Tonight's talk: **The Decorator Pattern and the Adapter Pattern discuss their differences.**

Decorator:

I'm important. My job is all about *responsibility*—you know that when a Decorator is involved, there's going to be some new responsibilities or behaviors added to your design.

That may be true, but don't think we don't work hard. When we have to decorate a big interface, whoa, that can take a lot of code.

Cute. Don't think we get all the glory; sometimes I'm just one decorator that is being wrapped by who knows how many other decorators. When a method call gets delegated to you, you have no idea how many other decorators have already dealt with it and you don't know that you'll ever get noticed for your efforts servicing the request.

Adapter:

You decorators want all the glory while us adapters are down in the trenches doing the dirty work: converting interfaces. Our jobs may not be glamorous, but our clients sure do appreciate us making their lives simpler.

Try being an adapter when you've got to bring several classes together to provide the interface your client is expecting. Now that's tough. But we have a saying: "A decoupled client is a happy client."

Hey, if adapters are doing their job, our clients never even know we're there. It can be a thankless job.

Decorator:	**Adapter:**
	But the great thing about us adapters is that we allow clients to make use of new libraries and subsets without changing *any* code; they just rely on us to do the conversion for them. Hey, it’s a niche, but we’re good at it.
Well, us decorators do that as well, only we allow *new behavior* to be added to classes without altering existing code. I still say that adapters are just fancy decorators—I mean, just like us, you wrap an object.	
	No, no, no, not at all. We *always* convert the interface of what we wrap; you *never* do. I’d say a decorator is like an adapter; it's just that you don’t change the interface!
Uh, no. Our job in life is to extend the behaviors or responsibilities of the objects we wrap; we aren’t a *simple pass through*.	
	Hey, who are you calling a simple pass through? Come on down and we’ll see how long *you* last converting a few interfaces!
Maybe we should agree to disagree. We seem to look somewhat similar on paper, but clearly we are *miles* apart in our *intent*.	
	Oh yeah, I’m with you there.