Philip Pesic

Week 17

December 11 2022

Week 17 Summary

1:

Design patterns are general solutions to general problems in programming. They are highly adaptable, and widely applicable. Often, design patterns will help streamline code and maintain understandability and readability. Design patterns are tested techniques, and they can be successful when applied to a number of general and specific problems.

2:

Prototype - A fully initialized instance to by cloned

Builder - Separates object construction from its representation

Singleton - A class where only one instance can exist

Factory - Object creation is separate from the client

Proxy - Object representing another object

Decorator - Adds responsibilities to objects dynamically or at runtime

Facade - A class that represents an entire subsystem

Adapter - Matching interfaces of different classes

Flyweight - Sharing instances of classes for efficiency

Chain - A way of passing a request through a chain of objects

Iterator - Sequentially access elements of a collection

State - Alter an object’s behavior when it changes state

Strategy - Encapsulates an algorithm in a class

Philip Pesic

Week 17

December 11 2022

Week 17 Summary

Observer - A way of notifying a change to several classes

Visitor - Defines a new operation to a class without making changes

Command - Encapsulates a command as an object

Memento - Capture and restore an object's state

Mediator - Defines simple communication between classes

3:

Anthony Ferrara claims that design patterns are for dummies. To be more specific, he claims that design patterns do not properly teach object oriented design principles, but rather they provide easy solutions for programmers who are struggling to solve a problem. Ferrara also believes that some design patterns are too general. When comparing the diagrams of Adapters, Facades, Bridges, Decorators, and Proxies, they all looked nearly indistinguishable. Essentially, because all of these patterns just control information flow, many of them are repetitive, unnecessary, and plain inefficient when compared to other solutions.