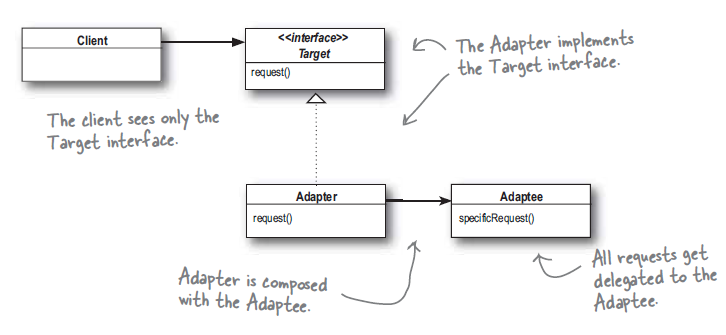
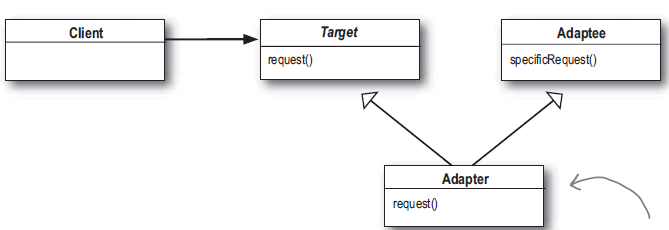
**The Adapter Pattern:**

1. **Definition**: The adapter pattern adapts interface of a class to the interface that a client expects. Adapter let incompatible interfaces work together.



1. **Advantages**:
   1. This pattern binds the client to an interface not to an implementation. We can use different adapters used to adapt different backend interfaces.
   2. We use object composition to wrap adaptee in adapter so any subclasses of adaptee can be used.
2. **Class adapter**: Instead of using composition to adapt the adaptee the adapter subclasses both target and adaptee classes.



* 1. Unlike object adapter pattern class adapter can only commit to a specific adaptee class.
  2. Adapter don’t have to implement whole adaptee instead override its behavior if needed.

1. You can also use Adapter design pattern in Java for conversion classes, e.g. Suppose your client do all calculation in Miles and the library you are using expects Kilometers. In this case, you can write an Adapter class, which takes miles from Client, converts it to Kilometer and leverages external library methods for all calculation. While returning a result, it can convert KM back to miles and send the result to Client.

