* 设计模式 - 享元模式（flyweight）：

FLYWEIGHT在拳击比赛中指最轻量级。享元模式以共享的方式高效的支持大量的细粒度对象。

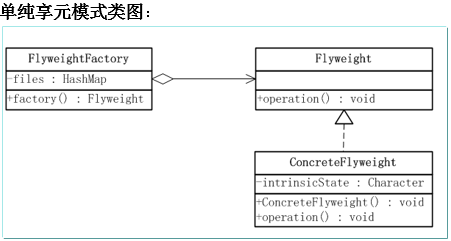
我们把一个对象中不同的属性变成方法的参数称为外蕴状态；那些相同的属性不作为方法的参数称为内蕴状态。

享元模式能做到共享的关键是区分内蕴状态和外蕴状态。

**内蕴状态：**是存储在享元对象内部的，并且是不会随环境的改变而有所不同。因此，一个享元可以具有内蕴状态并可以共享。比如自动拆装箱的时候，部分范围的Integer可以在内存中被重用。

**外蕴状态：**是随环境的改变而改变的、不可以共享的。享元对象的外蕴状态必须由客户端保存，并在享元对象被创建之后，在需要使用的时候再传入到享元对象内部。外蕴状态不可以影响享元对象的内蕴状态，它们是相互独立的。

享元模式可以分成**单纯享元模式**和**复合享元模式**两种形式。复合享元模式是单纯享元模式和合成模式的组合。

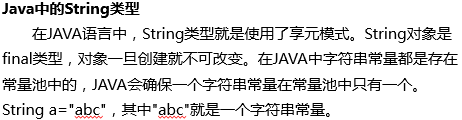


**单纯享元模式角色**：

1）抽象享元（Flyweight）角色：给出一个接口，规定享元需要实现的方法。

2）具体享元（Concrete Flyweight）角色：实现享元角色规定的接口。如果有内蕴状态，必须提供存储空间(数据或字段)。

3）享元工厂（Flyweight Factory）角色：负责创建和管理享元对象。本角色必须保证享元对象可以被系统适当的共享。如果有了合适的享元对象，就使用已有的；如果系统中没有合适的享元对象，就应该创建一个合适的享元对象。



* 设计模式 - 享元模式Demo：

