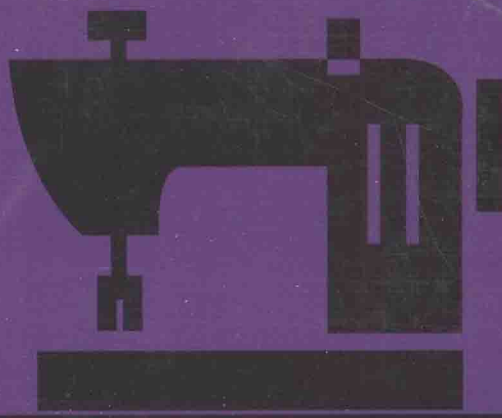


HZ BOOKS  
华章科技



# UML

## 统一建模基础教程



刘小松 等编著

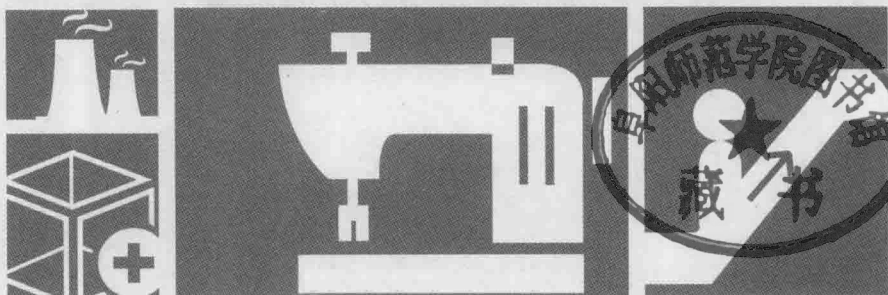
大量的范例讲解，丰富的扩展练习，使读者快速进入项目实战状态  
详解Rational Rose软件的操作方法，使UML建模过程变得轻松异常  
附赠多媒体教学视频、PPT教学课件，以及全书实例的源程序和相关素材



机械工业出版社  
China Machine Press

# UML

## 统一建模基础教程



刘小松 等编著

UML统一建模基础教程

机械工业出版社

机械工业出版社地址：北京市西城区百万庄大街24号

机械工业出版社发行部：北京市西城区百万庄大街24号

电话：(010) 68995100

网址：<http://www.cmpbook.com>

机械工业出版社发行部：北京市西城区百万庄大街24号

电话：(010) 68995100

网址：<http://www.cmpbook.com>

机械工业出版社发行部：北京市西城区百万庄大街24号

电话：(010) 68995100

网址：<http://www.cmpbook.com>



机械工业出版社  
China Machine Press

## 图书在版编目 ( CIP ) 数据

UML统一建模基础教程/刘小松等编著. —北京: 机械工业出版社, 2014.4

ISBN 978-7-111-46083-1

I. ①U… II. ①刘… III. ①面向对象语言—程序设计—教材 IV. ①TP312

中国版本图书馆CIP数据核字 (2014) 第045307号

UML 是以面向对象的方式来描述任何类型的系统, 应用领域非常广泛, 其中最常用的是建立软件系统的模型, 本书依据统一建模语言 UML 与面向对象编程语言, 结合实际案例, 深入全面地讲解了 UML 的基本概念和建模方法。

本书共 14 章, 前 4 章介绍了建模基础知识、面向对象技术、建模工具 Rational Rose 和 UML 的一些基本理论和概念; 第 5~13 章详尽介绍了 UML 中的用例图、类图、对象图、包图、序列图、协作图、活动图、状态图、构件图、部署图、Rational Rose 统一过程的概念以及在实际中的应用; 第 14 章通过一个建模的综合案例——学籍管理系统对使用 Rational Rose 进行 UML 建模的全过程进行深入剖析。此外, 章后配有适量的扩展练习, 并在附录中提供 5 个综合的建模案例, 以加深读者的理解。

本书既可以作为各大院校计算机专业学生学 UML 和面向对象技术的教材, 也可作为广大软件开发人员和系统架构设计人员自学 UML 的参考和指导用书。

# UML 统一建模基础教程

刘小松 等编著

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 夏非彼 迟振春

印刷: 中国电影出版社印刷厂

开本: 188mm×260mm 1/16

书号: ISBN 978-7-111-46083-1

ISBN 978-7-89405-334-3 (光盘)

版次: 2014 年 6 月第 1 版第 1 次印刷

印张: 22.25

定价: 49.00 元 (附光盘)

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

购书热线: (010) 68326294 88379649 68995259

投稿热线: (010) 82728184 88379604

读者信箱: hzjsj@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光/邹晓东

# 前言

本书是一本关于讲解 UML 基础和 Rose 建模的教材，书中包含了 UML 的基础知识、UML 的基本元素以及 UML 的使用方法，Rational Rose 是目前最为业界瞩目的可视化软件开发工具，通过它可以便捷、高效地完成 UML 的建模工作。

## 本书内容

本书共有 14 章和 5 个附录。

第 1 章：本章主要对软件工程的目标、过程和原则，以及 UML 等方面的问题和基本概念进行简要介绍，以便读者在进入软件建模之前，对软件工程的一些基本知识有一定的理解。

第 2 章：本章主要介绍面向对象开发、面向对象设计和面向对象实现的方法，以便读者对 UML 有初步的认识。

第 3 章：本章对 UML 的主流开发工具——Rational Rose 进行介绍，包括其安装、使用、视图模型和生成代码的方法。

第 4 章：本章对 UML 进行概要介绍，包括常用的 UML 元素、UML 的通用机制、UML 的扩展机制和对象约束语言等。

第 5 章：介绍用例图的概念和作用，讲解用例图的重要组成元素和如何通过 Rational Rose 创建用例图。

第 6 章：介绍了类图和对象图的基本概念、类图到数据库的转换以及在 UML 中的表示。通过讲解使用 Rational Rose 创建类图的方式，使读者在学完本章后，能够根据类图和对象图描绘出系统的静态结构。

第 7 章：本章主要是对包图中的基本概念以及它们的使用方法进行详细介绍。

第 8 章：针对 UML 中的序列图进行介绍，包括它的基本概念、组成和使用 Rational Rose 创建序列图的方法。

第 9 章：通过本章的学习，可以掌握协作图的基本概念以及使用方法，进而能够熟练使用协作图描述系统中对象之间的交互。

第 10 章：介绍活动图的基本概念、组成元素，以及使用 Rational Rose 创建活动图的方法。

第 11 章：读者在学习本章后，能够从整体上理解状态图并掌握状态图的画法。

第 12 章：对构件图和部署图进行详细讲解，读者在学完本章后，能够根据



构件图和部署图的基本概念，创建图中的各种模型元素，描绘出系统的物理结构，并将前面介绍过的其他图结合起来，完成对整个系统的建模。

第 13 章：介绍统一过程的概念、结构、配置和实现 Rational 统一过程的方法。

第 14 章：给出一个学籍管理系统的综合实例，将前面各章介绍的 UML 图形以及模型元素综合起来，实现对一个完整系统的设计和建模过程。整个过程按照软件设计的一般流程进行，让读者对实际项目的系统建模有一个直观的认识。

附录 A~E 通过新闻发布系统、网上教学系统、网上购物商店、汽车租赁系统和超市管理系统 5 个综合性的建模实例来巩固读者对 UML 建模知识的理解。

## 面向读者

本书既可以作为各大院校计算机专业学生学习 UML 和面向对象技术的教材，也可作为广大软件开发人员和系统架构设计人员自学 UML 的参考和指导用书。

## 鸣谢

本书主要由刘小松编写，此外，王坚宁、胡书敏、王相羽、袁润非、杨阳、王炳乾、苏宝保、张琴、李龙、王华、李辉、刘峰、徐浩、李建国、马建军、唐爱华、苏小平、朱丽云、马淑娟、周毅、张乐、李大勇、许小荣、魏勇、王云等同志也参与了本书的部分编写工作。

由于作者水平有限，书中错误、纰漏之处难免，欢迎广大读者、同仁批评斧正。

编者

2014 年 3 月

# 目 录

32	.....	2.3.4 面向对象类	12
30	.....	2.4 基于UML的面向对象建模	14
29	.....	2.5 扩展练习	15
20	.....	第3章 建模工具 Rational Rose	20
19	.....	3.1 Rational Rose 的基础知识	21
18	.....	3.2 Rational Rose 的进阶	22
17	.....	第1章 建模基础知识	1
16	.....	1.1 软件工程概述	1
15	.....	1.1.1 软件工程的产生	1
14	.....	1.1.2 软件工程的内容和基本原理	3
13	.....	1.1.3 现代软件工程	4
12	.....	1.2 建模概述	4
11	.....	1.2.1 什么是模型	4
10	.....	1.2.2 建模的原理	5
9	.....	1.2.3 为什么要建模	6
8	.....	1.2.4 建模的目标和原则	7
7	.....	1.3 UML 概述	7
6	.....	1.3.1 UML 的起源和发展	8
5	.....	1.3.2 UML 的主要特点	9
4	.....	1.3.3 UML 的应用领域	9
3	.....	1.3.4 UML 的模型种类	10
2	.....	1.4 扩展练习	15
1	.....	第2章 面向对象技术	16
0	.....	2.1 面向对象的基本知识	16
0	.....	2.1.1 类的含义	17
0	.....	2.1.2 消息和事件	17
0	.....	2.1.3 面向过程	18
0	.....	2.2 面向对象的特征	19
0	.....	2.2.1 抽象	19
0	.....	2.2.2 封装	20
0	.....	2.2.3 继承	21
0	.....	2.2.4 多态	22
0	.....	2.3 面向对象的开发	22
0	.....	2.3.1 系统调查和需求分析	23
0	.....	2.3.2 面向对象分析	23
0	.....	2.3.3 面向对象设计	26



## 目 录

2.3.4 面向对象实现	27
2.4 基于 UML 的面向对象建模	29
2.5 扩展练习	29
<b>第 3 章 建模工具 Rational Rose</b>	<b>30</b>
3.1 Rational Rose 的基础知识	30
3.2 Rational Rose 的安装	32
3.3 Rational Rose 的使用	34
3.3.1 Rational Rose 的启动界面	34
3.3.2 Rational Rose 的主界面	36
3.3.3 Rational Rose 的常用操作	41
3.4 Rational Rose 的 4 种视图模型	50
3.4.1 用例视图	51
3.4.2 逻辑视图	55
3.4.3 部署视图	60
3.4.4 构件视图	61
3.5 利用 Rational Rose 生成代码	63
3.5.1 生成代码的方法	63
3.5.2 逆向工程	68
3.6 扩展练习	69
<b>第 4 章 初识 UML</b>	<b>71</b>
4.1 UML 的构成	71
4.1.1 视图	71
4.1.2 图	72
4.2 模型元素	74
4.2.1 事物	74
4.2.2 关系	78
4.3 通用机制	79
4.3.1 规格说明	79
4.3.2 UML 的修饰	79
4.3.3 通用划分	80
4.4 UML 中的扩展机制	80
4.4.1 体系结构	81
4.4.2 构造型	85
4.4.3 标记值	88
4.4.4 约束	89
4.5 对象约束语言	91





4.5.1	OCL 概述	91
4.5.2	OCL 的标准类型	92
4.5.3	OCL 的表达式	97
4.5.4	OCL 的应用	98
4.6	扩展练习	99
<b>第 5 章</b>	<b>用例图</b>	<b>100</b>
5.1	用例图的概念	100
5.1.1	参与者	100
5.1.2	用例	102
5.2	参与者之间的关系	103
5.3	用例的确定	104
5.3.1	识别用例	104
5.3.2	用例粒度	105
5.3.3	用例规约	105
5.3.4	用例间的关系类型	106
5.4	系统边界	109
5.5	绘制用例图	110
5.5.1	创建用例图	111
5.5.2	绘制参与者	112
5.5.3	绘制用例	114
5.5.4	绘制用例和参与者的关系	115
5.5.5	绘制用例间的关系	116
5.6	用例图建模实例——图书管理系统用例图	117
5.6.1	需求分析	117
5.6.2	识别参与者	117
5.6.3	识别用例	118
5.6.4	构建用例模型	119
5.7	扩展练习	120
<b>第 6 章</b>	<b>类图和对象图</b>	<b>122</b>
6.1	类图	122
6.1.1	类图的概念	122
6.1.2	类的组成	123
6.1.3	类的类型	128
6.1.4	类的构造型	128
6.2	类图中的关系	130





110	6.3 类图到数据库的转换	134
110	6.3.1 基本映射转换	134
110	6.3.2 类到表的转换	136
110	6.3.3 关系的转换	139
110	6.3.4 完整性与约束验证	140
110	6.3.5 父表的约束	140
110	6.3.6 子表的约束	141
110	6.4 对象图	141
110	6.4.1 对象图的概念	141
110	6.4.2 对象图中的对象	142
110	6.4.3 对象图中的链	143
110	6.5 绘制类图	143
110	6.5.1 类图和类的绘制	143
110	6.5.2 类关系的绘制	152
110	6.6 绘制对象图	154
110	6.7 类图和对对象图的建模实例——图书管理系统	155
110	6.7.1 确定系统中的类	155
110	6.7.2 确定类的属性和操作	156
110	6.7.3 确定类的关系	157
110	6.7.4 根据类图绘制对象图	158
110	6.8 扩展练习	158
110	<b>第7章 包图</b>	<b>160</b>
110	7.1 包图的概念	160
110	7.1.1 模型的组织结构	160
110	7.1.2 包和包图	161
110	7.2 包的构成	162
110	7.2.1 包的名称	163
110	7.2.2 包的元素	163
110	7.2.3 包的可见性	163
110	7.2.4 包的构造型	164
110	7.2.5 包的子系统	165
110	7.3 包图中的关系	166
110	7.3.1 依赖关系	166
110	7.3.2 泛化关系	168
110	7.4 包的嵌套	168
110	7.5 绘制包图	169
110	7.5.1 包的创建	169

7.5.2	包中类的添加	171
7.5.3	绘制包的关系	172
7.6	包图的建模实例——图书管理系统包图	172
7.6.1	确定包的分类	172
7.6.2	确定包的关系	173
7.7	扩展练习	173
<b>第 8 章</b>	<b>序列图</b>	<b>174</b>
8.1	序列图的概念	174
8.2	序列图的组成	175
8.2.1	对象	175
8.2.2	生命线	176
8.2.3	激活	177
8.2.4	消息	177
8.3	序列图的对象行为	179
8.3.1	创建与销毁对象	179
8.3.2	分支与从属流	180
8.4	绘制序列图	180
8.4.1	创建序列图	181
8.4.2	创建对象	182
8.4.3	绘制和设置消息	184
8.4.4	绘制生命线	186
8.4.5	销毁对象	188
8.5	序列图的建模实例——系统管理员添加图书序列图	188
8.5.1	需求分析	188
8.5.2	确定序列对象	189
8.5.3	完成序列建模	190
8.6	扩展练习	190
<b>第 9 章</b>	<b>协作图</b>	<b>193</b>
9.1	协作图的概念	193
9.2	协作图的组成	195
9.2.1	对象	195
9.2.2	消息	195
9.2.3	链	196
9.3	绘制协作图	196
9.3.1	协作图的创建	197



9.3.2	对象的创建	198
9.3.3	链和消息的创建	200
9.3.4	序列图和协作图的转换	201
9.4	协作图的建模实例——系统管理员添加图书协作图	201
9.4.1	创建协作图的步骤	202
9.4.2	需求分析	202
9.4.3	确定协作图元素	202
9.4.4	确定元素之间的关系	203
9.4.5	创建完整的协作图	203
9.5	扩展练习	204
<b>第 10 章</b>	<b>活动图</b>	<b>206</b>
10.1	活动图的概念	206
10.2	活动图的组成元素	208
10.2.1	动作状态	208
10.2.2	活动状态	208
10.2.3	组合状态	209
10.2.4	分支与合并	209
10.2.5	分叉与结合	210
10.2.6	泳道	211
10.2.7	对象流	211
10.3	活动图的绘制	213
10.3.1	创建活动图	214
10.3.2	绘制初始和终止状态	215
10.3.3	绘制动作状态	215
10.3.4	绘制活动状态	216
10.3.5	绘制泳道	218
10.3.6	绘制转换	219
10.3.7	绘制分支	220
10.4	活动图的建模实例——借阅者借书活动图	221
10.4.1	确定需求分析	221
10.4.2	确定用例路径	222
10.4.3	完成活动图创建	223
10.5	扩展练习	223
<b>第 11 章</b>	<b>状态图</b>	<b>225</b>
11.1	状态图的概念	225
11.1.1	状态机	225

11.1.2	状态图	226
11.2	状态图的组成	228
11.2.1	状态	228
11.2.2	转换	231
11.2.3	判定	234
11.2.4	同步	235
11.2.5	事件	236
11.2.6	初始状态和终止状态	237
11.3	状态图的绘制	237
11.3.1	创建状态图	238
11.3.2	绘制初始和终止状态	239
11.3.3	绘制状态	239
11.3.4	绘制转换	240
11.3.5	绘制事件	241
11.3.6	绘制动作	241
11.3.7	绘制监护条件	242
11.4	状态图的建模实例——图书状态图	243
11.4.1	确定状态间的实体	243
11.4.2	确定状态图中实体的状态	243
11.4.3	确定相关事件	244
11.5	扩展练习	244
<b>第 12 章</b>	<b>构件图和部署图</b>	<b>246</b>
12.1	构件图的概念	246
12.1.1	构件	246
12.1.2	接口	249
12.1.3	关系	250
12.2	构件图的绘制	251
12.2.1	创建构件图	251
12.2.2	创建构件	253
12.2.3	绘制构建关系	254
12.3	部署图的概念	254
12.4	部署图的组成	255
12.4.1	节点	255
12.4.2	连接	256
12.4.3	设备	256
12.5	部署图的绘制	257



12.5.1	绘制节点	257
12.5.2	设置节点	258
12.5.3	绘制连接	260
12.6	构件图和部署图的建模实例——图书管理系统	261
12.6.1	确定系统构件图	261
12.6.2	确定系统部署图	262
12.7	扩展练习	264
<b>第 13 章 Rational 统一过程</b>		<b>265</b>
13.1	软件开发过程的概念	265
13.2	Rational 统一过程的概念	266
13.3	Rational 统一过程的结构	268
13.3.1	核心 workflow	269
13.3.2	迭代开发模式	270
13.4	Rational 统一过程的最佳实现	271
13.4.1	软件变更控制	271
13.4.2	软件质量验证	271
13.4.3	建立可视化的软件模型	272
13.4.4	基于构件的架构应用	272
13.4.5	需求管理	273
13.4.6	迭代式软件开发	274
13.5	Rational 统一过程的开发模型	275
13.5.1	动态开发	275
13.5.2	静态开发	280
13.5.3	面向架构的过程	281
13.6	Rational 统一过程的配置和实现	283
13.6.1	Rational 统一过程的配置	284
13.6.2	Rational 统一过程的实现	284
13.7	扩展练习	285
<b>第 14 章 学籍管理系统</b>		<b>286</b>
14.1	系统需求分析	286
14.2	系统建模	288
14.2.1	创建系统用例模型	289
14.2.2	创建系统静态模型	291
14.2.3	创建系统动态模型	294
14.2.4	创建系统部署模型	305

附录 A	新闻发布系统	308
A.1	需求分析	308
A.2	系统建模	308
A.3	创建系统动态模型	310
A.4	创建系统部署模型	313
附录 B	网上教学系统	315
B.1	需求分析	315
B.2	系统建模	315
B.3	创建系统动态模型	318
B.4	创建系统部署模型	320
附录 C	网上购物商店	321
C.1	需求分析	321
C.2	系统建模	322
C.3	创建系统动态模型	324
C.4	创建系统部署模型	326
附录 D	汽车租赁系统	328
D.1	需求分析	328
D.2	系统建模	328
D.3	创建系统动态模型	330
D.4	创建系统部署模型	333
附录 E	超市管理系统	335
E.1	需求分析	335
E.2	系统建模	335
E.3	创建系统动态模型	338
E.4	创建系统部署模型	340

# 第 1 章 建模基础知识

近年来，人们对环境工程、生物工程、软件工程等给予了极大的关注，许多高等院校都增设了这些专业，以培养研究这些新兴学科的专业人才。然而，由于人们对这些新兴工程学科认识不足，还处在一个艰难的探索阶段，如软件工程，因此国内许多软件企业对软件工程还不够重视，事实上，软件工程在计算机的发展和应用中的地位非常显著，它对软件产业的形成和发展起着定性的推动作用，是现代信息产业的巨大支柱。本章将对软件工程的目标、过程和原则，以及 UML 等方面的问题和基本概念进行简要介绍，以便读者在进入软件建模之前，对软件工程的一些基本知识有一定的理解。

## 1.1 软件工程概述

软件工程是在 20 世纪 60 年代末期提出的。这一概念的提出，其目的是倡导以工程的原理、原则和方法进行软件开发，以期解决当时出现的“软件危机”。

### 1.1.1 软件工程的产生

软件工程 (Software Engineering) 是一门指导计算机软件系统开发和维护的工程学科，涉及到计算机科学、工程管理科学、数学等多学科，其研究范围广泛，主要研究如何应用软件开发的科学理论和工程技术来指导大型软件系统的开发。例如，现代操作系统的开发，如果不采用软件工程的方法是不可能实现的。

在我国加入 WTO 后，大力推广、应用软件工程的开发技术及管理技术，提高软件工程的水平，对促进我国软件产业与国际接轨，推动我国软件产业的迅速发展起到了关键作用。

软件工程的产生和发展是与软件的发展紧密相关的。软件是计算机系统中与硬件相互依存的一部分，是包括程序、数据及其相关文档的完整集合。其中，程序是按事先设计的功能和性能要求执行的指令序列；数据是使程序能正常操纵信息的数据结构；文档是与程序开发、维护和使用有关的图文材料。

自 20 世纪 40 年代中期出现了世界上第一台计算机以后，就有了程序的概念，开始了软件的开发，其后经历了几十年的发展，计算机软件经历了三个发展阶段。

- 程序设计时代 (1946 年 ~ 1956 年)：采用“个体生产方式”，即软件开发完全依赖于程序员个人的能力水平。





- 程序系统时代（1956 年~1968 年）：由于软件应用范围及规模的不断扩大，个体生产已经不能够满足软件生产的需要，一个软件需要由几个人协同完成。在该阶段的后期，随着软件需求量、规模及复杂度的增大，生产作坊的方式已经不能够适应软件生产的需要，出现了所谓的软件危机。
- 软件工程时代（1968 年至今）：该阶段的主要任务是为了克服软件危机、适应软件发展的需要，而采用工程化的生产方式。

20 世纪 60 年代中期以后，需要开发一些大型软件系统。然而软件技术的进步一直未能满足形势发展的需要，在大型软件的开发过程中出现了复杂程度高、研制周期长、正确性难以保证的三大难题。遇到的问题找不到解决办法，致使问题堆积起来，形成了人们难以控制的局面，出现了所谓的“软件危机”。

软件危机（Software Crisis）是指在软件开发和维护中所产生的一系列严重的问题：一是如何开发新的软件，满足用户对软件的需求；二是如何维护数量众多的已有软件。其主要表现如下：

- 用户需求不明确、变更过多。
- 软件成本日益增长。
- 开发进度难以控制。
- 软件质量差。
- 软件维护困难。

软件危机的出现是由于软件的规模越来越大，复杂度不断增加，软件需求量增大。而软件开发过程是一种高密集度的脑力劳动，软件开发的模式及技术不能适应软件发展的需要，致使大量质量低劣的软件涌向市场，有的软件花费了大量的人力物力，却在开发过程中就夭折。例如：IBM 公司的 OS/360 操作系统由 5000 人开发，经费达数亿美元，但软件产品的每个版本均有 1000 多个大大小小的错误，系统根本无法正常运行。

到了 20 世纪 60 年代末期，软件危机已相当严重，这促使计算机领域的科学家们开始探索缓解软件危机的方法。他们提出了“软件工程”的概念，即利用现代工程的原理、技术和方法进行软件的开发、管理、维护和更新。于是，开创了计算机科学技术的一个新的研究领域。1968 年，北大西洋公约组织召开计算机科学会议，由 Fritz Bauer 首次提出了“软件工程”的概念。

软件工程是一门研究如何用系统化、规范化、数量化等工程原则和方法去进行软件的开发和维护的学科。它应用计算机科学、数学及管理科学等原理，借鉴传统工程的原则和方法创建软件以达到提高质量、降低成本的目的。其中，计算机科学和数学用于构造模型与算法；管理科学用于计划、资源、质量和成本等的管理。软件工程包括两方面内容：软件开发技术和软件项目管理。

- 软件开发技术包括软件开发方法学、软件工具和软件工程环境。
- 软件项目管理包括软件度量、项目估算、进度控制、人员组织、配置管理、项目计划等。



## 1.1.2 软件工程的内容和基本原理

软件工程研究的主要内容是软件开发技术和软件开发管理两个方面。在软件开发技术中,主要研究软件工程方法、软件工程过程、软件开发工具和环境。

- 软件工程方法为软件开发提供了“如何做”的技术。它包括了多方面的任务,如项目计划与估算、软件系统需求分析、数据结构、系统总体结构的设计、算法的设计、编码、测试以及维护等。软件工程方法常采用某种特殊的语言或图形的表达方法,以及一套质量保证标准。
- 软件工程过程是将软件工程的方法和工具综合起来以达到合理、及时地进行计算机软件开发的目的。软件工程过程定义了软件工程方法的使用顺序、要求交付的文档资料,为保证质量和协调变化所需要的管理及软件开发各个阶段完成的里程碑。
- 软件开发工具和环境为软件工程方法提供了自动的或半自动的软件支撑环境。目前,人们已经开发出了许多软件工具来支持上述的软件工程方法,而且已经有人把诸多软件工具集成起来,使得一种工具产生的信息可以被其他的工具所使用,从而建立起一种称为计算机辅助软件工程(CASE)的软件开发支撑系统。CASE将各种软件工具、开发机器和一个存放开发过程信息的工程数据库组合起来形成一个软件工程环境。

对于软件工程基本原理,早在1983年,B.W.Boehm归纳提出了7条基本原则。

### 1. 利用分阶段的生命周期计划严格管理

有人统计过,在不成功的软件项目中有50%左右是由于计划不周造成的。应该把软件生命周期划分为若干阶段,并制定出相应的切实可行的计划,严格按照计划对开发和维护进行管理。B.W.Boehm认为,应制定和严格执行6类计划:项目概要计划、里程碑计划、项目控制计划、产品控制计划、验证计划、运行维护计划。

### 2. 坚持进行阶段评审

设计的错误占软件错误的63%,编码错误只占37%,而且在后期纠正错误的代价非常高,因此,必须严格坚持阶段评审,及早发现和纠正错误。

### 3. 实行严格的产品质量控制

在现实中由于外部原因或要求对需求等进行修改是难免的,但必须有严格的管理制度和措施。

### 4. 采用现代程序设计技术和软件工程技术

多年来,人们一直致力于研究新的程序设计技术,例如结构化程序分析(Structured Analysis)和结构化设计(Structured Design)等。