

# SOLID SOFTWARE & DESIGN PATTERNS FOR MERE MORTALS

Philip Japikse (@skimedic)

skimedic@outlook.com

[www.skimedic.com/blog](http://www.skimedic.com/blog)

Microsoft MVP, ASPInsider, MCSD, MCDBA, CSM, CSP  
Principal Consultant/Architect, Strategic Data Systems



Phil.About()

🏠 Principal Consultant/Architect, Strategic Data Systems

🏠 <http://www.sds-consulting.com>

🏠 Developer, Coach, Author, Teacher

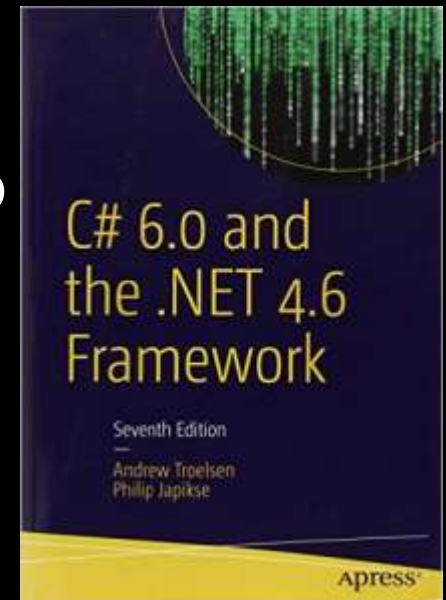
🏠 [http://bit.ly/pro\\_csharp](http://bit.ly/pro_csharp)

🏠 Microsoft MVP, ASPInsider, MCSD, MCDBA, CSM, CSP


🏠 Founder, Agile Conferences, Inc.

🏠 <http://www.dayofagile.org>

🏠 President, Cincinnati .NET User's Group



# Hallway Conversations Podcast

 Hosted by Phil Japikse, Steve Bohlen, Lee Brandt, James Bender

 Website: [www.hallwayconversations.com](http://www.hallwayconversations.com)

 iTunes: [http://bit.ly/hallway\\_convo\\_itunes](http://bit.ly/hallway_convo_itunes)

 Feed Burner: [http://bit.ly/hallway\\_convo\\_feed](http://bit.ly/hallway_convo_feed)

 Also available through Windows Store



# 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](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 Fine Grained Interfaces Client Specific





# DEPENDENCY INVERSION

Depend On Abstractions, Not Concrete Implementations





DON'T REPEAT YOURSELF

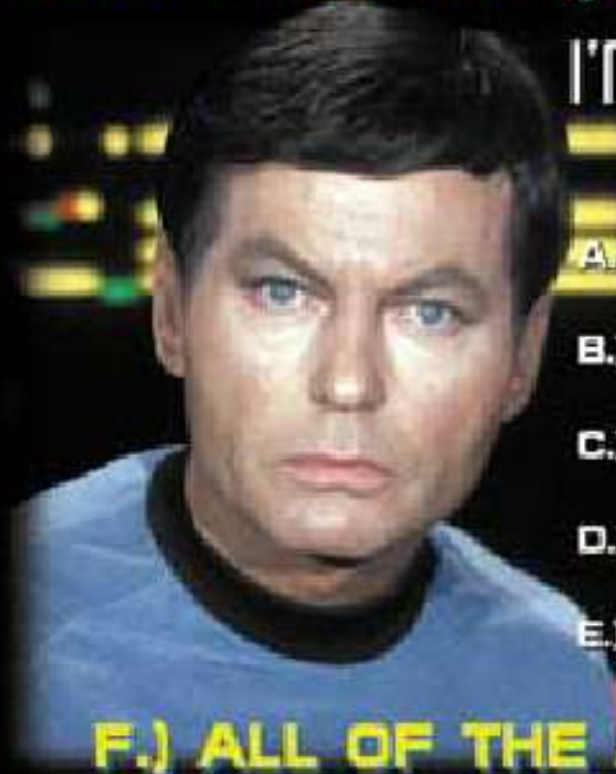


# THE BOY SCOUT PRINCIPLE



# SEPARATION OF CONCERNS

## DAMMIT, JIM!



I'M A DOCTOR,  
**NOT...**

A.) A BRICKLAYER

B.) AN ESCALATOR

C.) A MECHANIC

D.) AN ENGINEER

E.) A COAL MINER

**F.) ALL OF THE ABOVE  
AND THEN SOME**

© 1979 Paramount Pictures. All Rights Reserved.



“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)

## Structural

 Deal with Composition and Relations (Adapter, Decorator, Façade)

## Behavioral

 Deal with responsibilities and communication between objects  
(Command, Strategy)

# CREATIONAL

## Singleton

 Ensures class has only one instance with a single access point

## Simple Factory (Not a “true” pattern)

 Encapsulates Object Creation in one place

## Factory Method

 Provides an interface to allow subclasses to determine what is instantiated

## Abstract Factory

 Provides an interface for creating families of related objects without specifying their concrete class

# STRUCTURAL

## Adapter

 Converts the interface of a class into another interface the client expects

## Façade

 Provides a unified interface to a set of interfaces

## Decorator

 Attaches additional responsibilities to an object at runtime



# BEHAVIORAL

 Command

 Encapsulates a request as an object

 Strategy

 Encapsulates an algorithm inside a class

# RESOURCES

 “Design Patterns: Elements of Reusable Object Oriented Design”

 Eric Gamma, Richard Helm, Ralph Johnson, John Vlissides

 “Head First Design Patterns”

 [www.dofactory.com](http://www.dofactory.com)

# DEMO DEWNO

Patterns in C#

# Questions?





## Contact Me

[phil@sds-consulting.com](mailto:phil@sds-consulting.com)

[www.sds-consulting.com](http://www.sds-consulting.com)

[www.skimedic.com/blog](http://www.skimedic.com/blog)

[skimedic@outlook.com](mailto:skimedic@outlook.com)

[www.twitter.com/skimedic](https://twitter.com/skimedic)

<https://github.com/skimedic/presentations>

[www.hallwayconversations.com](http://www.hallwayconversations.com)

Thank  
You!