

*adapter pattern defined***Sharpen your pencil**

Let's say we also need an Adapter that converts a Duck to a Turkey.  
Let's call it DuckAdapter. Write that class:

How did you handle the fly method (after all we know ducks fly longer than turkeys)? Check the answers at the end of the chapter for our solution. Did you think of a better way?

*there are no*  
**Dumb Questions**

**Q:** How much “adapting” does an adapter need to do? It seems like if I need to implement a large target interface, I could have a LOT of work on my hands.

**A:** You certainly could. The job of implementing an adapter really is proportional to the size of the interface you need to support as your target interface. Think about your options, however. You could rework all your client-side calls to the interface, which would result in a lot of investigative work and code changes. Or, you can cleanly provide one class that encapsulates all the changes in one class.

**Q:** Does an adapter always wrap one and only one class?

**A:** The Adapter Pattern's role is to convert one interface into another. While most examples of the adapter pattern show an adapter wrapping one adaptee, we both know the world is often a bit more messy. So, you may well have situations where an adapter holds two or more adaptees that are needed to implement the target interface.

This relates to another pattern called the Facade Pattern; people often confuse the two. Remind us to revisit this point when we talk about facades later in this chapter.

**Q:** What if I have old and new parts of my system, the old parts expect the old vendor interface, but we've already written the new parts to use the new vendor interface? It is going to get confusing using an adapter here and the unwrapped interface there. Wouldn't I be better off just writing my older code and forgetting the adapter?

**A:** Not necessarily. One thing you can do is create a Two Way Adapter that supports both interfaces. To create a Two Way Adapter, just implement both interfaces involved, so the adapter can act as an old interface or a new interface.