Rust语言主要的设计模式

2020年5月1日 10:28

1	**创建型模式 ** 这些设计模式提供了一种在创建对象的同时隐藏创建逻辑的方式， 而不是使用 new 运算符直接实例化对象。 这使得程序在判断针对某个给定实例需要创建哪些对象时更加灵活。	工厂模式（Factory Pattern） [抽象工厂模式（Abstract Factory Pattern）] (Abstract Factory) 建造者模式（Builder Pattern） 单例模式（SingletonPattern）
2	**行为型模式** 这些设计模式特别关注对象之间的通信。	策略模式（Strategy Pattern） 状态模式（State Pattern） 命令模式（Command Pattern） 迭代器模式（Itera tor Pattern） 观察者模式（Observer Pattern） 责任链模式（Chain of Responsibility Pattern）
3	**结构型模式** 这些设计模式关注类和对象的组合。继承的概念被用来组合接口和定义组合对象获得新功能的方式。	适配器模式（Adapter Pattern） 装饰器模式（Decorator Pattern） 代理模式（Proxy Pattern）