

# 软件体系结构

Zhenyan Ji

— Beijing Jiaotong University —

## 结构型设计模式

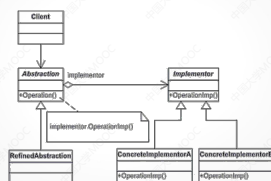
### 桥接模式

### 桥接模式

#### 目的:

- 桥接模式将抽象与其实现分离, 以便两者可以**独立变化**。

### 桥接模式



### 桥接模式

#### 参与者

#### Abstraction

- 定义了抽象的接口
- 维护对Implementor类型的对象的引用

#### RefinedAbstraction

- 扩展了Abstraction定义的接口

### 桥接模式

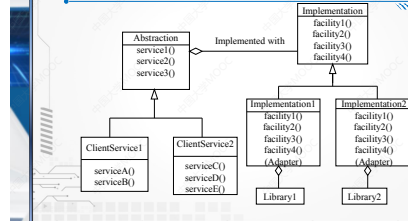
#### Implementor

- 定义实现类的接口。
- 该接口不必与Abstraction的接口完全对应; 实际上这两个接口可能完全不同。

## 桥接模式

- 通常, `Implementor` 接口仅提供基本操作, `Abstraction` 定义了基于这些基本操作的高级操作。
- » `ConcreteImplementor`
- 实现 `Implementor` 接口并定义其具体实现

## 例子



## 例子

```
class Abstraction {
    private Implementation implementation;
    public Abstraction(Implementation imp) {
        implementation = imp;
    }

    public void service1() {
        implementation.facility1();
        implementation.facility2();
    }
}
```

## 例子

```
public void service2() {
    implementation.facility2();
    implementation.facility3();
}

public void service3() {
    implementation.facility1();
    implementation.facility2();
    implementation.facility4();
}
```

## 适用性

- 01 当需要避免抽象和实现之间的绑定以及抽象和实现需要独立变化时, 使用桥接模式。
- 02 使用桥接模式将使客户端程序保持不变, 无需重新编译代码。