

普通高校本科计算机专业

特色

教材精选

软件工程导论学习辅导

张海藩 编著

<http://www.tup.com.cn>

清华大学出版社





软件工程师自主学习辅导

第 1 章



普通高校本科计算机专业**特色**教材精选

软件工程导论学习辅导

张海藩 编著

清华大学出版社
北京

内 容 简 介

作者编著的《软件工程导论》被国内许多高校选作软件工程课的教材,累计销售约70万册。为配合读者学习软件工程,作者又精心编写了这本学习辅导书。

本书正文共10章,每章由三部分内容组成:第一部分系统扼要地复习本单元的重点内容;第二部分给出与本单元内容密切配合的习题;第三部分是习题解答,对典型习题的解答不是简单地给出答案,而是仔细分析题目,讲解解题思路,从而帮助读者举一反三,学会用软件工程方法学分析问题、解决问题。

正文后面有两个附录,分别给出了模拟试题和模拟试题答案。读者可以用这些试题自我测试,检验学习效果。

本书可以与《软件工程导论(第四版)》配合使用,也可供学习软件工程课程的读者单独使用,以检测学习效果。

版权所有,翻印必究。举报电话:010-62782989 13901104297 13801310933

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

软件工程导论学习辅导/张海藩编著. —北京:清华大学出版社,2004.9

(普通高校本科计算机专业特色教材精选)

ISBN 7-302-09213-3

I. 软… II. 张… III. 软件工程—高等学校—教材 IV. TP311.5

中国版本图书馆CIP数据核字(2004)第082088号

出版者:清华大学出版社

<http://www.tup.com.cn>

社总机:010-62770175

地址:北京清华大学学研大厦

邮编:100084

客户服务:010-62776969

责任编辑:张 龙

印装者:清华大学印刷厂

发行者:新华书店总店北京发行所

开 本:185×260 印张:14.5 字数:317千字

版 次:2004年9月第1版 2004年9月第1次印刷

书 号:ISBN 7-302-09213-3/TP·6475

印 数:1~5000

定 价:19.00元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770175-3103 或(010)62795704

出版说明

INTRODUCTION

在我国高等教育逐步实现大众化后，越来越多的高等学校将会面向国民经济发展的第一线，为行业、企业培养各级各类高级应用型专门人才。为此，教育部已经启动了“高等学校教学质量和教学改革工程”，强调要以信息技术为手段，深化教学改革和人才培养模式改革。如何根据社会的实际需要，根据各行各业的具体人才需求，培养具有特色显著的人才，是我们共同面临的重大问题。具体地，培养具有一定专业特色的和特定能力强的计算机专业应用型人才则是计算机教育要解决的问题。

为了适应 21 世纪人才培养的需要，培养具有特色的计算机人才，急需一批适合各种人才培养特点的计算机专业教材。目前，一些高校在计算机专业教学和教材改革方面已经做了大量工作，许多教师在计算机专业教学和科研方面已经积累了许多宝贵经验。将他们的教研成果转化为教材的形式，向全国其他学校推广，对于深化我国高等学校的教学改革是一件十分有意义的事。

清华大学出版社在经过大量调查研究的基础上，决定编写出版一套“普通高校本科计算机专业特色教材精选”。本套教材是针对当前高等教育改革的新形势，以社会对人才的需求为导向，主要以培养应用型计算机人才为目标，立足课程改革和教材创新，广泛吸纳全国各地的高等院校计算机优秀教师参与编写，从中精选出版确实反映计算机专业教学方向的特色教材，供普通高等院校计算机专业学生使用。

本套教材具有以下特点：

1. 编写目的明确

本套教材是在深入研究各地各学校办学特色的基础上，面向普通高校的计算机专业学生编写的。学生通过本套教材，主要学习计算机科学与技术专业的基本理论和基本知识，接受利用计算机解决实际问题的基本训练，培养研究和开发计算机系统，特别是应用系统的基本能力。

2. 理论知识与实践训练相结合

根据计算学科的三个学科形态及其关系，本套教材力求突出学科的理论与实践紧密结合的特征，结合实例讲解理论，使理论来源于实践，又进一步指导实践，学生通过实践深化对理论的理解，更重要的是使学生学会理论方法的实际运用。在编写教材时突出实用性，并做到通俗易懂，易教易学，使学生不仅知其然，知其所以然，还要会其如何然。

3. 注意培养学生的动手能力

每种教材都增加了能力训练部分的内容，学生通过学习和练习，能比较熟练地应用计算机知识解决实际问题。既注重培养学生分析问题的能力，也注重培养学生解决问题的能力，以适应新经济时代对人才的需要，满足就业要求。

4. 注重教材的立体化配套

大多数教材都将陆续配套教师用课件、习题及其解答提示，学生上机实验指导等辅助教学资源，有些教材还提供能用于网上下载的文件，以方便教学。

由于各地区各学校的培养目标、教学要求和办学特色均有所不同，所以对特色教学的理解也不尽一致，我们恳切希望大家在使用教材的过程中，及时地给我们提出批评和改进意见，以便我们做好教材的修订改版工作，使其日趋完善。

我们相信经过大家的共同努力，这套教材一定能成为特色鲜明、质量上乘的优秀教材，同时，我们也希望通过本套教材的编写出版，为“高等学校教学质量和教学改革工程”作出贡献。

清华大学出版社

前言

PREFACE

《软件工程导论》已经出版了四个版本，先后荣获全国普通高等学校工科电子类专业优秀教材二等奖和一等奖，国内许多高校用它作为软件工程课的教材，累计销售约 70 万册。

在此期间陆续有读者来信索要习题答案或探讨习题解法。确实，软件工程是一门实践性很强的学科，学好软件工程不仅要深入理解它的基本概念、原理、技术和方法，更重要的是通过实践学会用软件工程方法学分析问题、解决问题。为配合读者学习软件工程，编者又精心编写了这本学习辅导书。

本书正文共 10 章。第 1 章“软件工程概论”，涵盖教材（第四版，下同）第 1 章的重点内容；第 2 章“结构化分析”，涵盖教材第 2、3、4 章的重点内容；第 3 章“结构化设计”，涵盖教材第 5、6 章的重点内容；第 4 章“结构化实现”，涵盖教材第 7 章的重点内容；第 5 章“维护”，涵盖教材第 8 章的重点内容；第 6 章“面向对象方法学引论”，涵盖教材第 9 章的重点内容；第 7 章“面向对象分析”，涵盖教材第 10 章的重点内容；第 8 章“面向对象设计”，涵盖教材第 11 章的重点内容；第 9 章“面向对象实现”，涵盖教材第 12 章的重点内容；第 10 章“软件项目管理”，涵盖教材第 13 章的重点内容。

每章均由三部分内容组成：第一部分系统扼要地复习本单元的重点内容；第二部分给出与本单元内容密切配合的习题，其中一些题目与教材上的题目相同，另一些题目是教材上没有的，当然，也有一些教材上的题目没有包含在本书中，可作为软件工程课的练习题，留给读者独立完成；第三部分是习题解答，对典型习题的解答不是简单地给出答案，而是仔细分析题目，讲解解题思路，从而有助于读者举一反三，学会用软件工程方法学分析问题和解决问题。

正文后面有两个附录：附录 A 是模拟试题，共给出三份试卷；附录 B 是模拟试题答案。读者可以用这些试题自我测试，检验学习效果。

编者

2004 年 5 月

目 录

CONTENTS

第 1 章 软件工程概论	1
1.1 软件危机	1
1.1.1 软件危机简介	1
1.1.2 产生软件危机的原因	1
1.1.3 消除软件危机的途径	2
1.2 软件工程	3
1.2.1 软件工程简介	3
1.2.2 软件工程的基本原理	3
1.2.3 软件工程方法学	3
1.3 软件生命周期	4
1.4 软件过程	5
习题	8
习题解答	10
第 2 章 结构化分析	13
2.1 可行性研究的任务	13
2.2 可行性研究过程	13
2.3 需求分析的任务	14
2.4 与用户沟通的方法	14
2.5 分析建模与规格说明	15
2.6 实体-联系图	16
2.7 数据流图	16
2.8 数据字典	17
2.9 状态转换图	17
2.10 其他图形工具	18
2.11 验证软件需求	19

2.12	成本/效益分析	19
2.12.1	成本估计	19
2.12.2	成本/效益分析方法	19
2.13	形式化说明技术	20
	习题	21
	习题解答	23
第3章	结构化设计	41
3.1	软件设计的任务	41
3.1.1	概要设计	41
3.1.2	详细设计	41
3.2	分析与设计的关系	42
3.3	设计原理	43
3.3.1	模块化与模块独立	43
3.3.2	抽象	43
3.3.3	逐步求精	44
3.3.4	信息隐藏	45
3.3.5	局部化	45
3.4	度量模块独立性的标准	45
3.4.1	耦合	45
3.4.2	内聚	45
3.5	启发规则	46
3.6	描绘软件结构的图形工具	46
3.7	面向数据流的设计方法	46
3.7.1	数据流的类型	47
3.7.2	设计步骤	47
3.8	人机界面设计	48
3.8.1	应该考虑的设计问题	49
3.8.2	人机界面设计过程	50
3.8.3	人机界面设计指南	50
3.9	过程设计	51
3.10	过程设计的工具	51
3.11	面向数据结构的设计方法	53
3.12	程序复杂程度的定量度量	54
3.12.1	McCabe 方法	54
3.12.2	Halstead 方法	54
	习题	55
	习题解答	59

第 4 章 结构化实现	79
4.1 编码	79
4.1.1 选择程序设计语言	79
4.1.2 编码风格	80
4.2 软件测试基础	80
4.3 单元测试	81
4.4 集成测试	82
4.5 白盒测试技术	84
4.5.1 逻辑覆盖	84
4.5.2 控制结构测试	85
4.6 黑盒测试技术	86
4.6.1 等价划分	86
4.6.2 边界值分析	87
4.6.3 错误推测	88
4.7 调试	88
4.7.1 调试过程	88
4.7.2 调试途径	88
4.8 软件可靠性	89
4.8.1 基本概念	89
4.8.2 估算平均无故障时间的方法	90
习题	92
习题解答	94
第 5 章 维护	105
5.1 软件维护的定义	105
5.2 软件维护的特点	106
5.3 软件维护过程	107
5.4 软件的可维护性	108
5.4.1 决定软件可维护性的因素	108
5.4.2 文档	109
5.4.3 可维护性复审	109
5.5 预防性维护	109
5.6 软件再工程过程	110
习题	111
习题解答	112
第 6 章 面向对象方法学引论	117
6.1 面向对象方法学概述	117

6.1.1	面向对象方法学的要点	117
6.1.2	面向对象方法学的优点	118
6.1.3	面向对象的软件过程	120
6.2	面向对象的概念	121
6.2.1	对象	121
6.2.2	其他概念	122
6.3	面向对象建模	123
6.4	对象模型	124
6.4.1	类图的基本符号	124
6.4.2	表示关系的符号	124
6.5	动态模型	126
6.6	功能模型	126
6.6.1	用例图	126
6.6.2	用例建模	128
6.7	3种模型之间的关系	128
	习题	129
	习题解答	130
第7章	面向对象分析	137
7.1	面向对象分析的基本过程	137
7.1.1	概述	137
7.1.2	3个子模型与5个层次	138
7.2	需求陈述	138
7.3	建立对象模型	139
7.3.1	确定类与对象	139
7.3.2	确定关联	140
7.3.3	划分主题	141
7.3.4	确定属性	141
7.3.5	识别继承关系	141
7.3.6	反复修改	142
7.4	建立动态模型	142
7.4.1	编写脚本	142
7.4.2	画事件跟踪图	143
7.4.3	画状态图	143
7.4.4	审查动态模型	144
7.5	建立功能模型	144
7.6	定义服务	144
	习题	145

习题解答	146
第 8 章 面向对象设计	151
8.1 面向对象设计的准则	151
8.2 启发规则	152
8.3 软件重用	153
8.3.1 概述	153
8.3.2 类构件	154
8.3.3 软件重用的效益	155
8.4 系统分解	155
8.5 设计问题域子系统	156
8.6 设计人机交互子系统	157
8.7 设计任务管理子系统	157
8.8 设计数据管理子系统	158
8.9 设计类中的服务	159
8.10 设计关联	159
8.11 设计优化	160
习题	161
习题解答	162
第 9 章 面向对象实现	171
9.1 程序设计语言	171
9.2 程序设计风格	172
9.3 面向对象的测试策略	173
9.4 设计测试用例	174
9.4.1 测试类的技术	174
9.4.2 集成测试技术	175
习题	175
习题解答	176
第 10 章 软件项目管理	181
10.1 估算软件规模	181
10.1.1 代码行技术	181
10.1.2 功能点技术	181
10.2 估算工作量	183
10.2.1 静态单变量模型	183
10.2.2 动态多变量模型	183
10.2.3 COCOMO2 模型	183

10.3	进度计划	184
10.3.1	估算开发时间	184
10.3.2	Gantt 图	185
10.3.3	工程网络	185
10.3.4	估算工程进度	185
10.3.5	关键路径	186
10.3.6	机动时间	186
10.4	人员组织	186
10.4.1	民主制程序员组	187
10.4.2	主程序员组	187
10.4.3	现代程序员组	187
10.5	质量保证	188
10.5.1	软件质量	188
10.5.2	软件质量保证措施	188
10.6	软件配置管理	188
10.6.1	软件配置	188
10.6.2	软件配置管理过程	189
10.7	能力成熟度模型	190
	习题	191
	习题解答	193
附录		199
附录 A	模拟试题	199
	试卷一	199
	试卷二	201
	试卷三	203
附录 B	模拟试题参考答案	206
	试卷一参考答案	206
	试卷二参考答案	208
	试卷三参考答案	210
参考文献		213

软件工程概论

1.1 软件危机

为吸取历史经验教训,应该认真研究产生软件危机的原因,探讨消除软件危机的途径。

1.1.1 软件危机简介

通常把在计算机软件的开发与维护过程中所遇到的一系列严重问题笼统地称为软件危机。这些问题绝不仅仅是不能正常运行的软件才具有的,实际上,几乎所有软件都不同程度地存在这些问题。

概括地说,软件危机包含下述两方面的问题:

- (1) 如何开发软件,以满足社会对软件日益增长的需求;
- (2) 如何更有效地维护数量不断膨胀的已有软件。

具体地说,软件危机主要有以下一些典型表现:

- 对软件开发成本和进度的估计常常很不准确。
- 经常出现用户对“已完成的”软件产品不满意的情况。
- 软件产品的质量往往达不到要求。
- 软件通常是很难维护的。
- 软件往往没有适当的文档资料。
- 软件成本在计算机系统总成本中所占的比例逐年上升。
- 软件开发生产率提高的速度远远不能满足社会对软件产品日益增长的需求。

鉴于软件危机周期长且难于预测,把它称为“软件萧条”或“软件困扰”可能更恰当一些。

1.1.2 产生软件危机的原因

1. 客观原因

软件是计算机系统逻辑部件而不是物理部件,其显著特点是缺乏

“可见性”，因此，管理和控制软件开发过程相当困难。此外，软件维护通常意味着改正或修改原有的设计，从而使得软件较难维护。

软件的另一个突出特点是规模庞大，而程序复杂性将随着程序规模增加以指数速度上升。软件可能具有的状态数通常都是天文数字，无法完全预见软件可能遇到的每一种情况。

2. 主观原因

在计算机系统发展的早期阶段开发软件的个体化特点，使得许多软件工程师对软件开发和维护有不少糊涂认识，在实际工作中或多或少地采用了错误的方法，这是使软件问题发展成软件危机的主要原因。

错误的认识和做法主要表现为，忽视软件需求分析的重要性，认为软件开发就是写程序，轻视软件维护。

事实上，对用户的需求没有完整准确的认识就匆忙着手编写程序，是许多软件开发工程失败的主要原因之一。

必须认识到，软件开发和维护要经历一个漫长的时期（称为软件生命周期），编写程序只是软件开发过程中的一个相对来说比较次要的阶段。

另一方面还必须认识到，程序只是完整的软件产品的一个组成部分，一个软件产品必须由一个完整的配置组成。软件配置主要包括程序、文档和数据等成分。

严酷的事实是，在软件开发的后期阶段引入一个变动比在早期引入同一个变动所需付出的代价高几百倍甚至上千倍。所谓软件维护，就是在软件开发工作已经结束之后在使用现场对软件进行修改。因此，维护是极端艰巨复杂的工作，需要花费很大代价。由此可见，轻视软件维护是一个最大的错误。软件工程的一个重要目标就是提高软件的可维护性，减少软件维护的代价。

1.1.3 消除软件危机的途径

首先应该树立对计算机软件的正确认识。软件是程序、数据及文档的完整集合。其中，程序是能够完成预定功能和性能的可执行的指令序列；数据是使程序能够适当地处理信息的数据结构；文档是开发、使用和维护程序所需要的图文资料。

软件开发应该是组织良好、管理严密、各类人员团结协作共同完成的工程项目。必须充分吸取和借鉴人类长期以来从事各种工程项目所积累的行之有效的原理、概念、技术和方法，并研究能更有效地开发软件的技术和方法。

应该积极开发和使用计算机辅助软件工程(CASE)工具。

总之，为了消除软件危机，既要有技术措施（方法和工具），又要有必要的组织管理措施。软件工程正是从技术和管理两方面研究如何更好地开发和维护软件的一门新兴的工程学科。

1.2 软件工程

1.2.1 软件工程简介

软件工程是指导计算机软件开发和维护的一门工程学科,该学科的目的是生产出能按期交付的、在预算范围内的、满足用户需求的、质量合格的软件产品。

软件工程具有下述本质特性:

- 软件工程关注于大型程序的构造。
- 软件工程的中心课题是控制复杂性。
- 软件产品交付使用后仍然需要经常修改。
- 开发软件的效率非常重要。
- 开发人员和谐地合作是成功地开发软件的关键。
- 软件必须有效地支持它的用户。
- 在软件工程领域中通常由具有一种文化背景的人替具有另一种文化背景的人开发产品。

1.2.2 软件工程的基本原理

- 用分阶段的生命周期计划严格管理。
- 坚持进行阶段评审。
- 实行严格的产品控制。
- 采用现代程序设计技术。
- 结果应能清楚地审查。
- 开发小组的人员应该少而精。
- 承认不断改进软件工程实践的必要性。

1.2.3 软件工程方法学

通常把在软件生命周期全过程中使用的一整套技术方法的集合称为方法学,也称为范型。

软件工程方法学包含 3 个要素:方法、工具和过程。其中,方法是完成软件开发各项任务的技术方法,回答“怎样做”的问题;工具是为运用方法而提供的自动的或半自动的软件工程支撑环境;过程是为了获得高质量的软件所需要完成的一系列任务的框架,它规定了完成各项任务的工作步骤,回答“何时做”的问题。

目前使用得最广泛的软件工程方法学分别是传统方法学和面向对象方法学。

1. 传统方法学(结构化范型)

(1) 采用结构化技术(结构化分析、结构化设计和结构化实现)完成软件开发的各项任务。

(2) 把软件生命周期划分成若干个阶段,然后顺序完成各个阶段的任务。

(3) 每个阶段的开始和结束都有严格的标准,对于任何两个相邻的阶段而言,前一阶段的结束标准就是后一阶段的开始标准。

(4) 在每个阶段结束之前都必须正式地进行严格的技术审查和管理复审。

2. 面向对象方法学(面向对象范型)

(1) 把对象作为融合了数据及在数据上的操作的软件构件。也就是说,用对象分解取代了传统方法的功能分解。

(2) 把所有对象都划分成类。

(3) 按照父类与子类的关系,把若干个相关类组织成一个层次结构的系统。

(4) 对象彼此间仅能通过发送消息互相联系。

使用结构化范型开发出的软件,在本质上是一个单元,这是用结构化范型开发大型软件产品时不甚成功的一个重要原因。相反,当正确地使用面向对象范型时,开发出的软件产品是由许多小的、相对独立的单元(对象)组成的。因此,面向对象范型降低了软件产品的复杂度,从而简化了软件开发与维护工作。

1.3 软件生命周期

概括地说,软件生命周期由软件定义、软件开发和运行维护(也称为软件维护)3个时期组成,通常把前两个时期再进一步划分成若干个阶段。

软件定义时期的基本任务是:确定软件开发工程的总目标;研究该项目的可行性;分析确定客户对软件产品的需求;估算完成该项目所需的资源和成本,并且制定工程进度表。这个时期的工作称为系统分析,由系统分析员负责完成。

通常把软件定义时期进一步划分成问题定义、可行性研究和需求分析3个阶段。其中需求分析阶段应该完成的工作包括需求获取和需求分析两部分。揭示客户需求的过程称为需求获取或需求收集;一旦确定了最初的一系列需求,就应该进一步提炼和扩展这些需求,并用软件需求规格说明书把客户需求准确地记录下来,这个过程称为需求分析。

软件开发时期具体设计和实现在前一个时期定义的软件,它通常由下述4个阶段组成:总体设计(又称为结构设计);详细设计;编码和单元测试;综合测试。其中前两个阶段又称为系统设计,后两个阶段又称为系统实现。

运行维护时期的主要任务是,通过对已交付使用的软件做必要的修改,使软件持久地满足客户的需求。具体地说,当软件在使用过程中发现错误时应该加以改正;当环境改变时应该修改软件以适应新的环境;当用户有新要求时应该及时改进或扩充软件以满足用户的新需要。通常对维护时期不再进一步划分阶段,但是每一次维护活动本质上都是一次压缩和简化了的定义和开发过程。

使用结构化范型开发软件时,软件生命周期各阶段中使用的概念及应完成的任务性质显著不同。需求分析阶段的基本任务是确定软件必须“做什么”,使用的概念主要是“功