

面向对象的思考过程（原书第4版）



[面向对象的思考过程（原书第4版）_下载链接1](#)

著者:[美] Matt Weisfeld

出版者:机械工业出版社/华章图书

出版时间:2016-11-1

装帧:平装

isbn:9787111553083

〔简介〕

本书是一部独具特色的面向对象技术著作。书中结合代码示例生动透彻地讲述了面向对象思想的精髓，让读者真正学会以对象方式进行思考。本书共15章，在结构上可以分为两部分：第一部分包括第1~10章，探讨面向对象的基本概念，比如封装、多态、继承

等；第二部分包括第11~15章，专注于将这些概念应用于一些通用的面向对象技术中，并结合了一些实用的应用程序。

本书内容精炼，示例简单明了，适合各层次面向对象开发人员阅读，也是高校相关专业面向对象课程的理想教学参考书。

〔译序〕

很多IT从业人员进入这个行业都是从学习一门编程语言开始的。对于编程，我们往往过于关注语言的语法细节，反而忽略了其背后的设计理念。面向对象的思考过程就是一个非常优秀的设计理念。它可以独立于语言存在。如果你熟练掌握了面向对象的思考过程，那么就可以轻松地在不同的面向对象的语言之间切换。

本书透彻地阐述了面向对象这一概念。作者Matt在书中反复强调学习面向对象的思考过程优于学习任何编程语言或工具。事实上，他也是这么做的。Matt阐述了面向对象的三要素：继承、封装、多态，并且自己加上了第四个要素：组合。关于组合，Matt不惜篇幅做了大量的讲解，并且列举了很多通俗易懂的例子，这也是本书的一大特色。

Matt也纠正了人们的一些普遍误解，比如面向对象的范式与面向过程的范式并不是完全对立的关系。而且在应用面向对象的设计和开发时，Matt也讲解了不少如何与遗留系统集成的技巧。同时，Matt也简要介绍了UML这个建模利器。为了不混淆重点，他把介绍UML的章节放置在很靠后的位置。因为他明白，先了解面向对象的各项概念是最重要的。

我虽然拥有多年的从业经验，但是再看本书时仍然有不少收获。其实自从我接触了函数式编程，就渐渐成为函数式编程的拥趸。我会时不时地“鼓吹”函数式编程范式的好处，顺便“贬低”一下面向对象编程。但同时我也有个疑问，既然函数式编程这么好，为什么这几年的发展只能算是波澜不惊，而没有掀起大风浪呢？读了本书之后，我似乎找到了答案。首先面向对象的思考过程更加符合大家对世界的直观感受，毕竟不是每个人都是数学家。函数式编程可以简化很多问题，但它并不能简化所有问题。其次是面向对象的编程范式和函数式编程的范式并不是完全对立的，正如作者讲过，面向过程的编程范式和面向对象的编程范式也不是完全对立的。比如目前流行的一些语言（Scala、Go等）都具备函数式的特点，也兼具面向对象的特点（只不过它们的面向对象的机制与传统的方式有所不同）。所以无论你喜欢哪种编程范式，了解彼此的不同之处是至关重要的。而本书则是了解面向对象范式的优秀书籍。

本书已经更新到了第4版。从本书长达10多年的跨度来看，面向对象范式经久不衰。Matt也适时地在新版中加入了一些新的主题，比如可移植数据、分布式系统、Web服务等。Matt不仅阐述了这些技术，还讲述了它们的前世今生。这样可以帮助读者更加充分地了解技术的演化之路。

无论你是否面向对象编程的经验，本书都适合你作为面向对象思考的旅程开端。最后，希望本书能给大家带来超凡的阅读体验。

作者介绍:

目录: 译者序

作者简介

前言

第1章 面向对象的概念简介 1

1.1 基本概念 1

1.2 对象及遗留系统 2

1.3 过程式编程与面向对象编程	3
1.4 由面向过程开发过渡到面向对象开发	5
1.4.1 过程式编程	5
1.4.2 面向对象编程	6
1.5 究竟什么是对象	6
1.5.1 对象数据	6
1.5.2 对象行为	7
1.6 究竟什么是类	10
1.6.1 创建对象	10
1.6.2 属性	11
1.6.3 方法	11
1.6.4 消息	12
1.7 使用类图作为可视化工具	12
1.8 封装和数据隐藏	12
1.8.1 接口	13
1.8.2 实现	13
1.8.3 接口/实现范式的一个真实示例	14
1.8.4 接口/实现范式的模型	14
1.9 继承	15
1.9.1 超类和子类	16
1.9.2 抽象	16
1.9.3 is-a关系	17
1.10 多态	18
1.11 组合	21
1.11.1 抽象	21
1.11.2 has-a关系	21
1.12 结语	21
1.13 本章中使用的示例代码	22
1.13.1 C#.NET版本的TestPerson类	22
1.13.2 C#.NET版本的TestShape类	23
第2章 如何以面向对象的方式进行思考	25
2.1 清楚接口和实现之间的区别	26
2.1.1 接口	27
2.1.2 实现	27
2.1.3 一个接口/实现示例	28
2.2 使用抽象思维设计接口	31
2.3 尽可能提供最小化的用户接口	32
2.3.1 确定用户?	33
2.3.2 对象行为	33
2.3.3 环境约束	34
2.3.4 识别公共接口	34
2.3.5 识别实现	34
2.4 结语	35
2.5 引用	35
第3章 高级的面向对象概念	36
3.1 构造函数	36
3.1.1 什么是构造函数调用	37
3.1.2 构造函数中包含什么	37
3.1.3 默认构造函数	37
3.1.4 使用多个构造函数	38
3.1.5 设计构造函数	41
3.2 错误处理	41
3.2.1 忽略问题	42
3.2.2 检查问题并中止应用程序	42

3.2.3 检查问题并试图恢复	42
3.2.4 抛出异常	43
3.3 作用域的重要性	45
3.3.1 局部属性?	45
3.3.2 对象属性	46
3.3.3 类属性	48
3.4 操作符重载	49
3.5 多重继承	49
3.6 对象操作	50
3.7 结语	51
3.8 引用	51
3.9 本章中使用的示例代码	51
第4章 类的剖析	53
4.1 类名	53
4.2 注释	55
4.3 属性	55
4.4 构造函数	56
4.5 访问器	58
4.6 公共接口方法	60
4.7 私有实现方法	60
4.8 结语	61
4.9 引用	61
4.10 本章中使用的示例代码	61
第5章 类设计指导	63
5.1 对现实世界系统建模	63
5.2 识别公共接口	64
5.2.1 最小化公共接口	64
5.2.2 隐藏实现	65
5.3 设计健壮的构造函数（以及析构函数）	65
5.4 在类中设计错误处理	66
5.4.1 使用注释给类加上文档	66
5.4.2 构造可以合作的对象	67
5.5 设计时请考虑重用	67
5.6 设计时请考虑扩展性	67
5.6.1 使用描述性的名称	67
5.6.2 抽象不可移植的代码	68
5.6.3 提供一种方式来复制和比较对象	68
5.6.4 保持尽可能小的作用域	69
5.6.5 类的职责与自身高度相关	70
5.7 设计时请考虑可维护性	71
5.7.1 在开发过程中使用迭代	71
5.7.2 测试接口	72
5.8 使用对象持久化	73
5.9 结语	75
5.10 引用	75
5.11 本章中使用的示例代码	75
第6章 使用对象进行设计	77
6.1 设计指导	77
6.1.1 提供正确的分析	79
6.1.2 编写工作陈述文档	80
6.1.3 收集需求	80
6.1.4 开发用户接口的原型	81
6.1.5 识别类	81
6.1.6 确定每个类的职责	81

6.1.7 确定类之间如何协作	81
6.1.8 创建类模型来描述系统	81
6.1.9 建立用户接口原型	82
6.2 对象包装	82
6.2.1 结构化代码	83
6.2.2 包装结构化代码	84
6.2.3 包装不可移植的代码	85
6.2.4 包装已有类	86
6.3 结语	87
6.4 引用	87
第7章 精通继承和组合	88
7.1 重用对象	88
7.2 继承	89
7.2.1 通用和特例	91
7.2.2 设计决策	92
7.3 组合	93
7.4 为什么封装是面向对象的本质	95
7.4.1 继承如何减弱封装	96
7.4.2 关于多态的一个具体例子	97
7.4.3 对象职责	98
7.4.4 抽象类、虚方法和协议	101
7.5 结语	102
7.6 引用	103
7.7 本章中使用的示例代码	103
第8章 框架和重用：使用接口和抽象类进行设计	105
8.1 代码：重用还是不重用	105
8.2 什么是框架	106
8.3 什么是契约	107
8.3.1 抽象类	108
8.3.2 接口	110
8.3.3 综合运用	112
8.3.4 编译器佐证	114
8.3.5 创建契约	115
8.3.6 系统插接点	117
8.4 一个电子商务示例	117
8.4.1 一个电子商务问题	117
8.4.2 非重用方式	118
8.4.3 电子商务解决方案	119
8.4.4 UML对象模型	120
8.5 结语	124
8.6 引用	124
8.7 本章中使用的示例代码	124
第9章 创建对象及面向对象设计	128
9.1 组合关系	129
9.2 分阶段构建	129
9.3 组合类型	131
9.3.1 聚合	131
9.3.2 联合	132
9.3.3 同时使用联合和聚合	133
9.4 避免依赖	133
9.5 基数	134
9.5.1 多个对象联合	136
9.5.2 可选的联合	137
9.6 一个综合性示例	137

9.7 结语	138
9.8 引用	138
第10章 创建对象模型	139
10.1 什么是UML	139
10.2 类图结构	140
10.3 属性和方法	141
10.3.1 属性	142
10.3.2 方法	142
10.4 访问符号	142
10.5 继承	143
10.6 接口	145
10.7 组合	145
10.7.1 聚合	145
10.7.2 联合	146
10.8 基数	147
10.9 结语	148
10.10 引用	149
第11章 对象与可移植数据：XML和JSON	150
11.1 可移植数据	150
11.2 XML	152
11.3 XML与HTML	152
11.4 XML和面向对象的语言	153
11.5 在企业间共享数据	154
11.6 使用DTD验证文档	155
11.7 将DTD集成到XML文档中	156
11.8 使用层叠样式表	161
11.9 JavaScript对象标记	163
11.10 结语	167
11.11 引用	167
第12章 持久化对象：序列化、封送及关系型数据库	168
12.1 持久化对象基础	168
12.2 将对象保存到平面文件中	169
12.2.1 序列化文件	170
12.2.2 再次讨论实现和接口	172
12.2.3 为什么不保存方法	173
12.3 序列化过程中使用XML	173
12.4 写入关系型数据库	176
12.5 结语	179
12.6 引用	179
12.7 本章中使用的示例代码	179
第13章 Web服务、移动应用及混合应用中的对象	183
13.1 分布式计算的演进	183
13.2 基于对象的脚本语言	184
13.3 JavaScript验证示例	186
13.4 网页中的对象	189
13.4.1 JavaScript对象	189
13.4.2 网页控制器	191
13.4.3 声音播放器	192
13.4.4 电影播放器	192
13.4.5 Flash动画	193
13.5 分布式对象及企业	193
13.5.1 公共对象请求代理体系结构	195
13.5.2 Web服务的定义	197
13.5.3 Web服务代码	201

13.5.4 表征状态转移	202
13.6 结语	203
13.7 引用	203
第14章 对象及客户端/服务器端应用程序	204
14.1 客户端/服务器端方式	204
14.2 私有方式	205
14.2.1 序列化对象代码	205
14.2.2 客户端代码	206
14.2.3 服务器端代码	208
14.2.4 运行该私有的客户端/服务器端示例	209
14.3 非私有方式	210
14.3.1 对象定义代码	211
14.3.2 客户端代码	212
14.3.3 服务器端代码	213
14.3.4 运行非私有客户端/服务器端示例	215
14.4 结语	215
14.5 引用	216
14.6 本章中使用的示例代码	216
第15章 设计模式	217
15.1 为什么使用设计模式	218
15.2 Smalltalk的模型/视图/控制器	219
15.3 设计模式类型	220
15.3.1 创建型模式?	220
15.3.2 结构型模式	224
15.3.3 行为型模式	227
15.4 反模式	228
15.5 结语	229
15.6 引用	229
15.7 本章中使用的示例代码?	229
• • • • •	(收起)

[面向对象的思考过程（原书第4版）](#) [下载链接1](#)

标签

面向对象

编程

面向对象思想

软件设计

程序设计

软件开发

架构

计算机

评论

浅尝则止，不够深入

对于初学编程，特别是JAVA的非常有帮助，当年学Java的时候就读到该有多好

用来入门还是不错的

主要使用Java和C#对面向对象程序设计所涉及的概念做了一个梳理，书不厚，还是可以看一看的！

绝对的标题党，xml的介绍能讲一章
js表单验证也能讲一章，面向对象的思想呢？不要被书名骗了，而且这价格都和几大经典一样了，内容肤浅得不行，完全是凑字数，东扯西扯，扯够两进多页。

结构化编程的最大问题，数据是全局的；类有接口（public），方法也有（方法名+参数列表）；引擎属于实现，方向盘属于接口；任何继承的属性都是完全可见的……？什么鬼？看到这句基本可以把书扔了；分析→工作陈诉→需求文档→用户接口原型→识别类及职责→确定类之间的协作→UML创建类模型；继承弱化了父子类间的封装（修改父类会影响子类）；如何通俗的理解框架，office全家桶的用户界面及功能按钮；组合有两种，其中聚合是has，联合是between；可移植数据XML和JSON，主要是为了跨行业垂直传递或本行业（如零售）数据共享；创建型模式：工厂，构造器，原型，单例；结构型：适配器，组合，装饰器等；行为型：解释器，迭代器，中介者，观察者等……本书章节结构有点混乱，重点不足

很棒的入门书。作者特意强调了组合在面向对象里面的地位，很棒的观点。缺点大概就是作者三步一回头，读起来看不到具体的脉络，结果最后啥都介绍了一点

对于面向对象的入门思考还是挺有帮助的，看这书前，对于日期的排序，我就像写C一样，main函数里面直接写函数，看完这书后，对于日期的排序，我就是先写个继承comparator接口的日期类，comparaotor的实现方法中，完成日期排序逻辑，根据日期生成不同的日期实例进行比较排序

作者是一个太极高手，本着蹭蹭不进去的原则，用了200余页的篇幅完美绕开了核心。

[面向对象的思考过程（原书第4版）](#) [下载链接1](#)

书评

这本书是我在Kindle上看的第一本书～～，买了Kindle后将上个世纪出版的一些OO著作都download下来了，准备好好补充一下OO的基础知识。刚看完OO启思录，然后看这本比较薄，然后就开始看这本。现在还没看完，不过有些话想说说。
这本书名为面向对象的思考过程，作者是提出了很多...

我看的是英文版，这是我在英文版下的评论，在这儿再复制一下～～
这本书是我在Kindle上看的第一本书～～，买了Kindle后将上个世纪出版的一些OO著作都download下来了，准备好好补充一下OO的基础知识。刚看完OO启思录，然后看这本比较薄，然后就开始看这本。现在还没看完，不过有...

你正准备学一门面向对象的编程语言？等等... 你知道什么是面向对象吗？听说过？... 那你知道面向对象有三个基本要素吗？不清楚？... 那赶紧读一读这本书吧！... 它如此的简单易读，却让你建立起最基本但最准确的面向对象的思想.. 强烈推荐！！...
