

Facade, Decorator, Adapter, Builder

10-19-2011

Opening Discussion

- Teach for America
- Project #2 (
<https://sites.google.com/a/trinity.edu/webapps/>)

Facade

- Put a simple interface on top of a complex library.
- Most users don't need to see everything that can be done. They just need to be able to set things up and run them. The Facade does that.

Decorator

- Add functionality to an object by “decorating it”.
- Typically this involves wrapping one object inside of another that has the same supertype, but performs different functionality.
- The `java.io` library is a great example of this.
- Scala mix-in traits can take this to a different level.

Adapter

- Old code has the wrong interface. Wrap it in an adapter with the interface you need.
- This doesn't matter at all if you have full control over the code. Might be avoidable if you have full control over one side.
- Really needed when you have to work with two pieces of code you can't completely change.

Builder

- Abstract the construction of a complex object.
- Have an interface that has methods for the different steps in construction process.
- Concrete implementations do the right thing for those steps.

Closing Comments

- Cool stuff in the news?