Adapter Pattern

What Is an Adapter Pattern?

The Adapter pattern is a useful pattern for making two different class types communicate in a similar manner. It performs much as its name implies: it creates an adaptation between two classes of different types so they can become interchangeable.

The Adapter pattern has three important components: the Target, Adapter, and Adaptee. The target is the class for which we wish to implement the adapter. We inherit from the target to create our adapter. The adapter is the class that provides the join for the two disparate class types and houses the methods for conjoining functionality between them. It is an inherited member from the target. The adaptee is the class that we wish to give access to the methods and functionality of the target. The adapter allows interchangeability between the target and the adaptee.

![Figure 4-1. UML for Adapter pattern](image)

This seems quite useful. In many real-world applications, it is similar mechanism that is actually used.

Problem Another Inheritance

For our final demonstration. I have two disparate functions, we will inherit the levels of each of them.

```java
//Adaptee
class Water {
    public void
        {...
    public void
        {...
    public void
        {...
}
}
```

This seems quite useful. In many real-world applications, it is similar mechanism that is actually used.