

Qt 工程师进阶技术《23 种设计模式》

设计模式是解决特定问题的一系列套路，这套方案提高代码可复用性、可读性、稳健性、可维护性及安全性。

23 种设计模式可分为三类：结构型模式（侧重类与对象之间的组合）、行为型模式（侧重对象之间的通信）、创建型模式（侧重对象的创建）。

结构型模式（7 种）：代理模式、桥接模式、适配器模式、外观模式、享元模式、组合模式、装饰器模式

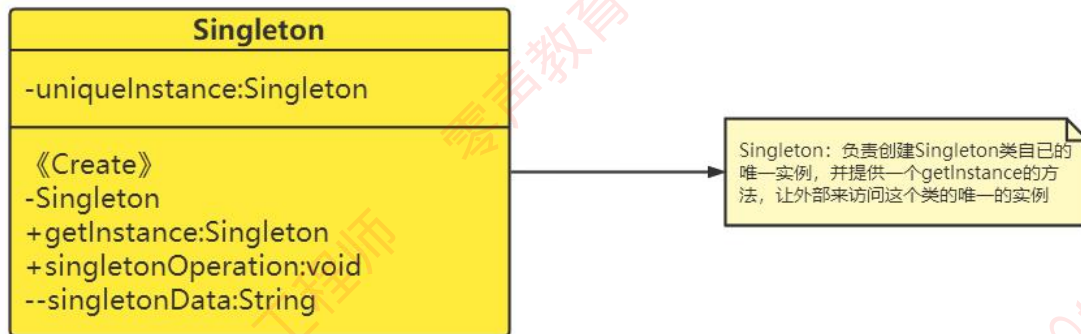
行为型模式（11 种）：备忘录模式、访问者模式、解释器模式、迭代器模式、中介模式、观察者模式、状态模式、责任链模式、命令模式、策略模式、模板方法模式。

创建型模式（5 种）：单例模式、工厂模式、抽象工厂模式、建造者模式、原型模式。

一、对象性能之 Singleton（单例模式）/Flyweight（享元模式）

1、单例模式（本质就是控制实例数目）定义：保证一个类仅有一个实例，并提供一个该实例的全局访问点。实例构造器一般设备为 private，但是也可以设置为 protected 以允许子类派生。一般情况不要支持拷贝构造函数和 Clone 接口。

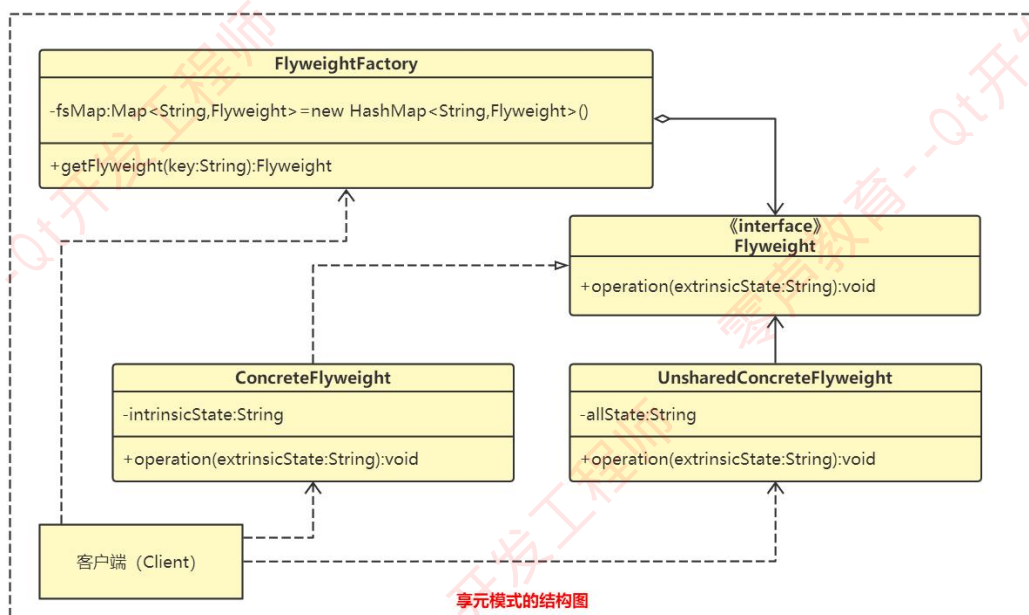
2、单例模式的结构图具体如下：



3、单例的优点和缺点

- ❖ 时间和空间：懒汉式是典型的时间换出空间。饿汉式是典型的空间换时间。
- ❖ 线程安全：不加同步的懒汉式的线程不安全的。饿汉式是线程安全的。

4、享元模式定义：运用共享技术有效地支持大量细粒度的对象。享元模式的结构图具体如下：

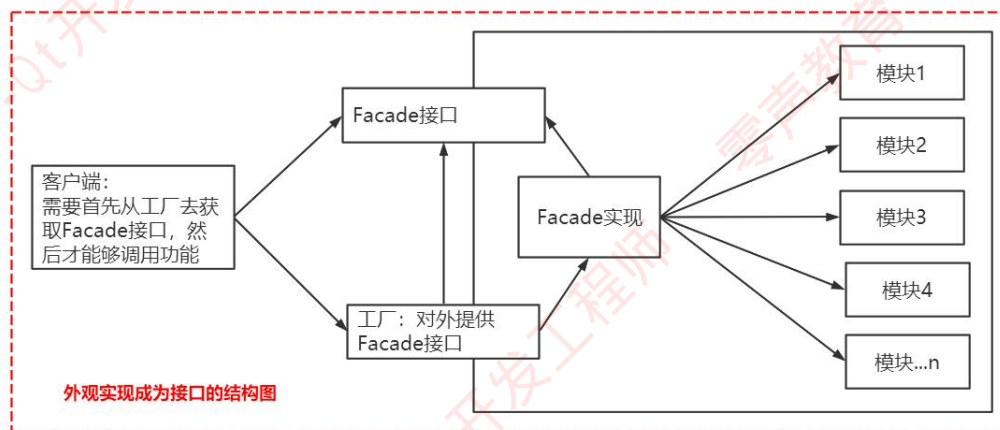
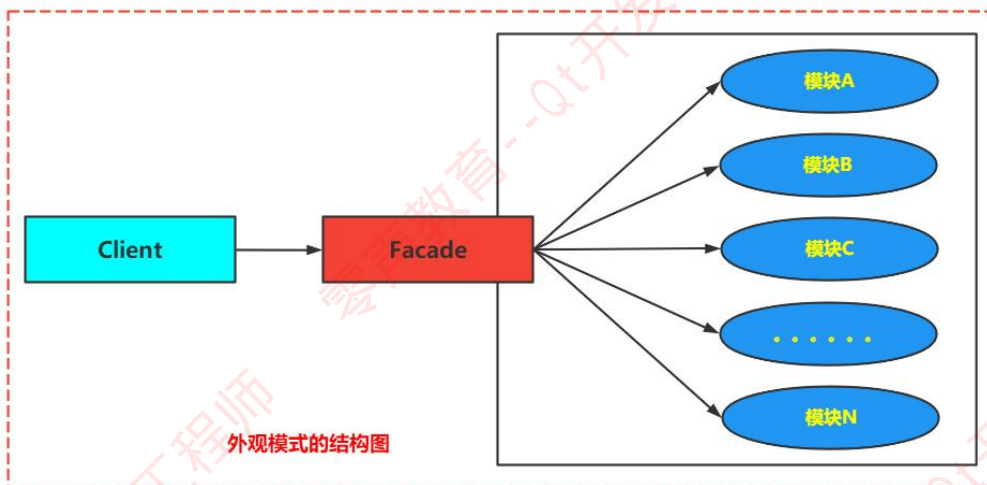


5、享元模式优点和缺点

- ❖ 减少对象数量，节省内存空间。
- ❖ 维护共享对象，需要额外开销。

二、接口隔离之 Facade (外观模式) /Proxy (代理模式) /Mediator (中介者模式) /Adapter (适配器模式)

1、外观模式定义：为子系统的一组接口提供一个一致（稳定）的界面。facade 模式定义一个高层接口，这个接口使得--子系统更加容易使用（复用）。



三、单一职责之 Bridge/Decorator

四、组件协作之 Template/Strategy/Observer

五、手把手教会，应用设计模式项目实战