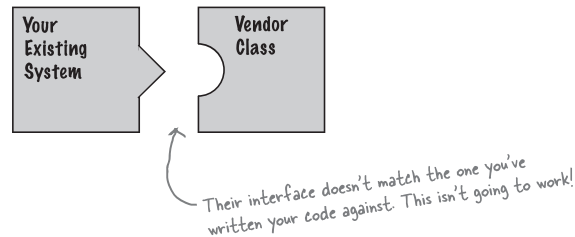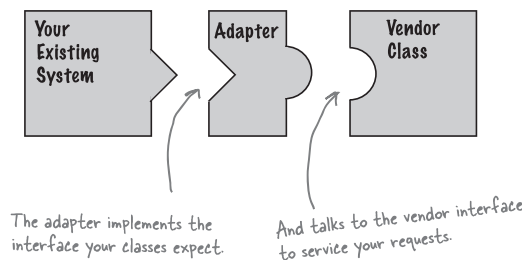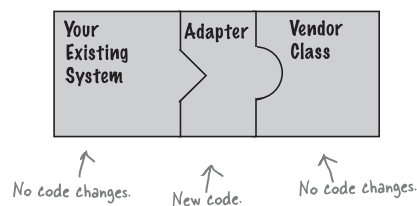*the* adapter *pattern*

Object oriented adapters

Say you've got an existing software system that you need to work a new vendor class library
into, but the new vendor designed their interfaces differently than the last vendor:

*Their interface doesn't match the one you've
written your code against. This isn't going to work!*

Okay, you don't want to solve the problem by changing your existing code (and you can't
change the vendor's code). So what do you do? Well, you can write a class that adapts the
new vendor interface into the one you're expecting.

*The adapter implements the
interface your classes expect.*

*And talks to the vendor interface
to service your requests.*

The adapter acts as the middleman by receiving requests from the client and converting
them into requests that make sense on the vendor classes.

No code changes.   *New code.*   *No code changes.*

*Can you think of a solution
that doesn't require YOU to
write ANY additional code
to integrate the new vendor
classes? How about making the
vendor supply the adapter class.*

you are here ▸   **237**