# **状态模式**

功能用途：

行为型模式，对象的行为依赖于它的状态(属性)，并且可以根据它的状态改变而改变它的行为。

应用实例：

当代码中包含有大量与对象状态有关的条件语句

角色和职责：

Context:用户对象Dog

拥有一个state类型的成员，以标识对象的当前状态

State:接口或基类

封装与Context的特定状态相关的行为

ConcreteState:接口实现类或子类 MState，LState，DState

实现了一个与Context某个状态相关的行为

优点：

封装了转换规则

枚举可能的状态，在枚举状态之前确定状态种类

将所有状态与某个状态有关的行为放置在一个类中，并且可以方便地增加行的状态，只需要改变对象状态即可改变对象行为

允许状态改变逻辑和状态对象合为一体，而不是某个巨大的条件语句块

可以让对个环境对象共享一个状态对象，从而减少系统中对象的个数

UML：

