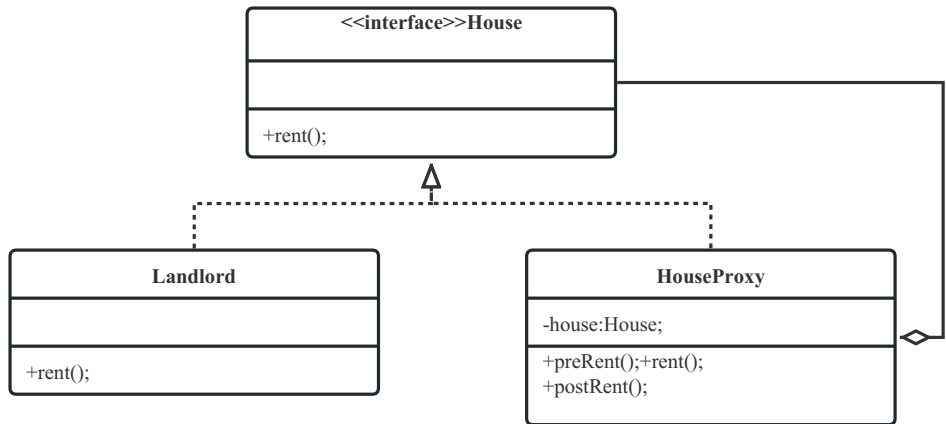


• 定义：由于某些原因需要给某对象提供一个代理以控制对该对象的访问。这时，访问对象不适合或者不能直接引用目标对象，代理对象作为访问对象和目标对象之间的中介。

- 三种实现方式：
- 静态代理
- 动态代理
- cglib代理：第三方代码生成类库

静态代理：



动态代理：又称**JDK**代理或接口代理

• 动态代理利用了JDK API，动态地在内存中构建代理对象，从而实现对目标对象的代理功能

静态代理 v.s. 动态代理

- 主要区别：
- 静态代理在编译时就已经实现，编译完成后代理类是一个实际的class文件
- 动态代理是在运行时动态生成的，即编译完成后没有实际的class文件，而是在运行时动态生成类字节码，并加载到JVM中

动态代理特点

- 动态代理对象不需要实现接口
- 但是要求目标对象必须实现接口，否则不能使用动态代理

代理模式优点

- 代理模式在客户端与目标对象之间起到一个中介作用和保护目标对象的作用
- 代理对象可以扩展目标对象的功能
- 代理模式能将客户端与目标对象分离，在一定程度上降低了系统的耦合度，增加了程序的可扩展性

代理模式缺点

- 增加了系统的理解 and 设计难度
- 代理模式会造成系统设计中类的数量增加
- 在客户端和目标对象之间增加一个代理对象，会造成请求处理速度变慢

代理模式适用环境

- 当无法或不想直接引用某个对象或访问某个对象存在困难时，可以通过代理对象来间接访问
- 使用代理模式主要有两个目的：一是保护目标对象，二是增强目标对象