1. 定义

享元模式（Flyweight Pattern）是以共享的模式来支持大量细粒度对象的复用。

例如java中 String a=”abc”; String b=”abc”; 其实a,b都指向了常量池中的某个字符串”abc”,这种对象创建的模式，就避免了大量对象创建时非必要的资源消耗。享元模式的”享”就有一物被众人共享的意思，所以享元模式也是池技术的重要方式。所以使用共享对象有效的支持大量的细粒度对象。

1. 使用实例

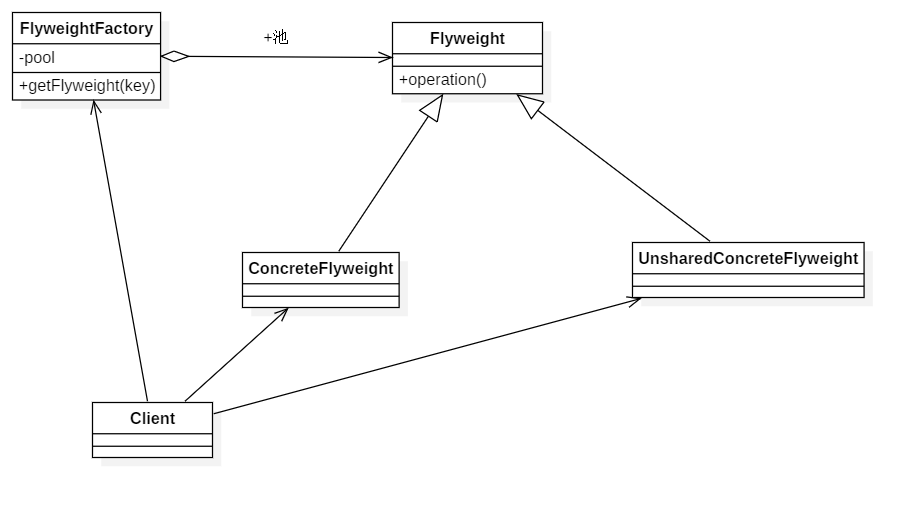
假如我们有2个科目，有三位考生分别报考，我们会定义考试实体ExamInofo,如果不适用模式，那么每次有考生参与一场考试科目的话，我们就会实例化一个ExamInfo,一共我们要实例化6个这样的实体。如果使用享元模式，我们就只需要实例化2个这样的实体，然后通过内部状态的set方法进行不同的对象赋值操作，节省大量的内存。







1. 组成角色



\*抽象享元角色（Flyweight）:一般是一个具体的抽象类，同时定义了对象的外部状态和内部状态的接口实现；

\*具体的享元角色（ConcreteFlyWEIGHT）:具体的一个产品类，实现了抽象享元角色中定义的接口，改角色需要注意的是内部状态的处理应该与环境无关；

\*享元工厂（FlyweightFactory）:改角色职责比较清晰，就是一个池工厂，提供池对象和获取池对象中的方法。

