

## 7 the Adapter and Facade Patterns

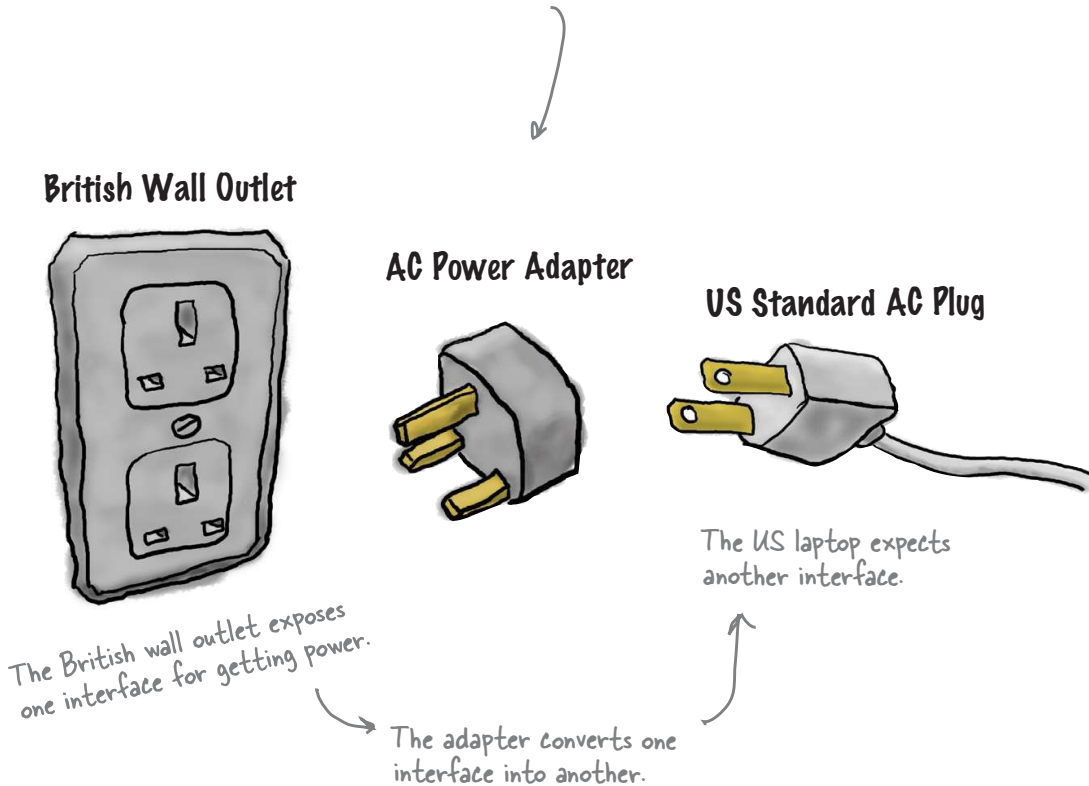
# *Being Adaptive*



**In this chapter we're going to attempt such impossible feats as putting a square peg in a round hole.** Sound impossible? Not when we have Design Patterns. Remember the Decorator Pattern? We **wrapped objects** to give them new responsibilities. Now we're going to wrap some objects with a different purpose: to make their interfaces look like something they're not. Why would we do that? So we can adapt a design expecting one interface to a class that implements a different interface. That's not all; while we're at it, we're going to look at another pattern that wraps objects to simplify their interface.

## Adapters all around us

You'll have no trouble understanding what an OO adapter is because the real world is full of them. How's this for an example: Have you ever needed to use a US-made laptop in Great Britain? Then you've probably needed an AC power adapter...



You know what the adapter does: it sits in between the plug of your laptop and the British AC outlet; its job is to adapt the British outlet so that you can plug your laptop into it and receive power. Or look at it this way: the adapter changes the interface of the outlet into one that your laptop expects.

Some AC adapters are simple—they only change the shape of the outlet so that it matches your plug, and they pass the AC current straight through—but other adapters are more complex internally and may need to step the power up or down to match your devices' needs.

Okay, that's the real world; what about object-oriented adapters? Well, our OO adapters play the same role as their real-world counterparts: they take an interface and adapt it to one that a client is expecting.

*How many other real-world adapters can you think of?*