

设计模式速记

2019-10-07 9:09

1. 工厂方法模式 (Factory Method) : 定义一个创建对象的接口, 但由子类决定需要实例化哪一个类。工厂方法使得子类实例化的过程推迟
 - a. 速记关键词: 动态生产对象
2. 抽象工厂模式 (Abstract Factory) : 提供一个接口, 可以创建一系列相关或相互依赖的对象, 而无需指定它们具体的类
 - a. 速记关键词: 生产成系列对象
3. 构建器模式 (Builder) : 将一个复杂的表示与其构造相分离, 使得相同的构建过程能够得出不同的表示
 - a. 速记关键词: 复杂对象构造
4. 原型模式 (Prototype) : 用原型实例指定创建对象的类型, 并且通过拷贝这个原型来创建新的对象
 - a. 速记关键词: 克隆对象
5. 单例模式 (Singleton) : 保证一个类只有一个实例, 并提供一个访问它的全局访问点
 - a. 速记关键词: 单实例
6. 适配器模式 (Adapter) : 将一个类的接口转换成用户希望得到的另一种接口。它使原本不相容的接口得以协同工作
 - a. 速记关键词: 转换接口
7. 桥接模式 (Bridge) : 将类的抽象部分和它的实现部分分离开来, 使它们可以独立地变化
 - a. 速记关键词: 继承树拆分
8. 组合模式 (Composite) : 将对象组合成树型结构以表示“整体-部分”的层次结构, 使得用户对单个对象和组合对象的使用具有一致性
 - a. 速记关键词: 树形目录结构
9. 装饰模式 (Decorator) : 动态地给一个对象添加一些额外的职责。它提供了用子类扩展功能的一个灵活的替代, 比派生一个子类更加灵活
 - a. 速记关键词: 动态附加职责
10. 外观模式 (Facade) : 定义一个高层接口, 为子系统中的一组接口提供一个一致的外观, 从而简化了子系统的使用
 - a. 速记关键词: 对外统一接口
11. 享元模式 (Flyweight) : 提供支持大量细粒度对象共享的有效方法
 - a. 速记关键词: 汉字编码
12. 代理模式 (Proxy) -- 结构型: 为其他对象提供一种代理以控制这个对象的访问
 - a. 速记关键词: 快捷方式
13. 职责链模式 (Chain of Responsibility) : 通过给多个对象处理请求的机会, 减少请求的发送者与接收者之间的耦合。将接收对象链接起来, 在链中传递请求, 直到有一个对象处理这个请求
 - a. 速记关键词: 传递职责
14. 命令模式 (Command) : 将一个请求封装为一个对象, 从而可用不同的请求对客户进行参数化, 将请求排队或记录请求日志, 支持可撤销的操作
 - a. 速记关键词: 日志记录, 可撤销
15. 解释器模式 (Interpreter) : 给定一种语言, 定义它的文法表示, 并定义一个解释器, 该解释器用来根据文法表示来解释语言中的句子
 - a. 速记关键词: 虚拟机的机制
16. 迭代器模式 (Iterator) : 提供一种方法来顺序访问一个聚合对象中的各个元素, 而不需要暴露该对象的内部表示
 - a. 速记关键词: 数据集
17. 中介者模式 (Mediator) -- 行为型: 用一个中介对象来封装一系列的对象交互。它使各对象不需要显式地相互调用, 从而达到低耦合, 还可以独立地改变对象间的交互
 - a. 速记关键词: 不直接引用
18. 备忘录模式 (Memento) : 在不破坏封装性的前提下, 捕获一个对象的内部状态, 并在该对象之外保存这个状态, 从而可以在以后将该对象恢复到原先保存的状态
 - a. 速记关键词: 游戏存档
19. 观察者模式 (Observer) : 定义对象间的一种一对多的依赖关系, 当一个对象的状态发生改变时, 所有依赖于它的对象都得到通知并自动更新
 - a. 速记关键词: 联动
20. 状态模式 (State) : 允许一个对象在其内部状态改变时改变它的行为
 - a. 速记关键词: 状态变成类
21. 策略模式 (Strategy) : 定义一系列算法, 把它们一个个封装起来, 并且使它们之间可互相替换, 从而让算法可以独立于使用它的用户而变化
 - a. 速记关键词: 多方案切换
22. 模板方法模式 (Template Method) : 定义一个操作中的算法骨架, 而将一些步骤延迟到子类中, 使得子类可以不改变一个算法的结构即可重新定义算法的某些特定步骤
 - a. 速记关键词: 框架
23. 访问者模式 (Visitor) : 表示一个作用于某对象结构中的各元素的操作, 使得在不改变各元素的类的前提下定义作用于这些元素的新操作
 - a. 速记关键词: 数据与操作分离