Steven Birkner

IS 217 Project 1

Website: Stevenbirkner.com/IS217

Github: <https://github.com/SteveBirkner>

For this project we needed to implement six different design patterns and write tests for each. A design pattern is a standard solution to a common problem in software. Design patterns help make your code more readable and understandable by others in the industry. The six design patterns I chose for this project were the constructor, decorator, iterator, singleton, … The two design patterns I chose to work together were the constructor and decorator.

The first design pattern I chose to use was the constructor because it is one of the most basic and important. The constructor is used to create objects, give them default values and define functions for them. In object oriented programming the constructor provides that basic blueprint for your objects.

The next design patter I used was the decorator because it is also the pattern I used to work with the constructor pattern. The decorator pattern is an alternate to object sub classing that makes your code reusable. It allows you to add additional functionality to an existing object. In my code I have a phone object class and than I decorate that as a cell phone and gave it functions to set specifics as to what model and how much storage it has.

Another important design pattern is the singleton because this pattern lets you reuse the same instantiation of an object if it already exists or create one if there isn’t one. This is important because you can save memory and processing power if you only need one instance of an object. One example would be a database connection in that you don’t want to keep opening a new connection each time you need to send or request things from the database. With the singleton you call the object it makes the connection if one wasn’t already make and than if you need to connect again you use the same object to do so.

One of the behavioral design patterns I chose was the iterator. An iterator is a pattern that allows you perform a certain task many times to an object. The example I chose was having the jQuery function each function to append list items to the unordered list from an array.

The composite design pattern is a pattern that says you can treat a group of objects the same way. It allows you to address all objects in a uniform manner. The example I chose was the addClass jQuery function which lets you treat all the elements you chose as a collection and uniformly apply the class change to all of them.

Finally the last pattern I chose was the observer pattern. The observer pattern observers an event happening, in my case the page loading, and than performs an action when that event happens. I chose to show this using the .on jQuery function and have it add the sentence at the bottom when the page loads.