

级别	内容	页码
<b>1</b>	<b>设计模式重点</b>	<b>1</b>
1.1	设计模式分类与概述【选择、简答】	1
1.2	模式本质定义、别名与本质	3
1.3	每个设计模式的可变方面	5
1.4	适用场景/应用实例【选择题重点】	6
1.5	文档编辑器的例子	13
1.6	MVC架构涉及到的模式	13
1.7	设计时存在的问题及选择的模式	14
<b>2</b>	<b>设计模式总览【详见附录】</b>	<b>18</b>
2.1	模式整理总结【重要！】	18
2.2	面向对象设计与复用	18
2.3	类间关系与区别【选择、简答】	19
2.4	面向对象设计原则	22
<b>3</b>	<b>创建型模式</b>	<b>24</b>
3.1	简单工厂模式（Simple Factory）	24
3.2	工厂方法模式（Factory Method）	28
3.3	抽象工厂模式（Abstract Factory）	35
3.4	建造者/生成器模式（Builder）	40
3.5	原型模式（Prototype）	45
3.6	单例模式（Singleton）	50
<b>4</b>	<b>结构型模式</b>	<b>57</b>
4.1	适配器模式(Adapter)	57
4.2	桥接模式(Bridge)	64
4.3	组合模式(Composite)	70
4.4	装饰模式(Decorator)	76
4.5	外观模式(Facade)	82
4.6	享元模式(Flyweight)	87
4.7	代理模式(Proxy)	92
<b>5</b>	<b>行为型模式</b>	<b>99</b>
5.1	职责链模式(Chain of Responsibility)	99
5.2	命令模式(Command)	103
5.3	解释器模式(Interpreter)	109
5.4	迭代器模式(Iterator)	110
5.5	中介者模式(Mediator)	116
5.6	备忘录模式(Memento)	119
5.7	观察者模式(Observer)	123
5.8	状态模式(State)	129
5.9	策略模式(Strategy)	133
5.1	模板方法模式(Template Method)	137
<b>6</b>	<b>模式比较/选择【简答】</b>	<b>144</b>
6.1	工厂方法模式 vs 建造者模式	144
6.2	抽象工厂模式 vs 建造者模式	144
6.3	代理模式 vs 装饰模式	144
6.4	装饰模式 vs 适配器模式	145
6.5	装饰模式 vs 桥接模式	145
6.6	中介者模式 vs 外观模式	146
6.7	备忘录模式 vs 命令模式	146
6.8	策略模式 vs 命令模式	147
6.9	状态模式 vs 观察者模式	147
7	策略模式 vs 状态模式	147
7.1	模板方法模式 vs 策略模式	148
7.2	观察者模式 vs 责任链模式	148
7.3	策略模式 vs 桥接模式	149
7.4	包装模式群PK	149
7.5	观察者模式 vs 中介者模式	150

级别	内容	页码
<b>7</b>	<b>模式组合【分析设计】</b>	<b>151</b>
7.1	适配器模式与桥接模式的联用	151
7.2	享元模式与其他模式的联用	153
7.3	命令模式和组合模式联用	153
7.4	备忘录模式和命令模式的联用	153
7.5	备忘录模式和原型模式的联用	153
<b>8</b>	<b>附录</b>	<b>155</b>
<b>8.1</b>	<b>黑皮书</b>	<b>155</b>
8.1.1	Smalltalk MVC中的设计模式	155
8.1.2	1.5 组织编目	156
8.1.3	模式分类	156
8.1.4	1.6 设计模式怎样解决设计问题	157
8.1.5	1.6.1 寻找合适的对象	157
8.1.6	1.6.2 决定对象的粒度	157
8.1.7	1.6.3 指定对象接口	158
8.1.8	1.6.4 描述对象的实现	159
8.1.9	1.6.5 运用复用机制	160
8.1.10	关联运行时候和编译时候的结构	161
8.1.11	1.6.7 设计应支持变化	162
8.1.12	1.7 设计可变方面	165
<b>8.2</b>	<b>附录A 设计模式概述</b>	<b>166</b>
8.2.1	设计模式的诞生与发展	166
8.2.2	设计模式的定义与分类	167
8.2.3	GoF设计模式简介	168
8.2.4	设计模式的优点	169
<b>8.3</b>	<b>附录B 面向对象设计与复用</b>	<b>169</b>
<b>8.4</b>	<b>附录C 类间关系及UML表示</b>	<b>172</b>
8.4.1	1 依赖(dependency)关系	174
8.4.2	2 关联(association)关系	183
8.4.3	3 聚合/聚集(aggregation)关系	190
8.4.4	4 组合/合成(composition)关系	195
8.4.5	5 泛化(generalization)关系	197
8.4.6	6 实现(realization)关系	198
8.4.7	总结	198
<b>8.5</b>	<b>附录D 面向对象设计原则</b>	<b>204</b>
8.5.1	1 开闭原则（OCP）	205
8.5.2	2 里氏替换原则（LSP）	209
8.5.3	3 迪米特原则（LoD）	214
8.5.4	4 单一职责原则（SRP）	218
8.5.5	5 接口分隔原则（ISP）	220
8.5.6	6 依赖倒置原则（DIP）	223
8.5.7	7 组合/聚合复用原则（CARP）	229
<b>8.6</b>	<b>附录E 面向对象知识</b>	<b>232</b>
8.6.1	静态方法	232
8.6.2	类和对象的区别	233
8.6.3	多态	234
8.6.4	抽象类和接口类	234
8.6.5	重定义 vs 重写（cpp）	235
8.6.6	多重继承	236
8.6.7	紧耦合	237
8.6.8	虚函数 vs 纯虚函数【cpp】	237