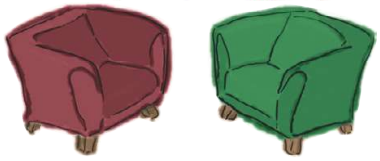
Fireside Chats



Tonight's talk: **Object Adapter and Class Adapter meet face to face.**

Object Adapter:

Because I use composition I've got a leg up. I can adapt not only an adaptee class, but any of its subclasses.

In my part of the world, we like to use composition over inheritance; you may be saving a few lines of code, but all I'm doing is writing a little code to delegate to the adaptee. We like to keep things flexible.

You're worried about one little object? You might be able to quickly override a method, but any behavior I add to my adapter code works with my adaptee class *and* all its subclasses.

Hey, come on, cut me some slack, I just need to compose with the subclass to make that work.

You wanna see messy? Look in the mirror!

Class Adapter:

That's true, I do have trouble with that because I am committed to one specific adaptee class, but I have a huge advantage because I don't have to reimplement my entire adaptee. I can also override the behavior of my adaptee if I need to because I'm just subclassing.

Flexible maybe, but efficient? No. There is just one of me, not an adapter and an adaptee.

Yeah, but what if a subclass of Adaptee adds some new behavior—then what?

Sounds messy...