

【VIP直播课】

# 模板模式及适配器模式

Tom





**咕泡学院-Tom**  
**前中电在线技术总监**  
**前超星网架构师**  
**现为咕泡学院联合创始人**

---

10余年Java经验。精通java语言。开发过多套企业UI框架、ORM框架。热衷于分享经验，共同进步。

不只做一个技术者，更要做一个思考者。





## 书法爱好者、绘画爱好者 编程界字写得最好的 书法界编程最牛逼的

自幼开始练习书法。中学期间，曾获市级青少年杯书法竞赛一等奖，获校园杯美术竞赛工笔画一等奖，获校园征文比赛二等奖。大学担任学生会宣传部长，负责校园黑板报、校园刊物的编辑排版设计。参加工作后，担任过家具建模、平面设计等工作。

亲自设计咕泡学院Logo。



扫码加入书法兴趣小组



群名称:Tom老师书法兴趣小组  
群 号:549209007

有兴趣的可以扫码加入书法兴趣小组



- 1、学会用模板模式梳理使用工作中流程标准化的业务场景。
- 2、通过学习适配模式，优雅地解决代码功能的兼容问题。
- 3、了解JDK源码和Spring源码中对模板模式的运用。



- 1、定位高级课程，不太适合接触业务场景比较单一的学员。
- 2、深刻了解模板模式和适配器模式的应用场景。





## 模板模式详解



**模板模式通常又叫模板方法模式（Template Method Pattern）是指定义一个算法的骨架，并允许子类为一个或者多个步骤提供实现。**

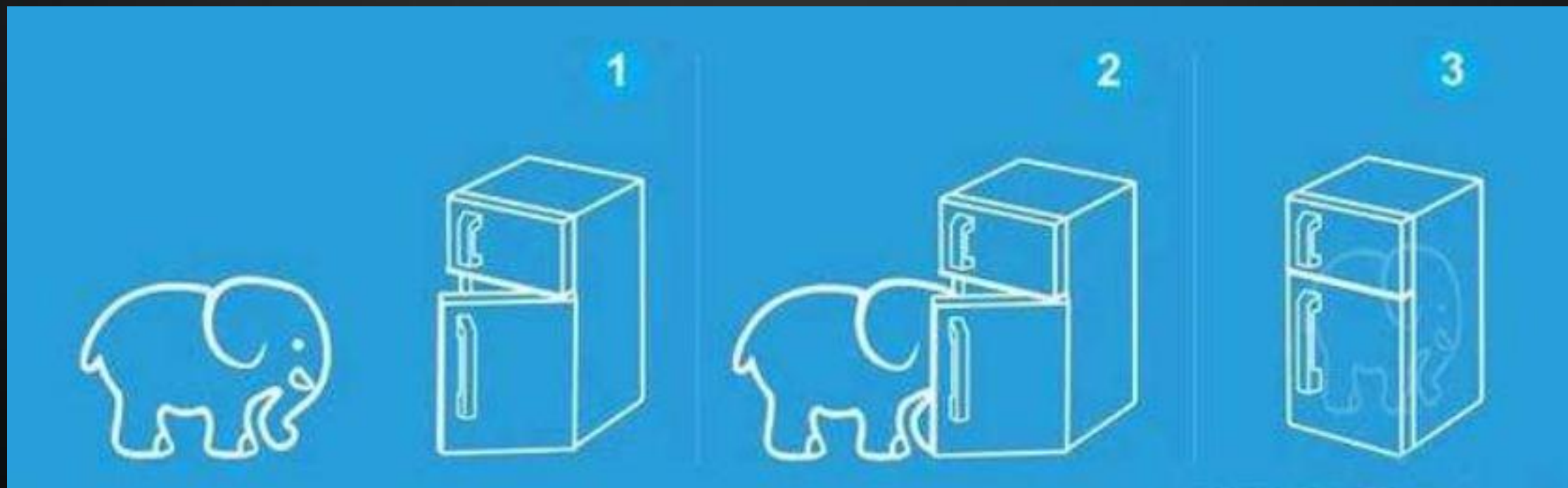
**模板方法使得子类可以在不改变算法结构的情况下，重新定义算法的某些步骤。**

**属于行为性设计模式。**





# 如何把大象装进冰箱？



- 1、一次性实现一个算法的不变的部分，并将可变的行为留给子类来实现。
- 2、各子类中公共的行为被提取出来并集中到一个公共的父类中，从而避免代码重复。



- 1、提高代码的复用性。
- 2、提高代码的扩展性。
- 3、符合开闭原则。



- 1、类数目的增加。
- 2、间接地增加了系统实现的复杂度。
- 3、继承关系自身缺点，如果父类添加新的抽象方法，所有子类都要改一遍。



## 适配器模式详解



**适配器模式（Adapter Pattern）是指将一个类的接口转换成客户期望的另一个接口，使原本的接口不兼容的类可以一起工作。**

**属于结构型设计模式。**





- 1、已经存在的类，它的方法和需求不匹配（方法结果相同或相似）的情况。
- 2、适配器模式不是软件设计阶段考虑的设计模式，是随着软件维护，由于不同产品、不同厂家造成功能类似而接口不相同情况下的解决方案。





- 1、能提高类的透明性和复用，现有的类复用但不需要改变。
- 2、目标类和适配器类解耦，提高程序的扩展性。
- 3、在很多业务场景中符合开闭原则。



- 1、适配器编写过程需要全面考虑，可能会增加系统的复杂性。
- 2、增加代码阅读难度，降低代码可读性，过多使用适配器会使系统代码变得凌乱。



# 谢谢观看

Tom



Tom老师QQ号：441221062

