

Mc  
Graw  
Hill  
Education

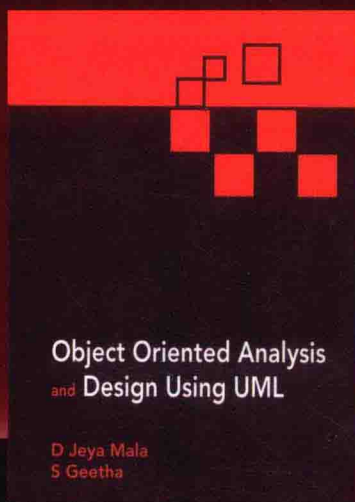
清华计算机图书·译丛

Object Oriented Analysis and Design Using UML

# UML面向对象 分析与设计

D Jeya Mala S Geetha 著

马恬煜 译



Mc  
Graw  
Hill  
Education

清华大学出版社

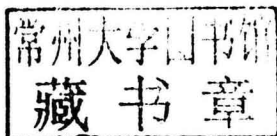
清华计算机图书·译丛

Object Oriented Analysis and Design Using UML

# UML面向对象 分析与设计

D Jeya Mala S Geetha 著

马恬煜 译



清华大学出版社  
北京

D Jeya Mala, S Geetha  
**Object Oriented Analysis and Design Using UML, First Edition**

ISBN: 9781259006746

Copyright ©2013 by McGraw-Hill Education.

All Rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including without limitation photocopying, recording, taping, or any database, information or retrieval system, without the prior written permission of the publisher.

This authorized Chinese translation edition is jointly published by McGraw-Hill Education and Tsinghua University Press Limited. This edition is authorized for sale in the People's Republic of China only, excluding Hong Kong, Macao SAR and Taiwan.

Copyright © 2017 by McGraw-Hill Education and Tsinghua University Press Limited.

版权所有。未经出版人事先书面许可，对本出版物的任何部分不得以任何方式或途径复制或传播，包括但不限于复印、录制、录音，或通过任何数据库、信息或可检索的系统。

本授权中文简体字翻译版由麦格劳-希尔（亚洲）教育出版公司和清华大学出版社有限公司合作出版。此版本经授权仅限于中华人民共和国境内（不包括中国香港、澳门特别行政区和中国台湾地区）销售发行。  
版权©2017 由麦格劳-希尔（亚洲）教育出版公司与清华大学出版社有限公司所有。

著作权合同登记号 图字：01-2017-6577

本书封面贴有 McGraw-Hill 防伪标签，无标签者不得销售。

版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

#### 图书在版编目（CIP）数据

UML面向对象分析与设计 / (印) D Jeya Mala (D. 耶娅·马拉), (印) S Geetha (S. 吉萨) 著; 马恬煜译.  
—北京: 清华大学出版社, 2018

(清华计算机图书译丛)

书名原文: Object Oriented Analysis and Design Using UML

ISBN 978-7-302-49213-9

I. ①U… II. ①D… ②S… ③马… III. ①面向对象语言—程序设计—高等学校—教材  
IV. ①TP312.8

中国版本图书馆CIP数据核字（2017）第319708号

责任编辑: 袁勤勇  
封面设计: 常雪影  
责任校对: 时翠兰  
责任印制: 丛怀宇

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社 总 机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, [c-service@tup.tsinghua.edu.cn](mailto:c-service@tup.tsinghua.edu.cn)

质 量 反 馈: 010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 刷 者: 北京富博印刷有限公司

装 订 者: 北京市密云县京文制本装订厂

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 22.75

版 次: 2018年6月第1版 字 数: 553千字

印 数: 1~1500 印 次: 2018年6月第1次印刷

定 价: 89.00元

产品编号: 073370-01

# 前 言

## 一、简介

在当今世界，计算机化系统和软件已经变得密不可分，后者成为驱动前者的工具。遗憾的是，软件并不能被所有人轻松利用。人们需要编写它、理解它、分析它、使用它和更新它来应对未来版本的变更。正是这个人类和编程世界的交织把建模复杂系统摆到了构造层面，其级别要高于“普通”的编程语言。这也提出了对方法学的需求，以此引导软件工程师和编程人员处理建模过程本身。

为了设计一个高级建模方法，需要一种专家图表式推理。单靠图表或流程图的帮助来进行解释并不够。图表语言都很重要，可以通过对验证和分析的计算机化支持来理解它们。

多年来，结构化分析（Structured Analysis, SA）和面向对象分析（Object Oriented Analysis, OOA）被认为在高级方法中占主导地位。SA 在 20 世纪 70 年代由 DeMarco、Yourdo 等人提出，它基于以图形的方式将传统的过程化编程概念“提升”到建模级别。这就要求通过功能分解和信息流来建模系统结构，并用数据流图进行描述。面向对象分析与设计（也称为面向对象建模）在 20 世纪 80 年代后期问世。系统结构的基本思想是以图形的方式将面向对象编程中的概念“提升”到建模级别。

现在，面向对象方法学已经成功地组合成通用的 UML，目的是想结合各种面向对象建模方法的优点。这是一个突破性的成果，因为越来越多的软件工程师现在声称有更多种类的软件可以以面向对象方式开发和使用。

为了捕获系统结构，UML 对类和对象采用了一种图表化语言，该语言基于实体关系方法。对于早期阶段行为分析，它推荐了使用用例和序列图。而另一方面，对于行为的完整构造规范，它采用了状态图，这样可以在上述可执行对象建模工作中进行修改。

伴随着近年来 UML 的流行浪潮，将会有大量书籍、报纸、报告、研讨会和工具来描述、利用和说明 UML。在目前市面上的众多图书中，本书凭借其组织和呈现思想的方式而独树一帜。

## 二、关于本书

本书通过 UML 解释面向对象分析与设计的主题，同时通过对主题的创新方法来吸引读者的兴趣。本书为读者提供了一些可以从理论上掌握并应用到现实生活中的章节内容。系统分析和设计是在面向对象方法环境中必须采取的活动。前 4 章在此上下文中基于面向开发范例进行信息系统的开发。

本书每一章的开头都有一个概念图以及学习目标，这清晰说明在后续章节中会介绍的主题，同时也指出它们的重要性。这是一个额外的优势，因为它为面向对象的主题打下了

一个坚实的基础。章末的练习分为“牢记”“应用”“分析”“实践”部分，用来测试学生的理解程度。第 10 章给出的已解决的示例既可帮助学生，也可帮助教师。他们所掌握的知识可应用到所有应用程序领域。提供的示例并不是相近的示例集，而是选自不同的公司项目，所以属于实践知识部分。因此，提供的解决方案可以用于任意相似类型的应用程序。这些特色使得本书成为一种有关面向对象软件开发方法的全面而有帮助的资料。

本书对于学习“计算机科学”“信息技术”和“计算机应用”课程的学生来说是非常有用的学习资料。它对于想理解业务信息系统如何开发的人或者想知道 UML 工具如何帮助设计的编程人员来说也很有用。每天需要与系统开发人员沟通的业务分析师和客户同样可参考本书。

本书介绍了 UML 图最实用的部分、每个图的作用、绘制它们的符号以及如何使用现实生活中的案例研究来应用它们。UML 为业务和技术方面的专业人员提供了一个共同基础。在合适的地方提供的示例和案例研究对于建立通用语言非常实用和有帮助。它们还通过分解图表和清晰解释为什么和如何使用图表描述关键业务系统。

### 三、本书特色

- 易于阅读、引人入胜和一致的语言。
- 深入覆盖了所有重要的主题，如软件开发生命周期、对象和类的识别、面向对象的图、UML 及其应用。
- 案例研究、示例和现实情形中应用的实用方法。
- 每章章首的概念图，使得学生可以形象地了解概念的层次结构。
- 独特的分级练习，分为“牢记”“理解”“应用”“分析”部分。
- 解释了如何将伪代码 UML 示例转换为编程示例。

### 四、本书组织结构

本书由 10 章组成。第 1 章介绍各种软件生命周期开发，特别是开发软件的统一软件过程的重要性以及相关概念。第 2 章指出 OOA 区别于 SA 之处、选择面向对象方法的根据以及类、抽象和实例。第 3 章解释如何进行现实世界的建模，以及识别系统中类和实例的 OOA 和 OOD 部分概念。第 4 章全面讲述 UML 建模工具，并且与其他可视化建模工具作对比，介绍其对面向对象软件开发的支持。第 5~8 章以系统的方式详细讨论 UML 中可用的各种图表元素，也就是可用的符号，以及每种表示法的用法和应用。所有这些图都通过贯穿全书的案例研究来解释，因此读者可以对所有 UML 图有更好的和完整的理解。第 9 章通过案例研究讨论包图、组件图和部署图。第 10 章介绍 4 个特殊的案例研究，包括桌面应用程序、内联网应用程序、客户机-服务器应用程序以及基于 Web 的应用程序。这些问题是从现实项目中选取出来的，当学生在不同公司实习时可帮到他们。另外，对这些系统的完整面向对象分析与设计（包括 UML 图集）也一并提供。

## 五、网络资源

学生和老师可通过 [http://www.mhhe.com/jeya\\_mala/ood](http://www.mhhe.com/jeya_mala/ood) 获取下列资源。

- Rational Rose Tool Kit——指导如何用于 UML 图（与书结合在一起）
- 按章节分类的测试
- 参考资料的链接
- 案例研究
- PowerPoint 幻灯片
- 作者视频的链接

作 者

# 目 录

第 1 章 统一软件过程 .....	1
1.1 软件开发过程基础 .....	1
1.1.1 瀑布模型 .....	2
1.1.2 增量模型 .....	3
1.1.3 演化模型 .....	3
1.1.4 敏捷模型 .....	4
1.2 RUP 简介 .....	5
1.3 统一软件过程组件 .....	5
1.4 统一过程模型的生命周期阶段 .....	6
1.4.1 初始阶段 .....	6
1.4.2 细化阶段 .....	7
1.4.3 构造阶段 .....	9
1.4.4 交付阶段 .....	9
1.5 面向对象图在 RUP 中的应用 .....	10
小结 .....	11
多选题 .....	11
练习 .....	13
第 2 章 面向对象开发的理论基础 .....	15
2.1 结构化方法和面向对象方法的对比 .....	16
2.1.1 结构化方法 .....	16
2.1.2 面向对象方法 .....	18
2.2 软件开发过程中的面向对象 .....	19
2.2.1 为什么需要面向对象方法 .....	19
2.2.2 算法和面向对象方法在抽象上的区别 .....	19
2.2.3 算法和面向对象方法在分解上的区别 .....	19
2.2.4 算法和面向对象方法在层次上的区别 .....	20
2.3 面向对象的特点 .....	21
2.4 面向对象中的基本实体 .....	21
2.4.1 什么是类 .....	21
2.4.2 什么是对象 .....	22
2.4.3 属性 .....	23
2.4.4 方法 .....	24
2.4.5 接口 .....	25
2.4.6 包 .....	26

2.4.7	组件 .....	27
2.4.8	访问说明符/可见性控制 .....	27
2.5	面向对象的构造 .....	28
2.5.1	抽象 .....	28
2.5.2	封装 .....	29
2.5.3	信息隐藏 .....	30
2.5.4	层次 .....	31
2.5.5	多态 .....	33
2.5.6	消息传递 .....	34
2.6	选择面向对象开发的因素 .....	35
2.7	案例研究 .....	35
	小结 .....	41
	多选题 .....	42
	练习 .....	44
<b>第3章</b>	<b>面向对象分析与设计中类和对象的识别 .....</b>	<b>45</b>
3.1	问题域的面向对象分析 .....	46
3.2	用于类和对象识别的面向对象分析技术 .....	49
3.2.1	基于抽象的分析 .....	50
3.2.2	实体分析 .....	51
3.2.3	基于场景的分析 .....	51
3.2.4	基于操作的分析 .....	53
3.2.5	基于问题域的分析 .....	53
3.2.6	基于CRC卡的分析 .....	55
3.2.7	基于数据流和控制流的分析 .....	56
3.2.8	基于名词-动词的分析 .....	57
3.3	问题域的面向对象设计 .....	57
3.4	面向对象设计的设计原则 .....	58
3.4.1	开闭原则 (OCP) .....	58
3.4.2	Liskov 替代原则 (LSP) .....	59
3.4.3	依赖性倒置原则 (DIP) .....	59
3.4.4	接口分离原则 (ISP) .....	59
3.4.5	发布重用等效原则 (REP) .....	60
3.4.6	共同封闭原则 (CCP) .....	60
3.4.7	共同重用原则 (CRP) .....	60
3.5	类和对象的识别和细化过程中的设计模式 .....	61
3.5.1	设计模式介绍 .....	61
3.5.2	设计模式——创建型 .....	63
3.5.3	设计模式——结构型 .....	66
3.5.4	设计模式——行为型 .....	68



3.6 OOA 技术的案例研究 .....	72
小结 .....	74
多选题 .....	75
练习 .....	80
<b>第 4 章 UML 建模</b> .....	<b>81</b>
4.1 分析和设计阶段 .....	82
4.1.1 问题领域理解 .....	82
4.1.2 传统的分析和设计方法及模型 .....	82
4.1.3 良好的分析具有的特点 .....	84
4.1.4 传统方法的缺陷 .....	85
4.2 基于 UML 的面向对象分析和设计 .....	86
4.3 可视化建模 .....	86
4.4 图形符号系统 .....	87
4.4.1 Booch 表示法 .....	88
4.4.2 对象管理技术 (OMT) .....	88
4.4.3 统一建模语言 (UML) .....	88
4.5 将 UML 用作一个有效的建模工具 .....	90
4.6 理解 UML 图 .....	91
4.6.1 结构图 .....	92
4.6.2 行为图 .....	92
4.7 对 OOA 和 OOD 的支持 .....	93
4.8 UML 中的众多图表 .....	94
4.9 UML 的范围 .....	95
4.9.1 已解决的问题 .....	95
4.9.2 案例研究 1 .....	95
4.9.3 案例学习 2 .....	96
小结 .....	96
多选题 .....	97
练习 .....	99
<b>第 5 章 面向对象分析——基于场景的模型</b> .....	<b>100</b>
5.1 用例分析 .....	100
5.1.1 用例 .....	100
5.1.2 用例的重要性 .....	101
5.2 主用例图 .....	101
5.3 次用例图 .....	103
5.4 用例图中使用的符号 .....	104
5.4.1 系统 .....	104
5.4.2 参与者 .....	105
5.4.3 用例 .....	106

5.4.4 关系 .....	106
5.5 用例图的目的 .....	110
5.6 如何绘制用例图 .....	111
5.7 案例研究 .....	111
5.8 已解决的问题 .....	118
小结 .....	120
多选题 .....	120
练习 .....	125
<b>第 6 章 面向对象设计——逻辑模型</b> .....	<b>127</b>
6.1 UML 类图 .....	127
6.1.1 类实例 .....	128
6.1.2 类图——结构化的表示 .....	128
6.2 类图中使用的基本符号 .....	128
6.2.1 类 .....	128
6.2.2 抽象类 .....	130
6.2.3 标准的类构造型 .....	130
6.2.4 非标准的类构造型 .....	131
6.2.5 类模板 .....	132
6.2.6 接口 .....	133
6.2.7 对象 .....	134
6.2.8 数据类型 .....	134
6.2.9 属性 .....	136
6.2.10 分类器属性 .....	136
6.2.11 关联属性 .....	137
6.2.12 限定符 .....	137
6.2.13 多重性 .....	137
6.2.14 操作 .....	138
6.2.15 抽象操作 .....	140
6.2.16 约束 .....	140
6.2.17 多重性 .....	141
6.2.18 可见性 .....	141
6.2.19 UML 关联 .....	142
6.2.20 关联终端 .....	143
6.2.21 可导航性 .....	144
6.2.22 不推荐的可导航性惯例 .....	144
6.2.23 元数 .....	145
6.2.24 共享聚合和复合聚合 .....	146
6.2.25 关联类 .....	147
6.2.26 链接 .....	147

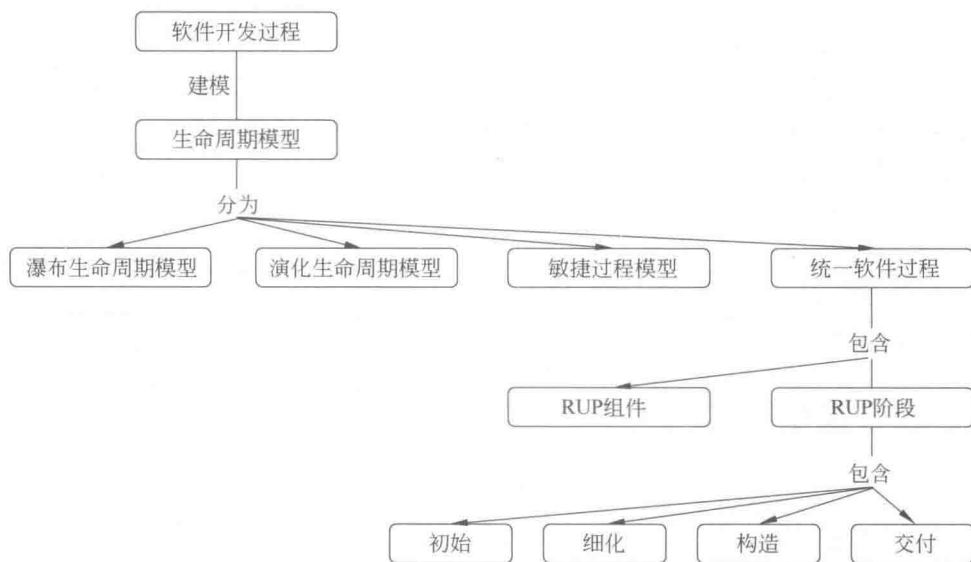
6.2.27	UML 中的泛化 .....	148
6.2.28	继承 .....	148
6.2.29	UML 中的依赖关系 .....	149
6.2.30	使用 .....	150
6.2.31	必需的接口 .....	152
6.2.32	UML 中的抽象 .....	152
6.2.33	实现 .....	153
6.2.34	接口实现 .....	153
6.2.35	跟踪 .....	154
6.3	目的 .....	154
6.4	类图构造指南 .....	154
6.4.1	在结构化视图中构建类图的清单 .....	155
6.4.2	识别和建模类 .....	155
6.4.3	识别和建模关联 .....	156
6.4.4	定义属性 .....	156
6.4.5	列出所需的查询和输入 .....	157
6.4.6	制定查询和输入 .....	157
6.4.7	进行信息分析 .....	158
6.4.8	整合类图 .....	159
6.4.9	验证类图 .....	159
6.5	已解决的案例研究 .....	159
小结	.....	160
多选题	.....	161
练习	.....	163
<b>第 7 章</b>	<b>交互图</b> .....	<b>167</b>
7.1	交互图 .....	167
7.1.1	序列图 .....	168
7.1.2	协作图 .....	182
7.1.3	决定交互图的范围 .....	185
7.2	已解决的案例研究 .....	186
小结	.....	188
多选题	.....	188
练习	.....	190
<b>第 8 章</b>	<b>面向对象设计——行为设计</b> .....	<b>194</b>
8.1	状态图 .....	194
8.1.1	状态图中使用的符号 .....	195
8.1.2	状态图的作用 .....	199
8.1.3	绘制状态图的指导原则 .....	199
8.1.4	状态图的应用 .....	199

8.1.5	案例研究——手机监控软件.....	200
8.2	活动图.....	202
8.2.1	活动图的类型.....	203
8.2.2	活动图中使用的符号.....	203
8.2.3	活动图的作用.....	210
8.2.4	绘制活动图的指导原则.....	210
8.2.5	活动图、序列图和状态图之间的对比.....	211
8.2.6	案例研究——手机监控软件.....	212
8.3	已解决的案例研究.....	214
小结	.....	216
多选题	.....	217
练习	.....	220
<b>第9章</b>	<b>面向对象设计——物理图</b> .....	<b>221</b>
9.1	包图.....	222
9.1.1	包图中使用的表示法.....	222
9.1.2	包图的作用.....	226
9.1.3	绘制包图的指导方针.....	226
9.1.4	案例研究.....	227
9.2	组件图.....	228
9.2.1	组件图中使用的表示法.....	228
9.2.2	UML 2.0 中的高级表示法.....	229
9.2.3	组件图的作用.....	231
9.2.4	绘制组件图的指导方针.....	231
9.2.5	案例研究——组件图.....	232
9.3	部署图.....	235
9.3.1	部署图中的表示法.....	235
9.3.2	部署图的作用.....	238
9.3.3	绘制部署图的指导方针.....	238
9.4	案例研究.....	239
9.5	已解决的案例研究.....	240
小结	.....	241
多选题	.....	242
练习	.....	247
<b>第10章</b>	<b>面向对象开发示例</b> .....	<b>248</b>
10.1	在现实世界中应用类和对象.....	248
10.2	IT 服务台.....	248
10.2.1	产品概述.....	248
10.2.2	项目的范围.....	249
10.2.3	组件.....	249

10.2.4	基于角色的特定要求.....	250
10.2.5	产品功能.....	251
10.2.6	用例图.....	251
10.2.7	类图.....	261
10.2.8	序列图.....	263
10.2.9	活动图.....	274
10.3	保险索赔管理系统.....	283
10.3.1	描述.....	283
10.3.2	功能需求.....	283
10.3.3	用例图.....	284
10.3.4	使用 UML 的逻辑设计.....	287
10.3.5	使用 UML 的行为设计.....	290
10.4	workflow 管理系统.....	296
10.4.1	描述.....	296
10.4.2	产品功能.....	296
10.4.3	功能需求.....	297
10.4.4	用例图.....	298
10.4.5	使用 UML 的逻辑设计.....	303
10.4.6	行为模型——使用 UML 的交互图.....	304
10.4.7	使用 UML 的行为设计.....	312
10.4.8	包图.....	321
10.4.9	组件图.....	323
10.4.10	部署图.....	323
10.5	桌面应用程序/工具开发——PreClean 工具.....	324
10.5.1	描述.....	324
10.5.2	功能需求.....	325
10.5.3	模块描述.....	325
10.5.4	用例图.....	328
10.5.5	使用 UML 的逻辑设计.....	331
10.5.6	使用 UML 的行为设计.....	336
10.5.7	包图.....	341
10.5.8	组件图.....	342
10.5.9	部署图.....	342
	术语表.....	344

# 第 1 章 统一软件过程

概念图



## 学习目标

在学完本章之后，读者将能够：

- 理解统一软件过程（Rational Unified Process, RUP）应用于软件开发的各个生命周期阶段。
- 了解统一建模语言（Unified Modeling Language, UML）图在每个生命周期阶段中的应用。
- 理解 RUP 的优点。
- 学习面向对象的开发过程。

## 1.1 软件开发过程基础

软件开发过程是指软件开发生命周期中明确定义的一系列活动的集合。通常情况下，开发过程首先制订计划（一般作为 0 阶段活动）。接着是需求获取过程，通过对客户或客户代表进行采访、发放调查问卷、讨论、频繁会晤等方式，更好地理解问题领域。一旦需求收集完毕，就进入分析阶段进行分析，发现和更正不全面的、理解错误的和缺失的需求信息。分析阶段的工作产品就是软件需求规范（Software Requirements Specification, SRS）文档，开发团队和客户必须对 SRS 文档进行审核并达成一致。

SRS 文档作为接下来设计、编码和测试阶段的基础。设计阶段包括了应用

档的开发，这些应用可以由结构化分析与设计（Structured Analysis and Design, SAD）开发，或者使用面向对象分析与设计（Object Oriented Analysis and Design, OOAD）开发。然后进入编码阶段，将 SRS 和设计文档中指定的软件和硬件要求作为基准。

如果产品使用敏捷方法或演化开发模型来开发，那么开发就是以迭代和增量的方式完成的，主要重点是变更管理和客户的动态输入。一旦产品由开发人员开发和进行了单元测试，系统由测试团队进行了测试，并通过了用户验收测试被客户接受，该产品就部署在客户现场。

接下来进入维护过程，重点是保留已部署软件的功能，根据用户需求更新现有功能，并更正客户现场可能发生的问题。

通过遵循工程方法开发软件，过去的一些研究人员提出了几种模型。软件开发生命周期模型的广泛类别有：

- ① 瀑布模型
- ② 增量模型
- ③ 演化开发模型
- ④ 敏捷模型

### 1.1.1 瀑布模型

这是由 Winston Royce 于 1970 年提出的线性顺序开发模型。瀑布模型的流程包括沟通和规划、需求分析、设计、编码、测试和维护几个顺序阶段。前一阶段的结果被用作下一阶段的输入。瀑布模型涉及的步骤如下：

- ① 向客户和企业代表收集需求。
- ② 制订项目计划，以估算项目的成本和时间表。
- ③ 根据收集的需求准备 SRS 文档。
- ④ 设计一套设计文档作为编码蓝图。
- ⑤ 根据设计文档开发软件模块或单元。
- ⑥ 集成模块并进行测试，以确保满足客户的所有需求。
- ⑦ 使用用户文档和手册部署完整的测试软件交付给客户。

#### 1. 优点

如果事先知道了需求，则易于以阶段化的线性顺序开发产品。

#### 2. 缺点

① 在实际应用中，由于客户可能无法在初始阶段完全明确地陈述自己的需求，因此开发人员可能无法提前了解明确的需求。

② 由于产品仅在生命周期结束时可供客户使用，客户的反馈信息来得太迟，因此可能无法在以后的阶段解决错误或任何需求下滑的情况。

③ 重新进行分析和测试可能会导致成本增加和时间延迟。

### 3. 哪些类型的项目可以使用它来开发

可以使用这种模型开发提前知道需求的项目，如下所示：

- ① 语言编译器、解释器
- ② 操作系统
- ③ 设备驱动

## 1.1.2 增量模型

在这种模型中，整个软件被分解为满足各种功能的增量。客户需要对其阐述的要求进行优先级分配，第一个增量将根据最高优先级要求开发。一旦增量的开发开始，就不能再对该增量进行进一步的修改。以第一增量收集的反馈为基础进行对下一组增量的开发，最终增量将提供客户所规定的完整功能。

### 1. 优点

- ① 项目的增量交付是基于系统功能的。
- ② 客户的定期反馈可以根据增量进行。
- ③ 随着高优先级功能首先被开发，项目整体的功能风险降低了。

### 2. 缺点

- ① 由于规范不具体，因此在每个增量之后不可能进行完整的系统功能测试。
- ② 这种以向客户交付增量为基础的软件变化频繁，因此软件没有很好的架构。另外，项目的开销和时间都会增加。

### 3. 哪些类型的项目可以使用它来开发

需要间歇式运行软件的应用程序遵循此模型。可以使用这个模型开发的一些应用程序有：

- ① 客户机-服务器应用程序，如医院管理系统、库存管理系统和保险管理系统。
- ② 专门用于内联网的 Web 应用程序，即在某个组织内运行的应用程序。

## 1.1.3 演化模型

这是一种增量式、迭代式的模型，其中包含了软件开发生命周期中描述的所有生命周期阶段。除此以外，该模型在每个演化周期结束时都会有用户反馈，下一个螺旋包含下一个周期的反馈信息。演化模型最重要的特点是风险评估。如果用户或客户在某一个级别上满意了，则项目经理就会终止该项目，并且这个项目已经明确地完成了它的每个阶段任务。由于用户参与了开发过程的整个生命周期，因此项目很少会有失败。演化模型包括以下几种类型：

- ① 螺旋模型
- ② 原型模型
- ③ 并行开发模型



## 1. 优点

- ① 风险分析。
- ② 用户参与全部的生命周期过程。
- ③ 在每次螺旋结束时会进行需求的验证。

## 2. 缺点

- ① 如果客户对结果不满意，那么所需的迭代次数将会大大增加，从而导致不可控的开发情况。
- ② 风险评估应严格执行，否则会导致软件的严重失败。

## 3. 哪些类型的项目可以使用它来开发

需要进行适当的风险分析和间歇式软件交付的应用程序遵循此模型。可以使用这种模型开发的一些应用程序有：

- ① 风险很高的硬实时应用程序，如病人监护系统、原子反应堆系统、原子能发电厂的温度监测系统。
- ② 具有高风险概率的软实时系统，如银行应用程序、金融应用程序、天气监测系统、企业资源计划（Enterprise Resource Planning, ERP）应用程序。

### 1.1.4 敏捷模型

敏捷模型侧重于与客户的持续合作，因为变化是不可避免的。考虑到客户不断变化的需求，敏捷过程模型支持客户/客户代表/用户作为开发阶段的一部分。随着用户故事的开发，开发人员将软件持续交付给客户，并根据客户提供的更改需求对软件进行持续更新。开发人员和客户之间的互动交流在敏捷模型中起着至关重要的作用，他们通过项目经理进行频繁的大脑风暴会议，所有的利益相关者都会讨论其关于软件产品质量交付的想法。代码也是通过应用重构技术开发的，这些技术有助于简化和扩展软件。

## 1. 优点

- ① 非常适合与客户交流频繁的中小型项目。
- ② 高度积极的小型开发团队使软件开发得更快。
- ③ 具有高度凝聚力的杰出团队可以根据客户的评价频繁地更新软件。
- ④ 测试驱动的开发有助于通过确保客户在整个开发过程中的需求来提高软件的质量。
- ⑤ 由于它强调最终产品，因此永远不会忽略软件的功能。

## 2. 缺点

- ① 由于敏捷模式不强调文档，不断变化的客户需求会导致频繁的更改，因此它不适合大型项目。
- ② 技术经验丰富的人才是敏捷开发团队的关键要求。如果缺少这一因素，将会导致不断地再编码和再修复。
- ③ 让客户、开发人员和测试人员一起开发每个用户故事的代价非常昂贵，如果没有正