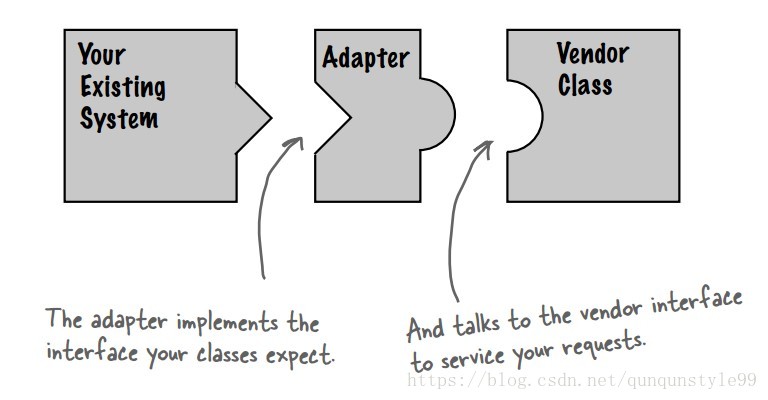
**适配器模式**

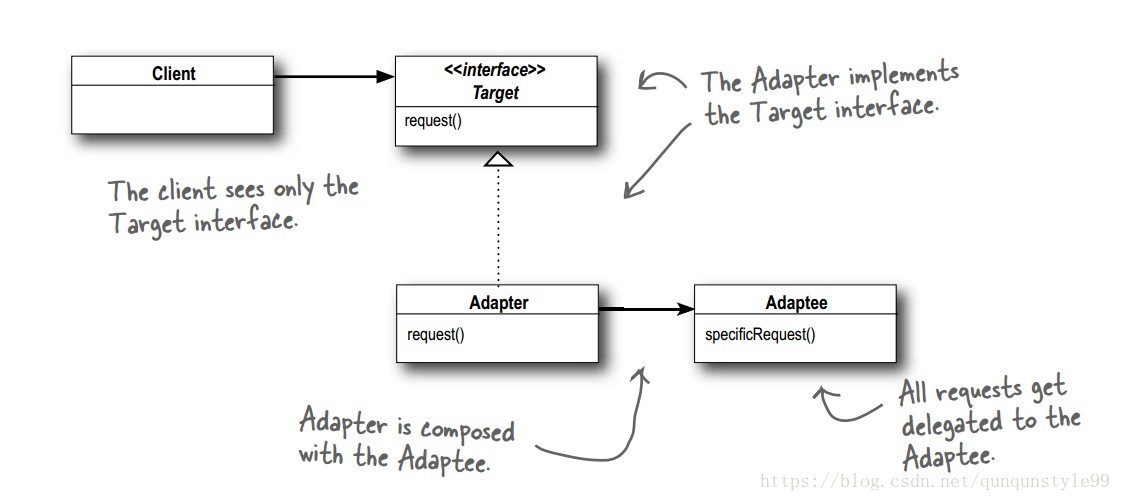
**1. 模式定义**

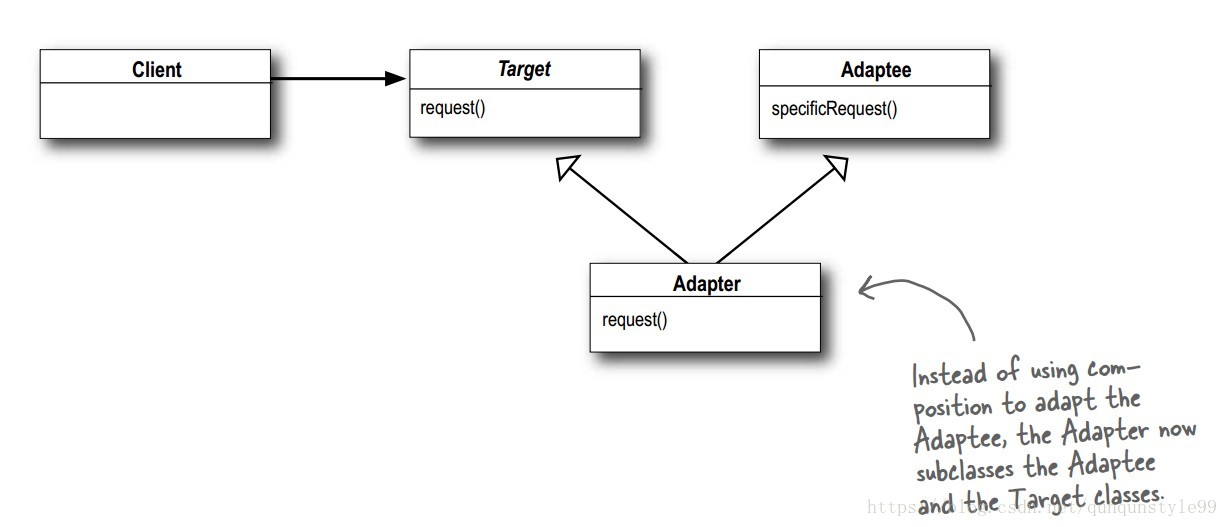
将一个类的接口，转换为客户期望的另一个接口。适配器让原本不兼容的类可以合作无间。



**2. 模式类图**

有两种适配器模式的实现，一种是对象方式，一种是类方式。对象方式是通过组合的方法，让适配器类（Adapter）拥有一个待适配的对象（Adaptee），从而把相应的处理委托给待适配的对象。类方式用到多重继承，Adapter 继承 Target 和 Adaptee，先把 Adapter 当成 Adaptee 类型然后实例化一个对象，再把它当成 Target 类型的，这样 Client 就可以把这个对象当成 Target 的对象来处理，同时拥有 Adaptee 的方法。



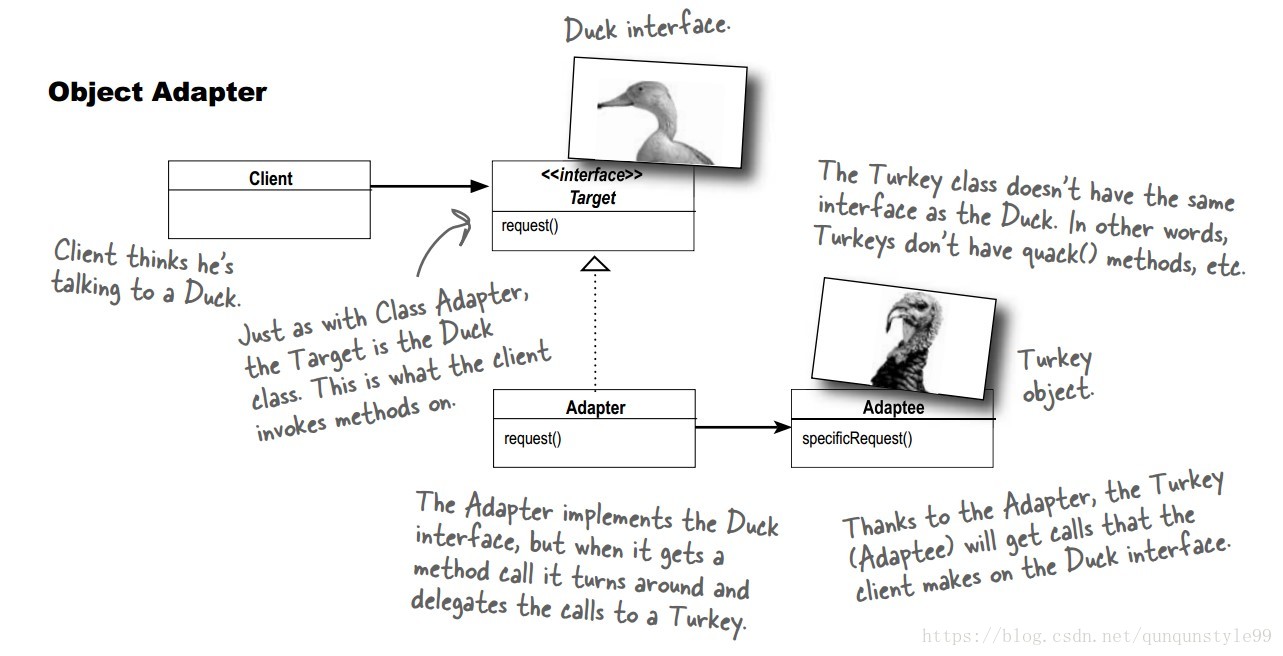


**3. 问题描述**

鸭子（Duck）和火鸡（Turkey）拥有不同的叫声，Duck 调用的是 quack() 方法，而 Turkey 调用 gobble() 方法。

要求将 Turkey 的 gobble() 方法适配成 Duck 的 quack() 方法。

**4. 解决方案类图**



**5. 代码实现**

public interface Duck {

public void quack();

}

public interface Turkey {

public void gobble();

}

public class WildTurkey implements Turkey{

@Override

public void gobble() {

System.out.println("gobble!");

}

}

public class TurkeyAdapter implements Duck{

Turkey turkey;

public TurkeyAdapter(Turkey turkey) {

this.turkey = turkey;

}

@Override

public void quack() {

turkey.gobble();

}

}

public class DuckTestDrive {

public static void main(String[] args) {

Turkey turkey = new WildTurkey();

Duck duck = new TurkeyAdapter(turkey);

duck.quack();

}

}

运行结果

gobble!