

<http://www.imooc.com/article/11927>

1.策略模式:

链接: <http://www.imooc.com/article/11475>

题例: 鸭子

原则:

1: 找出应用中可能需要变化之处, 把它们独立起来, 不要和那些不需要变化。

2: 针对接口编程, 而不是针对实现编程。

3: 多用组合, 少用继承。

策略模式定义了算法族, 分别封装起来, 让他们之间可以互相替换, 此模式让算法的变化独立于使用算法的客户。

2.观察者模式:

链接: <http://www.imooc.com/article/11579>

题例: 气象检测应用

原则:

4: 为了交互对象之间松耦合设计而努力。

观察者模式 定义了对象之间的一对多依赖, 这样一来, 当一个对象改变状态时, 它的所有依赖者都会受到通知并自动更新。

3.装饰者模式:

链接: <http://www.imooc.com/article/11631>

题例: 星巴兹咖啡连锁店

原则:

5: 类应该对扩展开放, 对修改关闭。(开放-关闭原则)

装饰者模式动态地将责任附加到对象上。若要扩展功能, 装饰者提供了比继承更有弹性的替代方案。

4.工厂模式:

链接: <http://www.imooc.com/article/11675>

题例: 披萨店

原则:

6: 要依赖抽象, 不要依赖具体类。(依赖倒置原则)

抽象工厂模式提供一个接口, 用于创建相关或依赖对象的家族, 而不需要明确指定具体类。

5.单件模式:

链接: <http://www.imooc.com/article/11735>

题例: 无

原则: 无

单件模式确保一个类只有一个实例，并提供一个全局访问点。

6.命令模式：

链接：<http://www.imooc.com/article/11923>

题例：电器自动化

原则：无

命令模式将“请求”封装成对象，一边使用不同的请求、队列或者日志来来参数化其他对象。命令模式也支持可撤销的操作。

7.适配器模式和外观模式：

链接：<http://www.imooc.com/article/11993>

题例：第一章的鸭子

原则：

7：只和你的密友谈话。（最少知识原则）

适配器模式将一个类的接口，转换成客户期望的另一个接口，适配器让原本接口不兼容的类可以合作无间。

外观模式提供了一个统一的接口，用来访问自系统中的一群接口。外观定义了一个高层接口，让子系统更容易使用。

8.模板方法模式：

链接：<http://www.imooc.com/article/12037>

题例：星巴兹咖啡冲泡

原则：

8：别调用我们，我们会调用你。（好莱坞原则）

模板方法模式在一个方法中定义一个算法的骨架，而将一些步骤延迟到子类中，模板方法使得子类可以在不改变算法结构的情况下，重新定义算法中的某些步骤。

9.迭代器与组合模式：

链接：<http://www.imooc.com/article/12103>

题例：餐厅和煎饼屋合并

原则：

9：一个类只有一个引起变化的原因。（单一责任原则）

迭代器模式提供一种方法顺序访问一个聚合对象中的各个元素，而又不暴露其内部的表示。

组合模式允许你将对象组合成树形节后来表现“整体/部分”层次结构。组合能让客户以一致的方式处理个别对象以及对象组合。

作者：尕小刘

链接：<https://www.imooc.com/article/11927>

来源：慕课网

本文原创发布于慕课网，转载请注明出处，谢谢合作