接口的定义，使用和赋值

第六章 完善类的设计（2 学时）

- 6.1 最终方法、最终类
- 6.2 内部类

- 6.3 匿名类
- 6.4 对象向上映射和向下映射
- 6.5 复杂软件的管理（package和import的使用）
- 6.6 高内聚、低耦合
- 6.7 面向对象编程六大原则
- 6.9 案例分析

主要支持教学目标2、3、4：掌握利用Java构造面向对象程序的原理、方法和机制。

重点：final关键字的使用，复杂软件的管理，对象向上映射和向下映射

难点：内部类，匿名类，对象向上映射和向下映射

第七章 泛型与集合框架（2 学时）

- 7.1 泛型的必要性
- 7.2 泛型的使用
- 7.3 集合框架的设计思路：Collection 接口、迭代器(Iterator)
- 7.4 案例分析

主要支持教学目标3

重点：泛型、Collection接口、迭代器(Iterator)、线性表(List、ArrayList、LinkedList)、枚举类在企业级软件编写中经常用到，要求在理解的基础上，熟练掌握。

难点：泛型的概念及其使用

第八章 完善对象，增强软件的鲁棒性：Java 异常处理（2 学时）

- 8.1 异常概述
- 8.2 异常处理机制
- 8.3 使用 try...catch...finally 处理异常
- 8.4 声明抛出异常
- 8.5 人工抛出异常
- 8.5 创建用户自定义异常类
- 8.6 异常处理原则
- 8.7 案例分析

主要支持教学目标3

重点：Java异常处理机制，关键字try、catch、finally、throws和throw的灵活使用

难点：自定义异常类及其使用

第九章 用户界面程序设计（2 学时）

- 9.1 字体和颜色的灵活使用
- 9.2 组件和容器
- 9.3 布局管理器
- 9.4 事件处理
- 9.5 高级图形用户界面（菜单、表格和多文档界面）
- 9.6 匿名类和内部类

主要支持教学目标3、4

重点：组件和容器、事件管理

难点：事件处理，匿名类和内部类

第十章 文件与数据流（3 学时）

- 10.1 控制台读入数据、标准输入输出、Scanner 类（基础的输入输出）
- 10.2 Java 文件管理（File 类）
- 10.3 输入/输出流类及其衍生类的使用（字节流的顺序读写）
- 10.4 随机访问文件

10.5 读写器及其派生类的使用（字符流的顺序读写）

10.6 对象序列化（对象流的读写）

主要支持教学目标3、4

重点：File类、InputStream/outputStream及其派生类、Reader/Writer及其派生类

难点：对象序列化

第十一章 多线程程序设计（3 学时）

11.1 程序、进程和多线程概念及特点

11.2 线程的状态、生命周期及调度策略

11.3 线程类(Thred)和 Runnable 接口

11.4 案例分析

主要支持教学目标3、4

重点：Java线程的状态、生命周期及调度策略、线程类(Thred)和Runnable接口

难点：线程类(Thred)和Runnable接口的灵活使用，数据一致性的管理

第十二章 网络程序设计（3 学时）

12.1 网络编程基础（协议、封装、拆封、IP、端口号、URL）

12.2 Socket 的理解和使用

12.3 C/S 与 B/S 架构

12.4 基于 TCP 的网络程序设计

12.5 基于 UDP 的网络程序设计

12.6 案例分析（综合 Java I/O，Java 多线程和 Java 网络程序设计实现自由聊天程序）

主要支持教学目标3、4

重点：基于TCP的编程模式、基于UDP的编程模式

难点：Socket的理解和使用

三、 教学方法：

面向对象程序设计（Java）是是一门实践性很强的课程，该课程的特点是既有理论基础，又需要大量实践。因此在教学中应理论联系实际，加强实践环节的教学。在教学方法上应尽量采用基于问题的方法、案例式方法、讨论式方法等多种教学方法。

为了使學生顺利完成大纲中的教学目标，本课采用授课、上机实验和学期 Team Work 相结合的“教学+实践”方式，逐步培养学生的程序设计能力。

- 1) 理论与实践相结合。本课程实践性强，因此通过大量的实例、案例分析、小项目等加深对课程内容的理解和掌握。PPT 中将展示大量案例分析，帮助学生理解编程思想，锻炼、启发、培养学生的实际动手能力。结合课程的知识点，对核心概念和基本原理，安排课下问答题，启发，帮助学生更好的掌握基本概念和基本原理。
- 2) 采用问题驱动法、任务驱动以及小组协作等教学方法。在实际教学和实验中尽可能采用多种教学方法，开拓学生思路，提高教学效率。例如可以先提出问题，根据问题寻找解决方案，从而启迪学生的思维，提高教学效果。
- 3) 课内实践和课外实践相结合。教师指定有关题目，要求学生在课内和课外完成。为了使學生能够更好的理解课程的知识点，上机实验内容力求做到紧密和授课进度

关联，同时设计实验的难度，从小项目入手，逐步丰富和细化，最终以团队形式完成自拟题目的学期大作业，注重循序渐进培养学生的综合分析问题，解决问题的能力。

- 4) MOOC 课程和课堂教学相结合，一方面，本课程所选择的教材有教学视频，另一方面，为学生提供和 Java 相关的 MOOC 课程。

四、课内外教学环节及基本要求

面向对象程序设计（Java）这门课共有 64 学时的课程安排，其中授课 32 学时，实验 32 学时。32 学时上机实验，分 16 次进行，每次两小时。上机实验内容应结合日常授课的核心知识点，由授课老师下发给学生，学生每次实验需要编写实验报告提交 Java 源程序、测试集和测试结果。

除了有针对本课知识点的相关实验内容外，本课程还开发了图书馆管理系统，教务管理系统和点餐系统等大型综合实验项目，结合课程知识点下发给学生，逐步培养学生由简到繁，解决复杂问题的能力。上述两部分实验占总分数的 20%，从而加强学生对知识点的理解和掌握，培养学生程序设计能力和 Java 语言的编程能力。

学期大作业为 2~3 人组成的团队 project，为必选题目，学期大作业占总成绩的 20%。作为学生的大作业，学生需上交的内容包括源代码和文档两部分。学期项目以 2~3 人团队进行，主要目的是培养学生的自学能力、创新能力，锻炼学生综合运用所学知识，进入实际项目的能力，以团队进行也旨在加强大家的团队工作意识，培养学生的团队合作能力、领导能力和驾驭实际工作中出现困难，解决问题的能力。为将来学生走向工作岗位打下坚实的基础。

五、考核方式及成绩评定

本课程成绩由课堂及实验的参与度、实践和期末考试三部分组合而成，采用百分制计分制。各部分所占比例如下：

- 1) 考勤（10%）
- 2) 实验（和课堂知识点相结合、10 个）（10%）
- 3) 学期项目（指定题目）（10%）
- 4) 学期大作业（自拟题目）（20%）
- 5) 期末考试（50%）

六、持续改进

本课程根据学生实践、课堂讨论和学生评教、教学督导等反馈，及时对教学中不足之处进行改进，并在下一轮课程教学中改进提高，确保相应毕业要求观测点的达成。此外，课程案例设计也会结合当前软件工程学科的发展，不断进行更新，以反映最新的理论和方法。

七、教材和参考资料

教材：

- 【1】 Java 面向对象编程(第 2 版) 孙卫琴著，电子工业出版社，2017-01-01 出版

【2】 Java 2 实用教程(第 5 版) 耿祥义, 张跃平著, 清华大学出版社, 2017-04-01

参考资料:

【1】 Java 编程思想, Bruce Eckel 编著, 机械工业出版社, 2007-6-1 出版 (第四版)

【2】 Java 语言程序设计 (基础篇、进阶篇), Daniel Liang 编著, 机械工业出版社, 2016-8 出版 (第一版)

【3】 Java 语言导学(The Java Tutorial)(第 6 版), 雷蒙德·盖拉多(Raymond Gallardo)等编著, 机械工业出版社, 2017-07 出版

【4】 《程序开发原理: 抽象、规格与面向对象设计》, Barbara Liskov, John Guttag 著, 裘健译, 电子工业出版社, 2006 年 6 月, 第 1 版