



# Rust 编程语言入门



Microsoft®  
Most Valuable  
Professional

杨旭，微软 MVP  
Rust、Go 开发者

## 17.3 实现面向对象的设计模式

# 状态模式

- 状态模式（state pattern）是一种面向对象设计模式：
  - 一个值拥有的内部状态由数个状态对象（state object）表达而成，而值的行为则随着内部状态的改变而改变
- 使用状态模式意味着：
  - 业务需求变化时，不需要修改持有状态的值的代码，或者使用这个值的代码
  - 只需要更新状态对象内部的代码，以便改变其规则。或者增加一些新的状态对象
- （例子）

# 状态模式的取舍权衡

- 缺点：
  - 某些状态之间是相互耦合的
  - 需要重复实现一些逻辑代码

# 将状态和行为编码为类型

- 将状态编码为不同的类型：
  - Rust 类型检查系统会通过编译时错误来阻止用户使用无效的状态
- （例子）

# 总结

- Rust 不仅能够实现面向对象的设计模式，还可以支持更多的模式
- 例如：将状态和行为编码为类型
- 面向对象的经典模式并不总是 Rust 编程实践中的最佳选择，因为 Rust 具有所有权等其它面向对象语言没有的特性！

再见

