

单例	Singleton	某个类只能生成一个实例，该类提供了一个全局访问点供外部获取该实例，其拓展是有限多例模式	创建	对象
原型	Prototype	将一个对象作为原型，通过对其进行复制而克隆出多个和原型类似的新实例。	创建	对象
工厂方法	Factory Method	定义一个用于创建产品的接口，由子类决定生产什么产品。	创建	类
抽象工厂	Abstract Factory	提供一个创建产品族的接口，其每个子类可以生产一系列相关的产品。	创建	对象
建造者	Builder	将一个复杂对象分解成多个相对简单的部分，然后根据不同需要分别创建它们，最后构建该复杂对象。	创建	对象
代理	Proxy	为某对象提供一种代理以控制对该对象的访问。即客户端通过代理间接地访问该对象，从而限制、增强或修改该对象的一些特性。	结构	对象
适配器	Adapter	将一个类的接口转换成客户希望的另外一个接口，使得原本由于接口不兼容而不能一起工作的那些类能一起工作。	结构	类
桥接	Bridge	将抽象与实现分离，使它们可以独立变化。它是用组合关系代替继承关系来实现，从而降低了抽象和实现这两个可变维度的耦合度。	结构	对象
包装	Decorator	动态的给对象增加一些职责，即增加其额外的功能。	结构	对象
外观	Facade	为多个复杂的子系统提供一个一致的接口，使这些子系统更加容易被访问。	结构	对象
享元	Flyweight	运用共享技术来有效地支持大量细粒度对象的复用。	结构	对象
组合	Composite	将对象组合成树状层次结构。使用户对单个对象和组合对象具有一致的访问性。	结构	对象
模板方法	Template Method	定义一个操作中的算法骨架，而将算法的一些步骤延迟到子类中，使得子类可以不改变该算法结构的情况下重定义该算法的某些特定步骤。	行为	类
策略	Strategy	定义了一系列算法，并将每个算法封装起来，使它们可以相互替换，且算法的改变不会影响使用算法的客户。	行为	对象
命令	Command	将一个请求封装为一个对象，使发出请求的责任和执行请求的责任分割开。	行为	对象
职责链	Chain of Responsibility	把请求从链中的一个对象传到下一个对象，直到请求被响应为止。通过这种方式去除对象之间的耦合。	行为	对象
状态	State	允许一个对象在其内部状态发生改变时改变其行为能力。	行为	对象
观察者	Observer	多个对象间存在一对多关系，当一个对象发生改变时，把这种改变通知给其他多个对象，从而影响其他对象的行为。	行为	对象
中介者	Mediator	定义一个中介对象来简化原有对象之间的交互关系，降低系统中对象间的耦合度，使原有对象之间不必相互了解。	行为	对象
迭代器	Iterator	提供一种方法来顺序访问聚合对象中的一系列数据，而不暴露聚合对象的内部表示。	行为	对象
访问者	Visitor	在不改变集合元素的前提下，为一个集中的每个元素提供多种访问方式，即每个元素有多个访问者对象访问。	行为	对象
备忘录	Memento	在不破坏封装性的前提下，获取并保存一个对象的内部状态，以便以后恢复它。	行为	对象
解释器	Interpreter	提供如何定义语言的文法，以及对语言句子的解释方法，即解释器。	行为	类

开闭原则	软件实体应当对扩展开放，对修改关闭。			
里氏替换原则	子类可以扩展父类的功能，但不能改变父类原有的功能【是开闭原则的补充】			
依赖倒置原则	高层模块不应该依赖低层模块，两者都应该依赖其抽象。通过要面向接口的编程来降低类间的耦合性【是实现开闭原则的重要途径之一，它降低了客户与实现模块之间的耦合。】			
单一职责原则	单一职责原则规定一个类应该有且仅有一个引起它变化的原因，否则类应该被拆分控制类的粒度大小、将对象解耦、提高其内聚			
接口隔离原则	要为各个类建立它们需要的专用接口而不要试图去建立一个很庞大的接口供所有依赖它的类去调用			
迪米特法则	如果两个软件实体无须直接通信，那么就不应当发生直接的相互调用，可以通过第三方转发该调用。其目的是降低类之间的耦合度，提高模块的相对独立性。			
合成复用原则	如果要使用继承关系，则必须严格遵循里氏替换原则。过将已有的对象纳入新对象中，作为新对象的成员对象来实现的，新对象可以调用已有对象的功能，从而达到复用。【合成复用原则同里氏替换原则相辅相成的，两者都是开闭原则的具体实现规范。】			