

# Golang 随记 (工厂模式)

- 工厂模式的分类
  - 简单工厂模式
  - 工厂方法模式
  - 抽象工厂模式
  - 容器工厂模式 (需要第三方依赖注入框架)
- 为什么需要工厂模式：工厂模式是一种创建型设计模式，通过提供一个工厂类来实现对类的构造，实现对类和业务方法的解耦，如果类的构造过程发生变更，也可以统一在工厂类中进行处理，对业务方法屏蔽细节。
- 简单工厂模式：
  - 对于需要构造的类或组件，根据其共性，抽离出一个公共 interface
  - 每个类或组件需要对 interface
  - 定义一个工厂类，在构造器方法中接受具体的组件类型，完成对应类型组件的构造
- 简单模式优缺点：
  - 优点
    - ◆ 工厂模式中最简单直观的一种类型
    - ◆ 对象的创建逻辑集中在工厂类中，便于维护和解耦
  - 缺点
    - ◆ 当有新的类需要实现构造时，需要在工厂类的对应构造器方法中进行修改 (比如增加 switch case 的分支) 不符合开闭原则，要求面向扩展开放，面向修改关闭
- 工厂方法模式：
  - 相较于简单工厂模式，工厂方法模式中将简单工厂模式中的工厂类定义为了 interface
  - 每个实例类定义实现自己的工厂类和构造器方法，并符合总的 interface 的要求
  - 这样后续需要增加或修改类组件，只需修改或增加对应类的工厂类和构造器方法
- 工厂方法模式的优缺点：
  - 优点
    - ◆ 符合开闭原则
  - 缺点
    - ◆ 需要为每个水果实现一个工厂类，代码冗余度高
- 抽象工厂模式：
  - 通过将组件拆分为产品族和产品等级的维度，将需要频繁扩展的维度和相对稳定的维度进行拆分，尝试兼具简单工厂模式和工厂方法模式的优点，但是需要明确产品族和产品等级的定义，否则出现偏差会产生事与愿违的效果。