SOLID And Design Patterns for Mere Mortals

Philip Japikse (@skimedic)

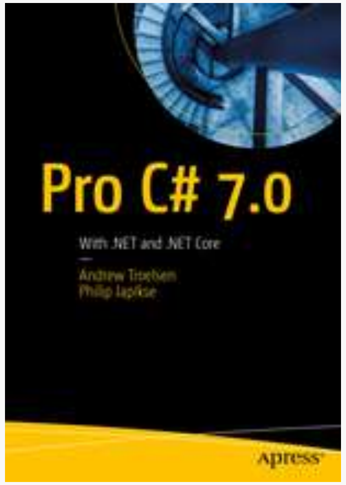
Learn. Imagine. Build.

.NET Conf

Phil.About()

- Consultant, Coach, Author, Teacher
  - Lynda.com (http://bit.ly/skimediclyndacourses)
  - Apress.Com (http://bit.ly/apressbooks)
- Microsoft MVP, ASPInsider, MCSD, CSM, CSP
- Founder, Agile Conferences, Inc
  - (http://www.dayofagile.org)
- President, Cincinnati .NET Users' Group

Building Web Applications with Visual Studio 2017
Using .NET Core and Modern JavaScript Frameworks
Philip Japikse
Kevin Grossnicklaus
Ben Dewey
Apress



A Look at SOLID

Single Responsibility Principle

- Do one thing and do it well!

http://joshlinkner.com/images/2012/05/SAN.jpg

Open Closed Principle

- Be Open for Extension, Closed for Modification

http://www.wellgolly.com/images/WWTT_house.jpg

Liskov Substitution Principle

- Derived Classes Can Stand In for Base Classes

http://beerimages.com/wp-content/uploads/2011/03/beer-collection.jpg

Interface Segregation Principle

- Make Interfaces Fine Grained and Client Specific

Dependency Inversion

- Depend On Abstractions, Not Concrete Implementations

Additional Considerations

Don’t Repeat Yourself (DRY)

- Clip-board Inheritance is an anti-pattern!

The Boy Scout Principle

- Clean up after yourself
- Clean up after others

YAGNI

- You Ain’t Gonna Need It

http://www.k-photography.info/srvgdata-gold-plated-toilets.asp

Separation Of Concerns

- Focusing one's attention upon some aspect – Edsger Dijkstra

https://sf.curbed.com/2017/3/10/14889950/kitchen-bathroom-sf-combined-toilet

Design Patterns

Motivation for Design Patterns

“The goal is not to bend developers to the will of some specific patterns, but to get them to think about their work and what they are doing”

--Phil Haack

What are Design Patterns?

- General Reusable Solutions To A Common Problem
- Conceptual
- Defined by Purpose and Structure
- Method of Communication
- Support SOLID development
- NOT CODE!

Types of Design Patterns

- Creational
  - Deal with instantiation of objects (Singleton, Factories, Prototype)
- Structural
  - Deal with Composition and Relations (Adapter, Façade, Decorator)
- Behavioral
  - Deal with responsibilities and communication between objects (Command, Strategy, Observer, Pub-Sub, Memento, Template Method)

Creational Design Patterns

Creational

- Singleton
  - Ensures class has only one instance with a single access point
- Simple Factory (Not a “true” pattern)
  - Encapsulates class creation in one place
- Factory Method
  - Uses methods to create objects without specifying the exact class
- Abstract Factory
  - Encapsulates a group of individual factories with a common theme without specifying their concrete class
- Prototype
  - Clones an instance to make more instances – usually for performance reasons

Structural Design Patterns

Structural

- Adapter
  - Converts the interface of a class into another interface the client expects
- Façade
  - Provides a simplified interface to a larger body of code
- Decorator
  - Attaches additional responsibilities to an object at runtime without effecting othe objects of the same class

Behavioral Design Patterns

Behavioral

- Command
  - Encapsulates a request as an object
- Strategy
  - Encapsulates an algorithm inside a class
- Observer/Pub-Sub
  - Messaging patterns (see next slide)
- The Memento Pattern
  - Used to restore an object to its previous state
- The Template Method
  - Defines skeleton of an algorithm , deferring some steps to sub classes

Observer vs Pub-Sub

Observer

- Sender and recipients know each other
- Send and receive one at a time
- Direct communication

Pub-Sub

- Sender and recipients unknown to each other
- Send once, every subject receives.
- Intermediary handles filtering and routing

Resources

- "Design Patterns: Elements of Reusable Object Oriented Design"
  - Eric Gamma, Richard Helm, Ralph Johnson, John Vlissides
- "Head First Design Patterns"
  - Freeman, Robson, Bates, Sierra
- Eight part series with Robert Green on Visual Studio Toolbox
  - https://channel9.msdn.com/Shows/Visual-Studio-Toolbox/Design-Patterns-CommandMemento (first of the series)

Patterns in C#

Me.Contact()

Thank You!

- Online
  - skimedic@outlook.com
  - www.skimedic.com/blog
  - www.twitter.com/skimedic
- Courses/Books
  - http://bit.ly/skimediclyndacourses
  - http://bit.ly/apressbooks
- Get the code and slide deck:
  - **https://github.com/skimedic/presentations**/tree/master/Patterns

Questions?