

《精通Swift设计模式》

图书基本信息

书名：《精通Swift设计模式》

13位ISBN编号：9787115433488

出版时间：2016-9-1

作者：[美]亚当·弗里曼

页数：444

译者：丘远乐

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu000.com

《精通Swift设计模式》

内容概要

Swift编程语言已然改变了iOS的世界，为Cocoa开发开启了新的篇章。

本书是帮助读者掌握Swift强大功能的又一利器，由技术畅销书作者Adam Freeman执笔，从设计模型应用的角度探讨Swift语言，抽丝剥茧，层层深入，囊括各类经典模式，并挖掘了那些看似十分简单、实际别有一番奥妙的知识。在本书的指导下，你可以玩转设计模式，用Swift语言开发出优秀的iOS应用。

《精通Swift设计模式》

作者简介

作者简介：

Adam Freeman

资深技术人员，畅销技术图书作家。除本书外还著有《HTML5权威指南》《ASP.NET 4高级程序设计（第4版）》等备受读者推崇的畅销书。

译者简介：

丘远乐

编程爱好者，iOS开发者，偶尔也写点Python。

书籍目录

第一部分 准备工作

第1章 设计模式	2
1.1 将设计模式置于上下文中	2
1.1.1 设计模式简介	3
1.1.2 设计模式的结构	3
1.1.3 量化设计模式	4
1.1.4 问题出现之后使用设计模式	4
1.1.5 设计模式的局限	5
1.2 关于本书	5
1.2.1 读者需要哪些知识背景	5
1.2.2 读者需要哪些软件	5
1.2.3 本书的结构	6
1.2.4 获取示例代码	6
1.3 总结	6
第2章 熟悉Xcode的使用	7
2.1 使用Xcode Playground	7
2.1.1 创建Playground	7
2.1.2 查看变量取值的历史记录	9
2.1.3 使用取值时间轴	11
2.1.4 在Playground中使用UI组件	13
2.2 使用OS X命令行工具项目	14
2.2.1 创建命令行工具项目	14
2.2.2 Xcode的布局	15
2.2.3 新建一个Swift文件	16
2.3 总结	19
第3章 开发SportsStore应用	20
3.1 创建一个简单的iOS应用项目	20
3.1.1 新建项目	21
3.1.2 熟悉Xcode的布局	22
3.1.3 定义数据	23
3.2 构建简单的布局	25
3.2.1 添加基本组件	26
3.2.2 配置Auto Layout	28
3.2.3 测试布局效果	29
3.3 实现与显示库存总量的Label相关的代码	30
3.3.1 创建引用	30
3.3.2 更新界面显示	31
3.4 实现table cell	32
3.4.1 配置自定义table cell和布局	32
3.4.2 配置table cell的布局约束	34
3.4.3 创建table cell类和Outlet	34
3.4.4 实现数据源协议	36
3.4.5 注册数据源	37
3.4.6 测试数据源	38
3.5 处理编辑操作	38
3.6 测试SportsStore应用	41
3.7 总结	42

第二部分 创建型模式

第4章 对象模板模式	44
4.1 准备示例项目	44
4.2 此模式旨在解决的问题	45
4.3 对象模板模式	47
4.4 实现对象模板模式	48
4.5 对象模板模式的优点	49
4.5.1 解耦的好处	49
4.5.2 封装的优点	50
4.5.3 不断演化外部呈现的好处	52
4.6 对象模板模式的陷阱	54
4.7 Cocoa中使用对象模板模式的示例	54
4.8 在SportsStore应用中使用对象模板模式	54
4.8.1 准备示例应用	54
4.8.2 创建Product类	56
4.8.3 使用Product类	58
4.8.4 扩展应用的总结信息	60
4.9 总结	61
第5章 原型模式	62
5.1 此模式旨在解决的问题	62
5.1.1 初始化过程开销大	62
5.1.2 对模板的依赖	64
5.2 原型模式	65
5.3 实现原型模式	65
5.3.1 克隆引用类型	67
5.3.2 浅复制与深复制	70
5.3.3 复制数组	73
5.4 原型模式的优点	75
5.4.1 避免内存开销大的初始化操作	75
5.4.2 分离对象的创建与使用	77
5.5 原型模式的陷阱	83
5.5.1 陷阱一：深复制与浅复制	83
5.5.2 陷阱二：扭曲代码结构	83
5.5.3 陷阱三：不标准的协议	83
5.6 Cocoa中使用原型模式的实例	83
5.6.1 使用Cocoa数组	84
5.6.2 使用NSCopying属性修饰符	86
5.7 在SportsStore应用中使用原型模式	87
5.7.1 准备示例应用	87
5.7.2 在Product类中实现NSCopying协议	87
5.7.3 创建Logger类	88
5.7.4 在View Controller中输出修改日志	89
5.7.5 测试修改	91
5.8 总结	91
第6章 单例模式	92
6.1 准备示例项目	92
6.2 此模式旨在解决的问题	92
6.3 单例模式	96
6.4 实现单例模式	96

6.4.1	快速实现单例模式	96
6.4.2	创建一个传统的单例实现	98
6.4.3	处理并发	100
6.5	单例模式的陷阱	104
6.5.1	泄露缺陷	104
6.5.2	代码文件共享带来的问题	105
6.5.3	并发陷阱	105
6.6	Cocoa中使用单例模式的实例	106
6.7	在SportsStore应用中使用单例模式	106
6.7.1	保护数据数组	107
6.7.2	保护回调	108
6.7.3	定义单例	109
6.8	总结	111
第7章	对象池模式	112
7.1	准备示例项目	112
7.2	此模式旨在解决的问题	113
7.3	对象池模式	113
7.4	实现对象池模式	114
7.4.1	定义Pool类	114
7.4.2	使用Pool类	118
7.5	对象池模式的陷阱	121
7.6	Cocoa中使用对象池模式的实例	121
7.7	在SportsStore应用中使用对象池模式	122
7.7.1	准备示例应用	122
7.7.2	创建一个(伪)服务器	122
7.7.3	创建对象池	123
7.7.4	应用对象池模式	124
7.8	总结	127
第8章	对象池模式的变体	128
8.1	准备示例项目	128
8.2	对象池模式的变体	128
8.2.1	对象创建策略	129
8.2.2	对象复用策略	132
8.2.3	空池策略	136
8.2.4	对象分配策略	146
8.3	对象池模式变体的陷阱	148
8.3.1	期望与现实之间的差距	149
8.3.2	过度利用与低利用率	149
8.4	Cocoa中使用对象池模式变体的示例	149
8.5	在SportsStore应用中使用对象池模式变体	149
8.6	总结	151
第9章	工厂方法模式	152
9.1	准备示例项目	152
9.2	此模式旨在解决的问题	154
9.3	工厂方法模式	156
9.4	实现工厂方法模式	156
9.4.1	定义全局工厂方法	157
9.4.2	使用基类	158
9.5	工厂方法模式的变体	162

9.6	工厂方法模式的陷阱	164
9.7	Cocoa中使用工厂方法模式的示例	164
9.8	在SportsStore应用中使用工厂方法模式	164
9.8.1	准备示例应用	164
9.8.2	实现工厂方法模式	166
9.8.3	应用工厂方法模式	167
9.9	总结	168
第10章	抽象工厂模式	169
10.1	准备示例项目	169
10.2	此模式旨在解决的问题	172
10.3	抽象工厂模式	172
10.4	实现抽象工厂模式	174
10.4.1	创建抽象工厂类	174
10.4.2	创建具体工厂类	174
10.4.3	完善抽象工厂类	175
10.4.4	使用抽象工厂模式	176
10.5	抽象工厂模式的变体	177
10.5.1	隐藏抽象工厂类	177
10.5.2	在具体工厂类中使用单例模式	178
10.5.3	在实现类中使用原型模式	180
10.6	抽象工厂模式的陷阱	185
10.7	Cocoa中使用抽象工厂模式的示例	185
10.8	在SportsStore应用中使用抽象工厂模式	186
10.8.1	准备示例应用	186
10.8.2	定义实现类和协议	186
10.8.3	定义抽象与具体工厂类	187
10.8.4	使用工厂与实现类	188
10.9	总结	189
第11章	建造者模式	190
11.1	准备示例项目	190
11.2	建造者模式	194
11.3	实现建造者模式	195
11.3.1	定义建造者类	195
11.3.2	使用建造者类	196
11.3.3	此模式的影响	196
11.4	建造者模式的变体	200
11.5	建造者模式的陷阱	202
11.6	Cocoa中使用建造者模式的实例	202
11.7	在SportsStore应用中使用建造者模式	203
11.7.1	准备示例应用	203
11.7.2	定义建造者类	203
11.7.3	使用建造者类	204
11.8	总结	205
第三部分	结构型模式	
第12章	适配器模式	208
12.1	准备示例项目	208
12.1.1	创建数据源	209
12.1.2	实现示例应用	210
12.2	此模式旨在解决的问题	211

12.3	适配器模式	212
12.4	实现适配器模式	213
12.5	适配器模式之变体	215
12.5.1	定义一个适配器包装类	215
12.5.2	创建一个双向适配器	216
12.6	适配器模式的缺陷	219
12.7	Cocoa中使用适配器模式的实例	219
12.8	在SportsStore应用中使用适配器模式	219
12.8.1	准备示例应用	220
12.8.2	定义Adapter类	220
12.8.3	使用适配功能	221
12.9	总结	222
第13章	桥接模式	223
13.1	准备示例项目	223
13.2	此模式旨在解决的问题	225
13.3	桥接模式	226
13.4	实现桥接模式	228
13.4.1	实现信息部分	228
13.4.2	实现通道部分	229
13.4.3	创建桥接类	229
13.4.4	增加信息类型与通道类型	230
13.5	桥接模式的变体	233
13.6	桥接模式的陷阱	237
13.7	Cocoa中使用桥接模式的实例	237
13.8	在SportsStore应用中使用此模式	237
13.8.1	准备示例应用	237
13.8.2	理解待解决的问题	237
13.8.3	定义桥接类	238
13.9	总结	239
第14章	装饰器模式	240
14.1	准备示例项目	240
14.2	此模式旨在解决的问题	242
14.3	装饰器模式	244
14.4	实现装饰器模式	244
14.5	装饰器模式的变体	246
14.5.1	创建具有新功能的装饰器	246
14.5.2	合并多个装饰器	249
14.6	装饰器模式的陷阱	251
14.7	Cocoa中使用装饰器模式的实例	251
14.8	在示例项目中使用装饰器模式	251
14.8.1	准备示例应用	252
14.8.2	创建装饰器	252
14.8.3	使用装饰器	253
14.9	总结	253
第15章	组合模式	254
15.1	准备示例项目	254
15.2	此模式旨在解决的问题	256
15.3	组合模式	257
15.4	实现组合模式	257

15.5	组合模式的陷阱	260
15.6	Cocoa中使用组合模式的实例	261
15.7	在SportsStore应用中使用此模式	261
15.7.1	准备示例应用	261
15.7.2	定义组合类	262
15.7.3	使用此模式	263
15.8	总结	264
第16章	外观模式	265
16.1	准备示例项目	265
16.2	此模式旨在解决的问题	267
16.3	外观模式	268
16.4	实现外观模式	269
16.5	外观模式之变体	270
16.6	外观模式的陷阱	271
16.7	Cocoa中使用外观模式的实例	272
16.8	在SportsStore应用中使用外观模式	273
16.8.1	准备示例应用	273
16.8.2	创建外观类	274
16.8.3	使用外观类	274
16.9	总结	275
第17章	享元模式	276
17.1	准备示例项目	276
17.2	此模式旨在解决的问题	278
17.3	享元模式	278
17.4	实现享元模式	279
17.4.1	创建Flyweight协议	279
17.4.2	创建FlyweightImplementation类	280
17.4.3	增加并发保护	281
17.4.4	创建FlyweightFactory类	282
17.4.5	应用享元模式	284
17.5	享元模式之变体	285
17.6	享元模式的陷阱	286
17.6.1	陷阱一：重复的外部数据	286
17.6.2	陷阱二：可变的外部数据	287
17.6.3	陷阱三：并发访问	287
17.6.4	陷阱四：过度优化	287
17.6.5	陷阱五：误用	287
17.7	Cocoa中使用享元模式的实例	287
17.8	在SportsStore应用中使用享元模式	288
17.8.1	准备示例应用	289
17.8.2	创建Flyweight协议和实现类	289
17.8.3	创建享元工厂	289
17.8.4	应用享元模式	290
17.9	总结	290
第18章	代理模式	291
18.1	准备示例项目	291
18.2	此模式旨在解决的问题	292
18.2.1	远程对象问题	292
18.2.2	开销大的问题	292

18.2.3	访问控制问题	293
18.3	代理模式	293
18.3.1	解决远程对象问题	293
18.3.2	解决操作开销大的问题	294
18.3.3	解决访问限制问题	294
18.4	实现代理模式	295
18.4.1	实现远程对象代理	295
18.4.2	实现开销大的操作的代理	297
18.4.3	实现访问限制代理	299
18.5	代理模式之变体	302
18.6	代理模式的陷阱	305
18.7	Cocoa中使用代理模式的实例	306
18.8	在SportsStore应用中使用代理模式	306
18.8.1	准备示例应用	306
18.8.2	定义协议、工厂方法和代理类	306
18.8.3	更新ProductDataStore类	307
18.8.4	发送库存更新	308
18.9	总结	310
第四部分 行为型模式		
第19章 责任链模式 312		
19.1	准备示例项目	312
19.2	此模式旨在解决的问题	314
19.3	责任链模式	315
19.4	实现责任链模式	315
19.4.1	创建并使用责任链	317
19.4.2	使用责任链模式	318
19.5	责任链模式之变体	318
19.5.1	使用工厂方法模式	318
19.5.2	责任链响应反馈	319
19.5.3	通知责任链中的其他链	321
19.6	此模式的陷阱	323
19.7	Cocoa中使用责任链模式的实例	323
19.8	在SportsStore应用中使用责任链模式	323
19.8.1	准备示例应用	323
19.8.2	定义责任链及其链	323
19.9	总结	325
第20章 命令模式 326		
20.1	准备示例项目	326
20.2	此模式旨在解决的问题	327
20.3	命令模式	328
20.4	实现命令模式	328
20.4.1	定义Command协议	329
20.4.2	定义Command实现类	329
20.4.3	使用命令模式	330
20.4.4	添加并发保护	332
20.4.5	使用撤销操作的功能	333
20.5	命令模式之变体	333
20.5.1	创建复合命令	334
20.5.2	将命令当作宏来用	335

20.5.3	将闭包作为命令	337
20.6	命令模式的陷阱	339
20.7	Cocoa中使用命令模式的实例	339
20.8	在SportsStore应用中使用命令模式	340
20.8.1	准备示例项目	340
20.8.2	实现撤销功能	340
20.9	总结	343
第21章	中介者模式	344
21.1	准备示例项目	344
21.2	此模式旨在解决的问题	346
21.3	中介者模式	348
21.4	实现中介者模式	348
21.4.1	定义Meditator类	349
21.4.2	遵循Peer协议	349
21.4.3	实现并发保护	351
21.5	中介者模式之变体	353
21.5.1	将更多的逻辑置于中介者中	353
21.5.2	通用化中介者与其管理的对象之间的关系	355
21.6	中介者模式的陷阱	360
21.7	Cocoa中使用中介者模式的实例	361
21.8	在SportsStore应用中使用中介者模式	363
21.9	总结	363
第22章	观察者模式	364
22.1	准备示例项目	364
22.2	此模式旨在解决的问题	366
22.3	观察者模式	366
22.4	实现观察者模式	367
22.4.1	创建SubjectBase类	368
22.4.2	遵循Subject协议	369
22.4.3	遵循Observer协议	369
22.4.4	使用观察者模式	370
22.5	观察者模式之变体	371
22.5.1	通知的通用化	371
22.5.2	使用弱引用	375
22.5.3	处理生命周期较短的主题对象	376
22.6	观察者模式的陷阱	379
22.7	Cocoa中使用观察者模式的实例	379
22.7.1	用户界面事件	379
22.7.2	观察属性变化	380
22.8	在SportsStore应用中使用此模式	381
22.8.1	准备示例应用	381
22.8.2	应用观察者模式	381
22.9	总结	383
第23章	备忘录模式	384
23.1	准备示例项目	384
23.2	此模式旨在解决的问题	386
23.3	备忘录模式	387
23.4	实现备忘录模式	388
23.4.1	实现Memento类	388

23.4.2	使用备忘录对象	390
23.5	备忘录模式之变体	390
23.6	备忘录模式的陷阱	392
23.7	Cocoa中使用备忘录模式的实例	393
23.8	在SportsStore中使用备忘录模式	396
23.8.1	准备示例项目	396
23.8.2	实现备忘录模式	396
23.9	总结	398
第24章	策略模式	399
24.1	准备示例项目	399
24.2	此模式旨在解决的问题	400
24.3	策略模式	401
24.4	实现策略模式	402
24.4.1	定义策略和环境类	402
24.4.2	使用策略模式	403
24.5	策略模式之变体	404
24.6	策略模式的陷阱	405
24.7	Cocoa中使用策略模式的实例	405
24.7.1	Cocoa框架中基于协议的策略	405
24.7.2	Cocoa中基于选择器的策略	406
24.8	在SportsStore应用中使用策略模式	407
24.9	总结	407
第25章	访问者模式	408
25.1	准备示例项目	408
25.2	此模式旨在解决的问题	410
25.3	访问者模式	412
25.4	实现访问者模式	412
25.4.1	遵循Shape协议	414
25.4.2	创建访问者	416
25.4.3	使用访问者	416
25.5	访问者模式之变体	417
25.6	访问模式的陷阱	417
25.7	Cocoa中使用访问者模式的实例	417
25.8	在SportsStore应用中使用访问者模式	417
25.9	总结	417
第26章	模板方法模式	418
26.1	准备示例项目	418
26.2	此模式旨在解决的问题	420
26.3	模板方法模式	420
26.4	实现模板方法模式	420
26.5	模板方法模式之变体	422
26.6	模板方法模式的陷阱	424
26.7	Cocoa中使用模板方法模式的实例	424
26.8	在SportsStore应用中使用模板方法模式	424
26.9	总结	424
第五部分	MVC模式	
第27章	MVC模式	426
27.1	准备示例项目	426
27.2	此模式旨在解决的问题	426

27.3	MVC模式	426
27.4	实现MVC模式	430
27.4.1	实现通用的代码	431
27.4.2	定义一个框架	432
27.4.3	创建模型	433
27.4.4	实现视图	435
27.4.5	定义控制器	436
27.4.6	完善框架	439
27.4.7	运行应用	439
27.4.8	扩展应用	440
27.5	MVC模式之变体	444
27.6	MVC模式的陷阱	444
27.7	Cocoa中使用MVC模式的实例	444
27.8	总结	444

精彩短评

1、了解一下设计模式，这本书读起来还算简单易懂。

精彩书评

1、作者还在使用Xcode 6，我书看了一半也没有用过protocol extension的特性。所以基本可以断定里面的swift版本是2.0之前。这就很尴尬了。可以感觉到作者是一个有着丰富面向对象经验的老派程序员。里面的设计模式的介绍都没有什么问题，准确清晰，所以道理是没错。问题是，这些模式在推崇面向协议的swift里，并不太合适。格式上，作者的示例代码里说到因为自己是一个多年的老司机，所以不太习惯一些老的代码格式风格，每行后面都要加分号，调用自身的属性方法还是忍不住加self。我觉得这种对style的放纵是不能接受的。所以虽然作者很努力，但我对这本书的定位表示遗憾。如果你想学习设计模式，显然不应该来看这本。swift的实践的话，现在已然更新到3.0的语法，书中介绍swift的部分没有突出的地方。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:www.tushu000.com