

# 软件体系结构

Zhenyan Ji

— Beijing Jiaotong University —

## 结构型模式--外观模式

### 外观模式

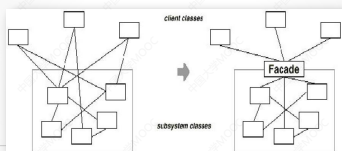
#### » 目的

- 为系统中的接口集合提供**统一**的接口。外观定义了使子系统更易于使用的更高层次接口。

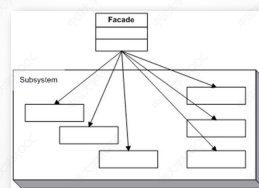
### 外观模式

- » 子系统或子系统集中的类暴露的接口可能很复杂。
- » 降低复杂性的方法是引入外观对象，它为子系统的通用功能提供单一的、简化的接口。

### 外观模式



### 外观模式



## 外观模式

### 参与者

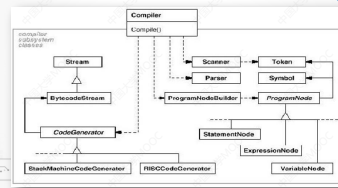
#### Facade

- 了解哪个子系统类负责哪个请求。
- 将客户端请求委托给合适的子系统对象。

#### Subsystem Classes

- 实现子系统功能。
- 处理Facade对象分配的工作。
- 不知道Facade对象，没有引用Facade对象。

## 外观模式



## 外观模式

### 什么时候使用外观模式

- 为复杂的子系统提供简单的接口。此接口对于大多数客户程序来说足够好，更复杂的客户程序可以调用子系统接口。
- 将子系统的类与其客户程序和其他子系统解耦，从而提升子系统的独立性和可移植性。

## 结果

### 好处

- 向客户程序隐藏了子系统的实现，使得子系统更易于使用。
- 降低了子系统和其客户程序之间的耦合度。
- 减少了大型软件系统中的编译依赖关系。

## 结果

- 使移植系统到其他平台变得更简单。
- 不会阻止复杂客户程序访问底层类。
- 注意：Facade类没有添加任何功能，只是简化了接口。

### 它不会阻止客户程序访问底层类！