

《软件工程》教学大纲

课程编码：112720

课程名称：软件工程

学时/学分：60/2.5

先修课程：《C语言》、《数据结构》、《面向对象程序设计》、《数据库原理》

适用专业：信息与计算科学

开课教研室：信息与计算科学教研室

一、课程性质与任务

1. 课程性质：《软件工程》是计算机科学与技术的一门专业必修课程。

2. 课程任务：通过学习，应使学生能熟练掌握生存周期与软件开发模式；掌握结构化程序设计的编程思想；掌握面向对象分析、设计与编码方法；掌握有关软件的评审、测试与维护，项目计划与项目管理方法；能用软件工程的方法参与软件项目的分析、设计、实现和维护应能掌握开发高质量软件的方法，以及有效地策划和管理软件开发活动，为学生参加大型软件开发项目打下坚实的理论基础。

二、课程教学基本要求

通过学习可以使学生掌握软件工程的基本概念、基本原理、实用的开发方法和技术；掌握软件生存周期各阶段的任务；了解软件工程各领域的发展动向；如何用工程化的方法开发软件项目，在开发过程中应遵循的流程、准则、标准和规范。同时，本课程也是一门实践性很强的课程，它是各种软件开发经验的总结与提炼，对于本课程的学习不但应注重概念、原理、方法和技术的掌握，也应注重方法和技术的应用，学习该课程为计算机科学领域培养高级软件开发人员和系统分析员打下坚实的基础。

成绩考核形式：末考成绩（闭卷考查）（70%）+平时成绩（平时测验、作业、课堂提问、课堂讨论等）（30%）。成绩评定采用百分制，60分为及格。

三、课程教学内容

第一章 软件工程学概述

1. 教学基本要求

掌握软件的特征、软件危机、软件工程中的五种软件开发模式，使学生理解软件工程的观念及产生软件危机的原因；软件生命周期中各阶段的基本任务。

2. 要求学生掌握的基本概念、理论、技能

通过本章教学使学生了解软件的特点，软件生产发展的三个阶段以及各阶段的特点，软件危机的产生及其表现形式；初步了解软件的生存周期模型；理解软件工程的定义及特点，

软件生存期各阶段的特点和内容；深刻理解软件危机产生的原因；掌握软件生命周期中各阶段的基本任务。

3. 教学重点和难点

教学重点是软件危机、软件工程产生的原因，软件工程过程，软件生命周期模型。教学难点是软件生命周期模型。

4. 教学内容

(1) 软件危机

主要知识点：软件危机：概念、典型表现形式、软件危机案例；产生软件危机的原因；消除软件危机的途径。

(2) 软件工程

主要知识点：软件工程概念和特性；软件工程基本原理；软件工程方法学。

(3) 软件生命周期

主要知识点：软件定义时期；软件开发时期；运行维护（软件维护）时期。

(4) 软件过程

主要知识点：软件过程和软件生命周期模型概念；瀑布模型；快速原型模型；增量模型；螺旋模型；四种模型比较；喷泉模型。

第二章 可行性研究

1. 教学基本要求

掌握软件的可行性分析、软件计划及成本估算，使学生明确可行性分析的基础和主要任务，掌握可行性研究的方法和步骤。

2. 要求学生掌握的基本概念、理论、技能

通过本章教学使学生深刻理解可行性研究的必要性，掌握可行性研究的任务及可行性研究的具体步骤；了解系统流程图的作用及符号表示，重点掌握数据流图的画法及数据字典的编写。

3. 教学重点和难点

教学重点是可行性研究任务；数据流图基本符号、绘制过程及应用；数据字典的用途和建立。教学难点是数据流图的应用。

4. 教学内容

(1) 可行性研究的任务

主要知识点：可行性研究的目的；可行性研究的基本内容：经济、技术、操作。

(2) 可行性研究过程

主要知识点：过程；报告编写规范。

(3) 系统流程图

主要知识点：概念；符号及例子；分层。

(4) 数据流图

主要知识点：符号及例子；命名；用途。

(5) 数据字典

主要知识点：数据字典的内容；定义数据的方法；数据字典的用途；数据字典的实现。

(6) 成本、效益分析

主要知识点：成本估计；成本/效益分析的方法。

第三章 软件需求分析

1. 教学基本要求

掌握需求分析的任务和步骤、结构化分析方法及原型化分析方法，使学生掌握结构化分析方法的指导原则、数据流图、数据词典和加工说明等描述手段。了解需求分析文档建立和复审的方式。

2. 要求学生掌握的基本概念、理论、技能

通过本章教学使学生深刻理解需要分析阶段的概念及任务，熟练掌握数据流图的细化及ER图以及规范化方法（如范式），IPO图的画法；掌握常用图形工具的使用方法。

3. 教学重点和难点

教学重点是基本技术、需求规格说明书的作用与组成。教学难点：基本技术。

4. 教学内容

(1) 需求分析的任务与获取需求的方法

主要知识点：任务；面向数据流自顶向下求精；快速建立软件原型。

(2) 分析建模与规格说明

主要知识点：分析建模；软件需求规格说明；实体—联系图；状态转换图

(3) 图形工具

主要知识点：层次方框图；Warnier图；IPO图。

第四章 形式化说明技术

1. 教学基本要求

掌握形式化方法的概念、优点和应用准则；初步掌握有穷状态机、Petri网的概念和使用。

2. 要求学生掌握的基本概念、理论、技能

通过本章教学使学生了解非形式化方法的缺点，掌握形式化方法的概念、优点和应用准则；掌握有穷状态机、Petri网的特点以及相关应用。

3. 教学重点和难点

教学重点是形式化方法的概念、优点和应用准则；有穷状态机。教学难点是有穷状态机。

4. 教学内容

(1) 概述

主要知识点：非形式化方法的缺点；形式化方法的优点；应用形式化方法的准则

(2) 有穷状态机

主要知识点：概念、保险箱、电梯例子、评价

(3) Petri 网

主要知识点：概念、电梯例子

第五章 总体设计

1. 教学基本要求

掌握软件结构设计中结构化设计方法和面向数据流的设计方法，使学生明确结构设计的目标和任务，理解结构化设计的概念和原理，掌握结构化设计的步骤以及相关方法。

2. 要求学生掌握的基本概念、理论、技能

通过本章教学使学生了解总体设计的过程；深刻理解软件总体设计的基本任务，软件设计的基本原理。理解模块化，抽象，信息隐蔽，模块独立性等概念，明确度量模块独立性的校准—耦合性及内聚性。熟练掌握结构化设计方法（要求能划分数据流的类型，将其能换成软件结构图，并能根据优化准则将其优化。）

3. 教学重点和难点

教学重点是软件设计过程中应遵循的基本原理；面向数据流的设计方法。教学难点是变换分析、事务分析法的过程和应用。

4. 教学内容

(1) 设计过程

主要知识点：设计过程

(2) 设计原理

主要知识点：模块化、抽象、逐步求精、信息隐藏和局部化、模块独立。

(3) 启发规则

主要知识点：改进软件结构提高模块独立性；模块规模应该适中；深度、宽度、扇出和扇入都应适当；模块的作用域应该在控制域之内；力争降低模块接口的复杂程度；设计单入口单出口的模块；模块功能应该可以预测。

(4) 描绘软件结构的图形工具

主要知识点：层次图和 HIPO 图；结构图。

(5) 面向数据流的设计方法

主要知识点：变换流和事务流的概念；变换分析；事务分析；设计优化。

第六章 详细设计

1. 教学基本要求

掌握详细设计方法采用的结构化程序设计的思想和原理；常用的算法过程的描述工具；算法的复杂性评价问题。使学生明确详细设计阶段的主要任务，掌握算法复杂性的度量方法。

2. 要求学生掌握的基本概念、理论、技能

通过本章教学使学生能熟练的使用详细设计描述工具来设计模块中的算法及程序的逻辑结构。理解 Jackson 方法的概念及程序复杂度的度量方法，学会使用 Jackson 方法设计输入输出数据结构和程序结构。

3. 教学重点和难点

教学重点是流程图、PAD 图、过程设计语言的简单应用；过程设计的工具以及面向数据结构的设计方法，程序复杂程度的定量度量。教学难点是用 Jackson 方法设计小型题目。

4. 教学内容

(1) 结构程序设计

主要知识点：设计任务、概念、原则。

(2) 人机界面设计

主要知识点：界面设计概念、特性、设计问题；系统响应时间；设计过程；设计指南。

(3) 过程设计的工具

主要知识点：程序流程图、盒图、判定表、过程设计语言。

(4) 面向数据结构的设计方法

主要知识点：Jackson 图；改进的 Jackson 图；Jackson 方法。

(5) 程序复杂程度的定量度量

主要知识点：程序图；环形复杂度概念及计算方法。

第七章 实现

1. 教学基本要求

了解程序设计语言的概念以及编码应当遵循的原则；理解编码设计的目标与任务、编码使用的程序设计语言的特性以及程序设计风格的重要性；掌握软件测试的概念与主要任务，软件的排错技术和软件可靠性的评测方法。理解软件测试的目标、原则以及测试方案的建立和测试过程。掌握测试实例的构造方法。

2. 要求学生掌握的基本概念、理论、技能

通过本章教学了解几种常见的程序设计语言的特点，领会程序设计中应注意的问题，

注意培养良好的编程风格；掌握白盒，黑盒测试技术，能针对某一问题，熟练采用白盒法或黑盒法进行测试用例的设计。理解测试过程中单元测试，集成测试，验收测试的任务及使用的方法；掌握调试程序的方法

3. 教学重点和难点

教学重点是领会写程序的风格；软件测试的目的和原则；白盒与黑盒测试方法；单元测试，集成测试的方法；自顶向下及自底向上结合模块的步骤；递增式及非递增式测试的区别；归纳法与演绎法调试策略；测试用例的设计。教学难点：白盒测试中各种逻辑覆盖的基本思想；黑盒测试中等价划分、边界值分析方法。

4. 教学内容

(1) 编码

主要知识点：程序设计语言的分类；编码风格。

(2) 软件测试基础

主要知识点：软件测试的目标、准则、方法、步骤。

(3) 单元测试

主要知识点：测试重点；驱动程序；存根程序。

(4) 集成测试

主要知识点：概念；递增式组装方式；递减式组装方式；组装步骤；回归测试。

(5) 确认测试

主要知识点：概念；范围；步骤； α 测试和 β 多测试。

(6) 白盒测试技术

主要知识点：逻辑覆盖；控制结构测试。

(7) 黑盒测试技术

主要知识点：等价划分；边界值分析。

(8) 调试

主要知识点：调试过程；调试途径；调试原则。

(9) 软件可靠性

主要知识点：可靠性；可用性；估算平均无故障时间 MTTF；错误植入模型；分别测试法。

第八章 软件维护

1. 教学基本要求

掌握软件维护的类型、软件的可维护性以及软件的维护方法；了解软件维护的目标与任务，软件维护的特殊性以及软件维护的管理流程。

2. 要求学生掌握的基本概念、理论、技能

通过本章教学使学生掌握软件维护的内容、特点、方法、技术、文档；了解软件维护的各种困难，软件维护的特点，软件维护的文档；理解软件维护的内容，维护任务的实施。深刻理解如何提高软件的维护性。

3. 教学重点和难点

教学重点和教学难点是 领会改正性维护、适应性维护、完善性维护、预防性维护；可维护性的定义、可维护性的度量及提高可维护性的方法。

4. 教学内容

(1) 软件维护

主要知识点：概念；软件维护的特点；改正性维护；适应性维护；扩充与完善性维护；预防性维护；软件可维护性。

(2) 软件再工程过程

主要知识点：逆向工程；正向工程；软件再工程

第九章 面向对象方法学

1. 教学基本要求

掌握面向对象的概念，包括对象和类、对象的属性和操作，以及对象之间的关系；掌握面向对象程序设计的特性，如封装、继承和多态性；面向对象的建模，包括对象模型、动态模型和功能模型的概念。

2. 要求学生掌握的基本概念、理论、技能

通过本章教学使学生深刻理解对象、类、类的层次结构、方法和消息的实质以及对象之间的关系；掌握面向对象程序设计的特性以及面向对象建模的概念。

3. 教学重点和难点

教学重点和教学难点是对象之间的关系；面向对象建模的概念

4. 教学内容

(1) 面向对象方法学概述

主要知识点：面向对象程序设计的要点；优点。

(2) 面向对象概念

主要知识点：对象的定义和特点；类；实例；消息；方法；属性；封装；继承；多态性；重载。

(3) 面向对象建模

主要知识点：模型；建模顺序；与结构化建模的区别。

(4) 对象模型

主要知识点：类图的基本符号；关联；聚集；泛化；依赖和细化。

(5) 动态模型

主要知识点：事件；状态；行为。

(6) 功能模型

主要知识点：用例图；用例建模。

(7) 三种模型之间的关系

主要知识点：对象模型、动态模型以及功能模型之间的关系。

第十章 面向对象分析

1. 教学基本要求

掌握面向对象分析的基本过程；建立对象模型、动态模型和功能模型的方法确定对象、确定属性、定义操作、对象间通信。

2. 要求学生掌握的基本概念、理论、技能

通过本章教学使学生理解对象分析的基本过程；深刻理解对象模型、动态模型、功能模型的元素、结构和构造方法；掌握画对象图、画状态图、画数据流程图的方法；能够确定对象类、确定关联、确定属性，识别继承。

3. 教学重点和难点

教学重点和教学难点是：面向对象分析的过程、需求陈述，能够根据具体问题建立对象模型、动态模型、功能模型以及确定对象的服务。

4. 教学内容

(1) 面向对象分析的基本过程

主要知识点：概述；3个子模型；5个层次。

(2) 需求陈述

(3) 建立对象模型

主要知识点：确定类与对象；找出候选的类与对象；筛选出正确的类与对象；确定关联；划分主题；确定属性；识别继承关系。

(4) 建立动态模型

主要知识点：脚本；事件跟踪图。

(5) 建立功能模型

主要知识点：画出基本系统模型图；画出功能级数据流图；描述处理框功能。

(6) 定义服务

主要知识点：如何确定服务。

第十一章 面向对象设计

1. 教学基本要求

掌握面向对象设计的准则、启发性规则、软件重用、系统分解等方法；了解设计问题域、人机交互、管理等子系统以及设计优化。

2. 要求学生掌握的基本概念、理论、技能

通过本章教学使学生掌握面向对象设计的准则；掌握面向对象设计所用启发性规则、软件重用、系统分解等方法；了解设计问题域、人机交互、管理等子系统；设计优化。

3. 教学重点和难点

教学重点和教学难点是面向对象设计的准则和启发规则；软件重用的概念和方法；系统分解的方法。

4. 教学内容

(1) 面向对象设计

主要知识点：设计准则；启发规则

(2) 软件重用

主要知识点：概念；重用级别；类构件概念；类构件重用方式；软件重用的效益。

(3) 系统分解

主要知识点：子系统之间的两种交互方式；组织系统的两种方案；设计问题域子系统；设计人机交互子系统；设计任务管理子系统；设计数据管理子系统。

第十二章 面向对象实现

1. 教学基本要求

了解面向对象程序设计语言；程序设计风格；面向对象测试。

2. 要求学生掌握的基本概念、理论、技能

通过本章教学使学生了解面向对象程序设计风格；了解面向对象测试。

3. 教学重点和难点

教学重点是面向对象语言的特点以及如何选择面向对象的开发工具，养成良好的程序设计风格；教学难点是面向对象的软件测试的特点和策略，能够设计测试用例。

4. 教学内容

(1) 程序设计语言

主要知识点：面向对象语言的优点；技术特点。

(2) 程序设计风格

主要知识点：提高可重用性；提高可扩充性；提高健壮性。

(3) 测试策略

主要知识点：面向对象的单元测试；面向对象的集成测试；面向对象的确认测试。

(4) 设计测试用例

主要知识点：测试类的方法；集成测试方法。

第十三章 软件项目管理

1. 教学基本要求

了解软件开发成本结算、软件项目进度安排、软件项目计划内容及软件质量的定义、质量保证。

2. 要求学生掌握的基本概念、理论、技能

通过本章教学使学生了解软件开发成本结算、软件项目进度安排、软件项目计划内容及软件质量的定义、质量保证的基本概念。

3. 教学重点和难点

教学重点是估算软件规模和工作量的技术，能够根据项目制定合理的进度计划，团队组织的基本方法，能力成熟度模型。教学难点是软件规模的估算和能力成熟度模型。

4. 教学内容

(1) 估算软件规模

主要知识点：代码行技术；功能点技术。

(2) 工作量估算

主要知识点：静态单变量模型；动态多变量模型；COCOM02 模型。

(3) 进度计划

主要知识点：估算开发时间；Gantt 图；工程网络；估算工程进度；关键路径；机动时间

(4) 人员组织

主要知识点：民主制程序员组；主程序员组；现代程序员组。

(5) 质量保证

主要知识点：软件质量；软件质量保证措施。

(6) 软件配置管理

主要知识点：软件配置；软件配置管理过程。

(7) 能力成熟度模型

主要知识点：CMM 的作用；5 个成熟度等级。

四、学时分配表

1. 讲授内容及学时分配

序号	内容	课时	备注
1	软件工程学概述	2	

2	可行性研究	4	
3	需求分析	4	
4	形式化说明技术	4	
5	总体设计	3	
6	详细设计	4	
7	实现	4	
8	维护	4	
9	面向对象方法学引论	3	
10	面向对象分析	2	
11	面向对象设计	2	
12	面向对象实现	2	
13	软件项目管理	2	
合计		40	

2. 实践内容及学时分配

序号	项目名称	内容提要	学时	必/选开
1	软件开发的准备工作	确定项目，分组，合理分工，熟悉软件开发环境，培养团队精神。	2	必做
2	软件可行性分析	开展团队合作，完成软件可行性分析报告编制。	2	必做
3	软件需求分析	完成软件需求分析，并建立相关模型。掌握面向对象分析和传统分析方法的不同内容。	4	必做
4	软件设计	完成软件的设计，包括子系统划分、数据库设计及用户界面设计并完成设计报告。	6	必做
5	软件实现	熟练掌握一种面向对象程序设计语言，将设计转化为可运行的程序代码。	4	必做
6	软件测试	树立正确的软件测试思想，正确运用软件测试的过程与方法，给出测试报告。	2	必做
合计			20	

五、主用教材及参考书

(一) 主用教材：

《软件工程导论》主编：张海藩 出版社：清华大学出版社 出版时间：2012年。

(二) 参考书:

- 1.《软件工程 原理、方法与应用》主编:史济民 出版社:高等教育出版社 出版时间:2009年。
- 2.《软件工程》主编:齐治昌 出版社:高等教育出版社 出版时间:2012年。
- 3.《实用软件工程》主编:殷人昆 出版社:清华大学出版社 出版时间:2010年。

执笔:张秦 王晓洁

审定:皮磊 梁桂珍