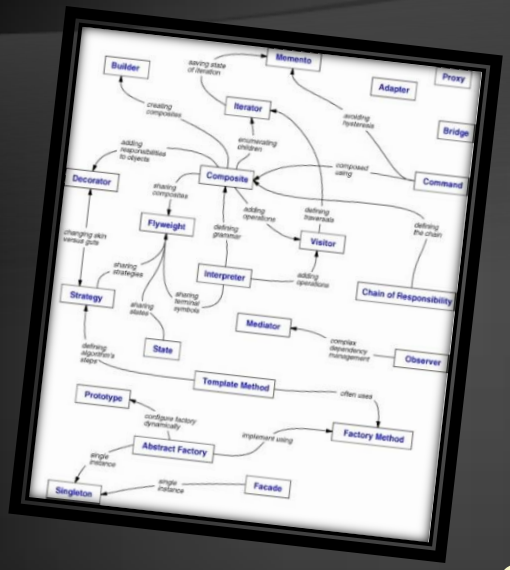
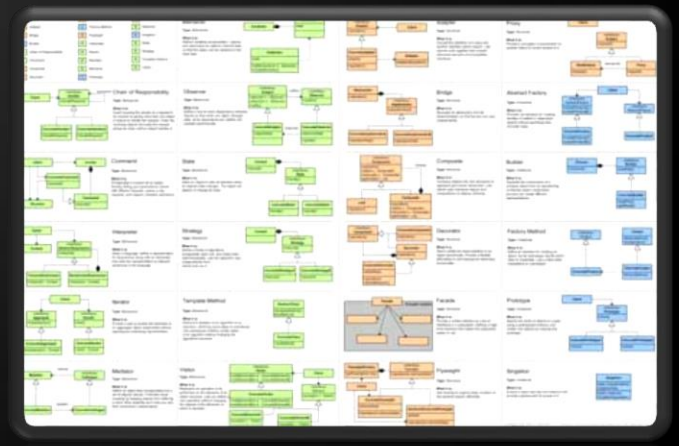
Telerik Academy



Introduction to Design Patterns

General and reusable solutions to common problems in software design



High-Quality Code

Telerik Software Academy

http://academy.telerik.com

Telerik Academy

What is a Design Pattern?

Name, Problem, Solution and Consequences

Telerik Academy

What Design Patterns Are?

- General, repeatable and reusable solution to a common problem in software design
- Problem/solution pairs within a given context
  - Aren't a finished design that can be transformed directly into code
  - Not code snippets
  - Not algorithms
  - Not components/libraries
- Templates for solving certain OOP problems
- With names to identify and talk about them

Telerik Academy

What Design Patterns Are? (2)

- What do patterns deal with?
  - Application and system design
  - Abstractions on top of code (code structure)
  - Relationships between classes or other collaborators
  - Problems that someone already solved
- Can speed up the development process by providing tested, proven development paradigms

Origins of Design Patterns

> "Each pattern describes a problem which occurs over and over again in our environment and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it in the same way twice".

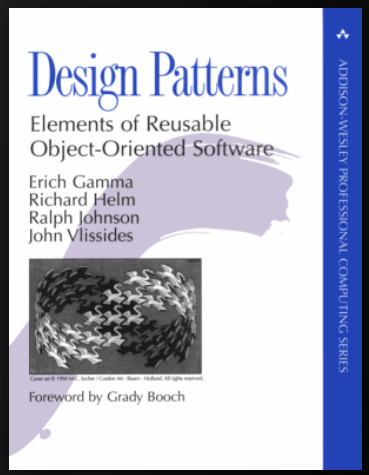
- Christopher Alexander
  - Very successful architect
  - A Pattern Language: Towns, Buildings, Construction, 1977

- Context:
  - City Planning and Building architectures

Telerik Academy

Origins of Design Patterns (2)

- Search for recurring successful designs
  - Emergent designs from practice
- Supporting higher levels of design reuse is quite challenging
- Design Patterns: Elements of Reusable Object-Oriented Software
  - Gama, Helm, Johnson, Vlissides, 1994
    - Came to be known as the "Gang of Four."



Describing Design Patterns

- Graphical notation is generally not sufficient
- In order to reuse design decisions the alternatives and trade-offs that led to the decisions are critical knowledge
- Concrete examples are also important
- The history of the why, when, and how set the stage for the context of usage

Pattern name
Motivation
Structure
Collaborations
Applicability
Intent
Participants
Known Uses
Implementation
Sample Code
Also Known As
Consequences
Related Patterns

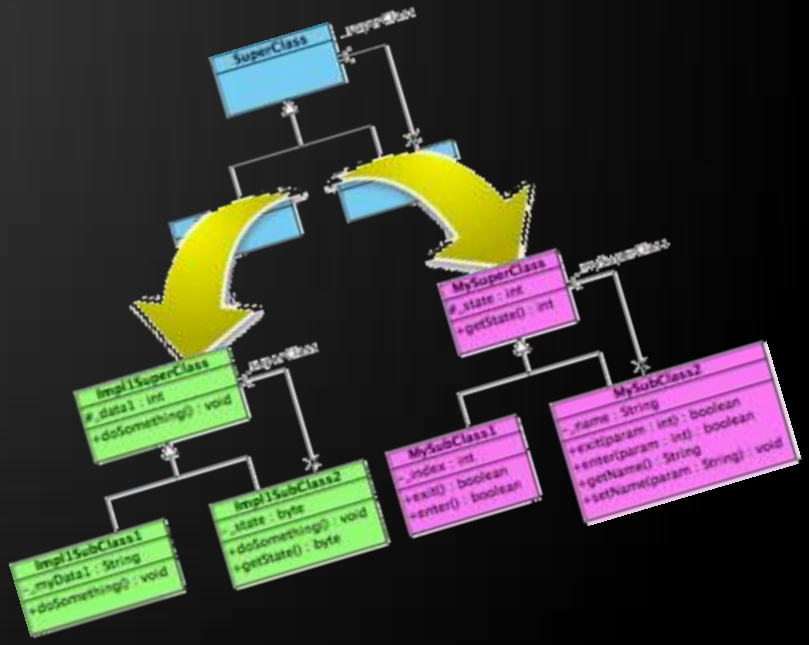
Telerik Academy

Elements of Design Patterns

- Design patterns have four essential elements:
  - Pattern Name
    - Increases vocabulary
  - Problem
    - Intent, context, when to apply
  - Solution
    - UML-like structure, abstract code
  - Consequences
    - Results and tradeoffs

Telerik Academy

Why Design Patterns?



Benefits of Design Patterns

- Improves system and application design
- Design patterns enable large-scale reuse of software architectures
  - Help document how systems work
- Patterns explicitly capture expert knowledge and design trade-offs
  - Provide a starting point for a solution
- Can speed production in a team
  - Patterns help improve developer communication (shared language)
  - Pattern names form a common vocabulary

When to Use Patterns?

- Solutions to problems that recur with variations
  - No need for reuse if problem only arises in one context
- Solutions that require several steps:
  - Not all problems need all steps
  - Patterns can be overkill if solution is a simple linear set of instructions!
- Do not use patterns when not required
  - Overdesign is evil!

Telerik Academy

Drawbacks of Design Patterns

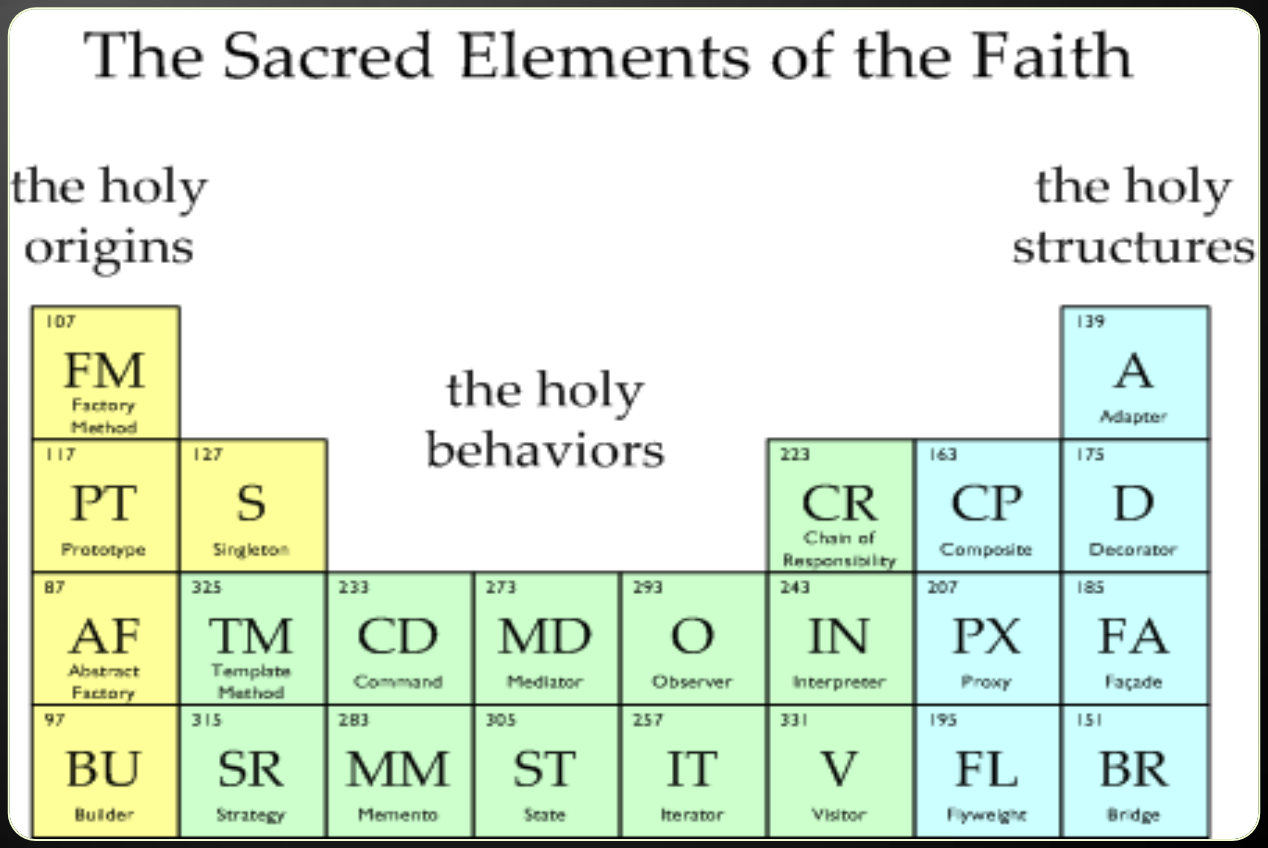
- Patterns do not lead to a direct code reuse
- Patterns are deceptively simple
- Teams may suffer from pattern overload
- Patterns are validated by experience and discussion rather than by automated testing
- Integrating patterns into the software development process is a human-intensive activity
- Use patterns if you understand them well

Criticism of Design Patterns

- Targets the wrong problem
  - The design patterns may just be workarounds of some missing features of a given language
- Lacks formal foundations
  - The study of design patterns has been excessively ad-hoc
- Leads to inefficient solutions
- Does not differ significantly from other abstractions

Telerik Academy

Types of Design Patterns



Three Main Types of Patterns

- Creational patterns
  - Deal with initializing and configuring classes and objects
- Structural patterns
  - Describe ways to assemble objects to implement a new functionality
  - Composition of classes or objects
- Behavioral patterns
  - Deal with dynamic interactions among societies of classes and objects
  - How they distribute responsibility

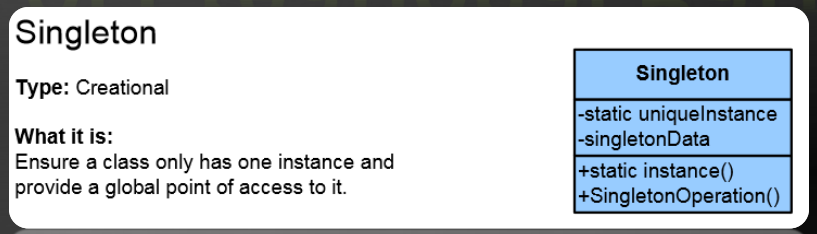
Creational Patterns

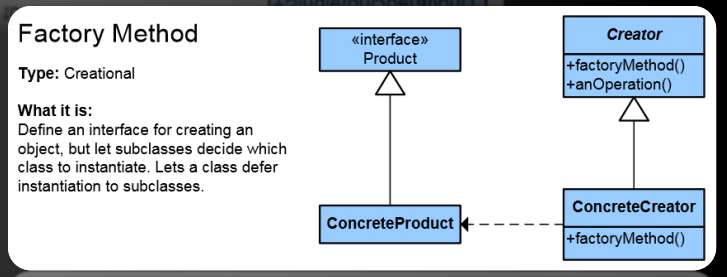
- Deal with object creation mechanisms
- Trying to create objects in a manner suitable to the situation
- Composed of two dominant ideas
  - Encapsulating knowledge about which concrete classes the system uses
  - Hiding how instances of these concrete classes are created and combined

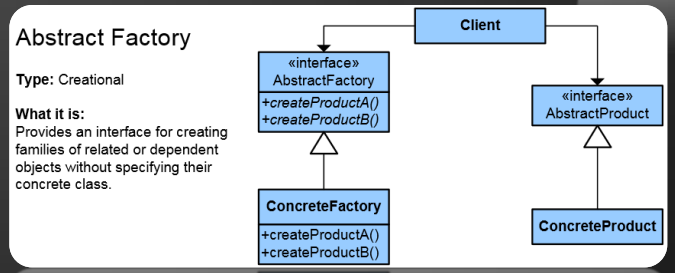
Telerik Academy

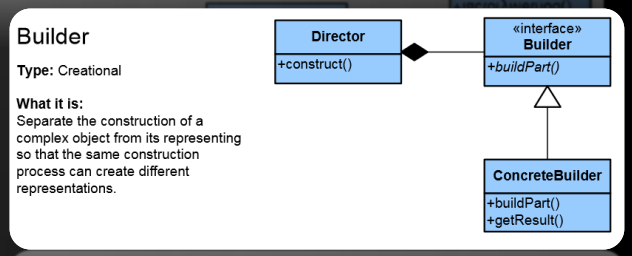
List of Creational Patterns

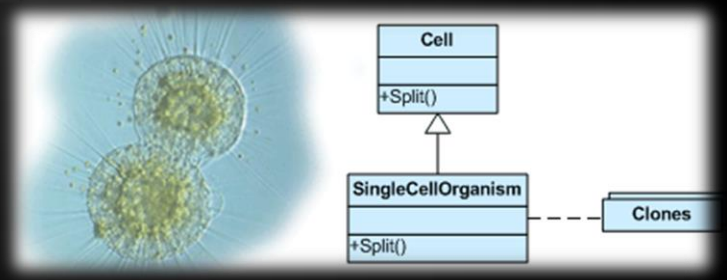
- Singleton
- Simple Factory
- Factory Method
- Abstract Factory
- Builder
- Prototype
- Fluent Interface
- Object Pool
- Lazy initialization











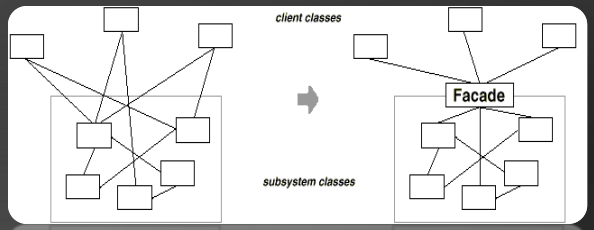
Structural Patterns

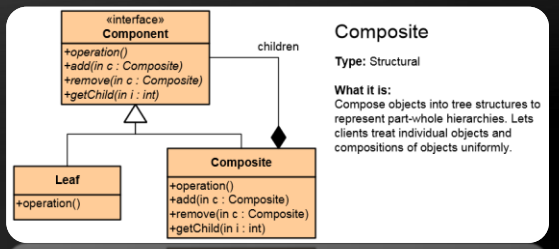
- Describe ways to assemble objects to implement a new functionality
- Ease the design by identifying a simple way to realize relationships between entities
- This design patterns is all about Class and Object composition
  - Structural class-creation patterns use inheritance to compose interfaces
  - Structural object-patterns define ways to compose objects to obtain new functionality

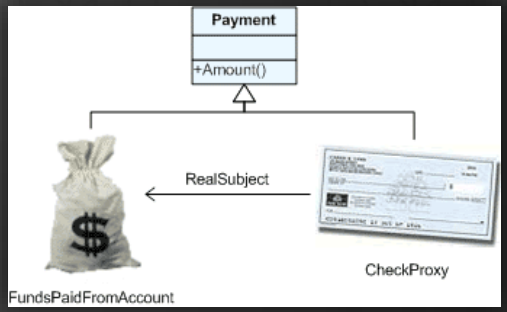
Telerik Academy

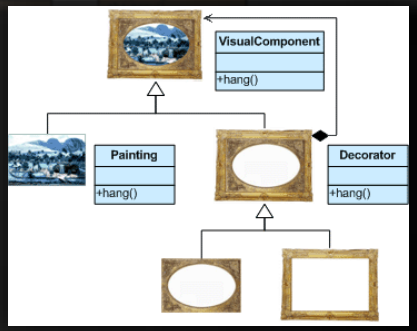
List of Structural Patterns

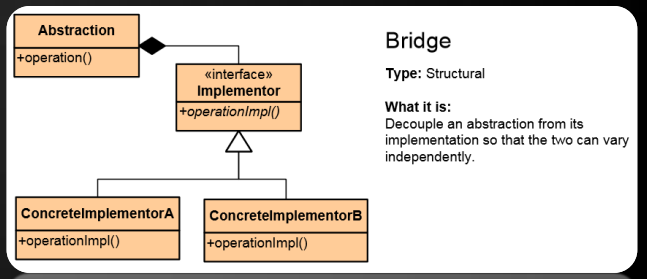
- Façade
- Composite
- Flyweight
- Proxy
- Decorator
- Adapter
- Bridge











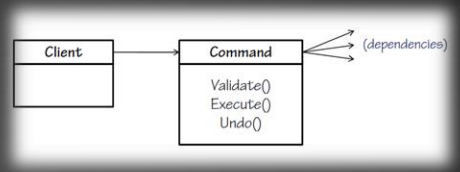
Behavioral Patterns

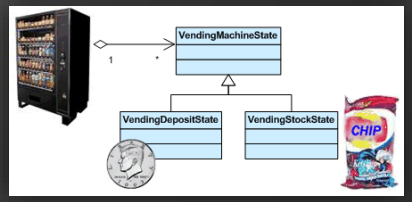
- Concerned with communication (interaction) between the objects
  - Either with the assignment of responsibilities between objects
  - Or encapsulating behavior in an object and delegating requests to it
- Increase flexibility in carrying out cross-classes communication

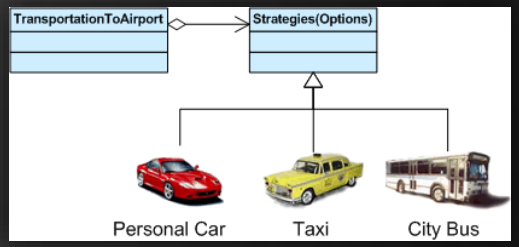
Telerik Academy

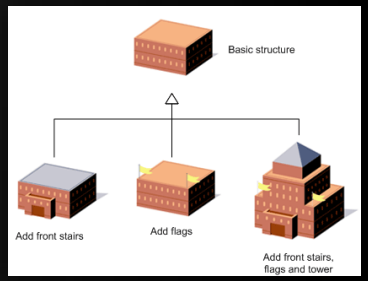
List of Behavioral Patterns

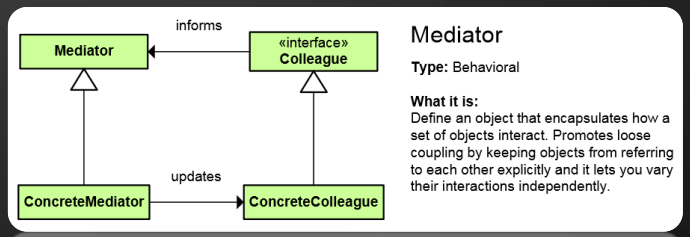
- Chain of Responsibility
- Iterator
- Command
- Template Method
- Strategy
- Observer
- Mediator
- Memento
- State
- Interpreter
- Visitor

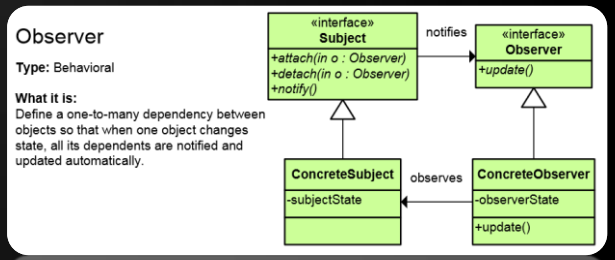


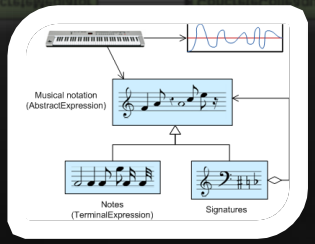


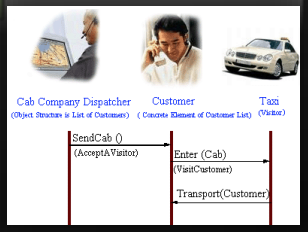










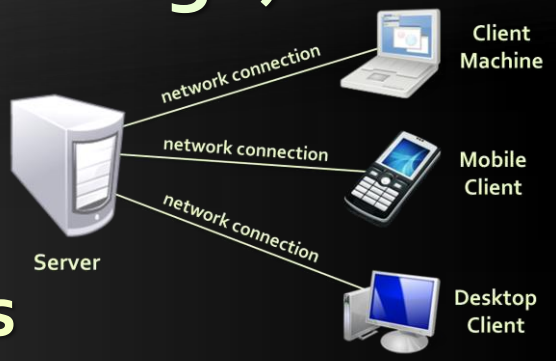


Concurrency patterns

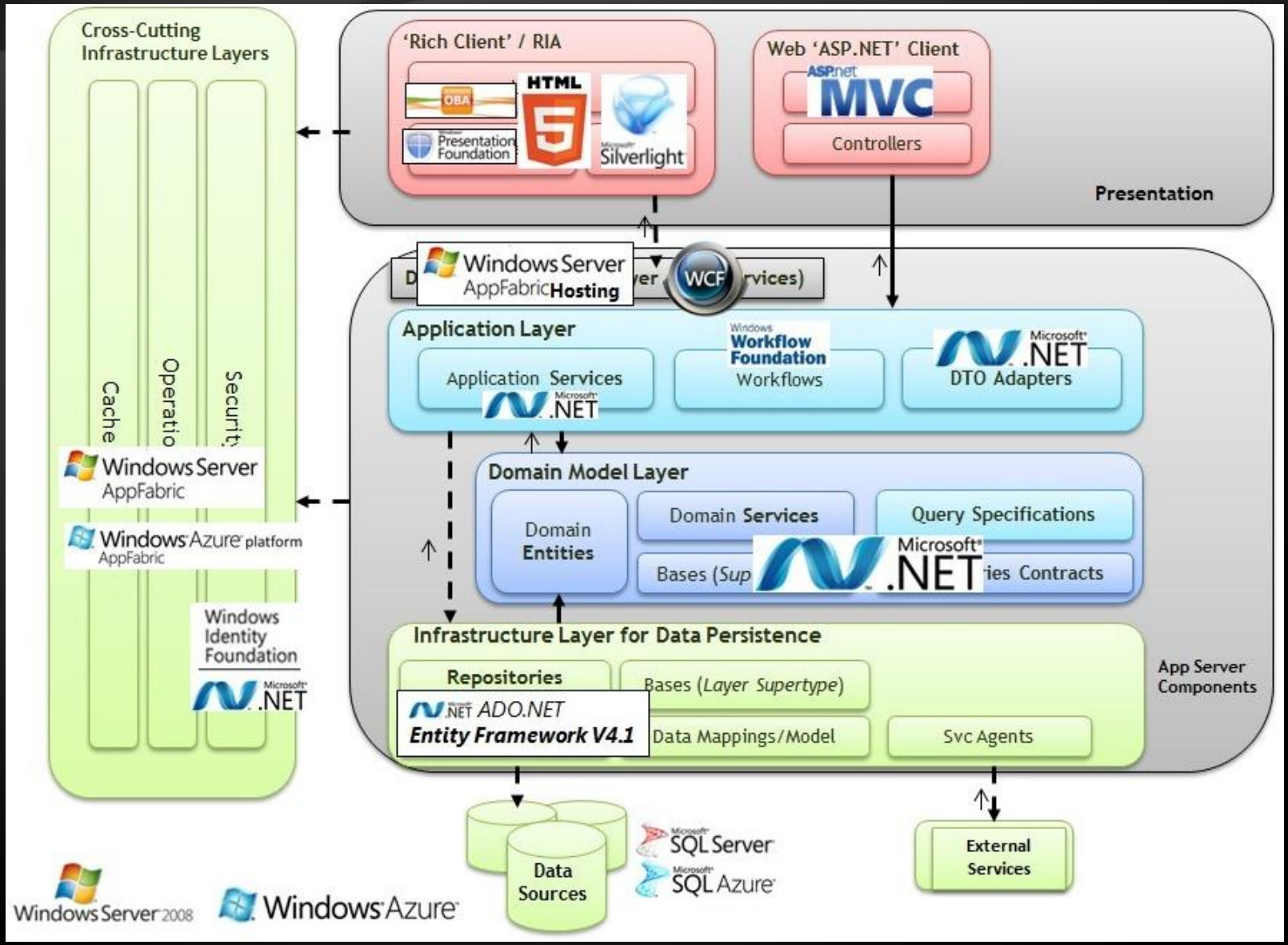
- Active Object
- Double Checked Locking pattern
- Monitor Object
  - An object to can be safely used by many threads
- Read-Write Lock pattern
- Thread Pool pattern
  - A number of threads are created to perform a number of tasks
- And many more

Architectural Patterns

- Architectural patterns (systems design)
  - Client-Server Architecture
  - 3-Tier Architecture
    - Client, Business and Data layers
  - Multi-Tier Architecture
  - MVC (Model-View-Controller)
  - MVP (Model-View-Presenter)
  - MVVM (Model-View-ViewModel)
  - SOA (Service-Oriented Architecture)
    - Using reusable building blocks

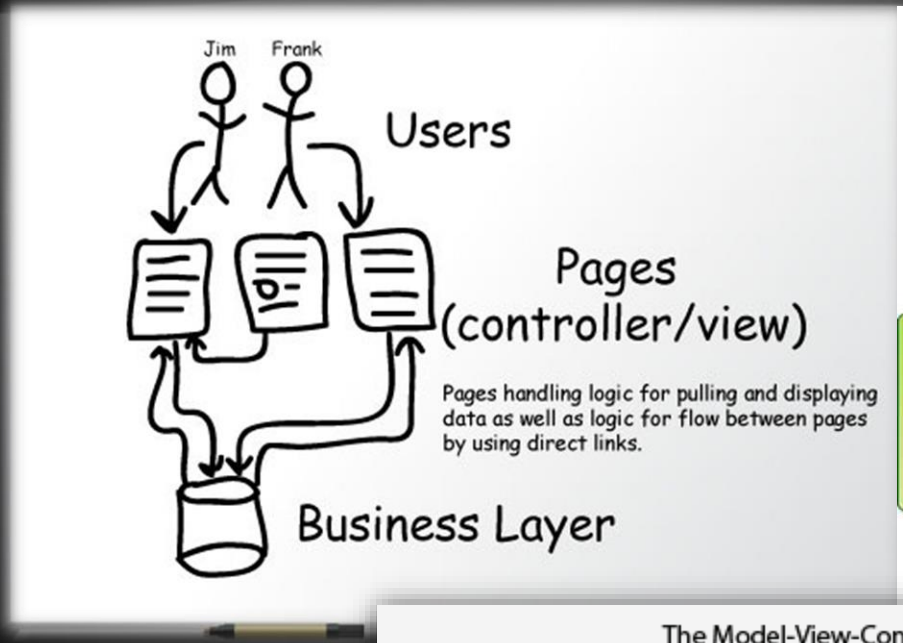


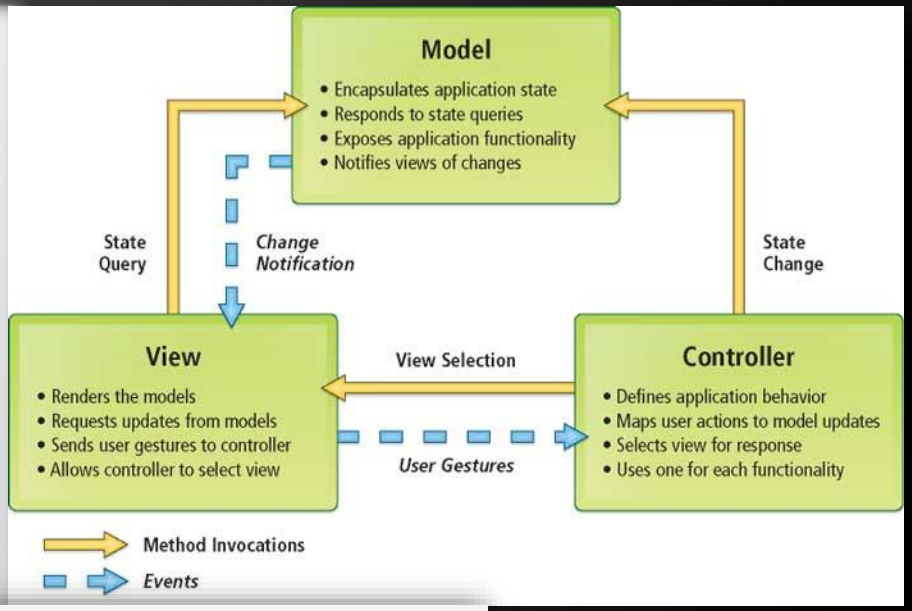
Multi-Tier Architecture

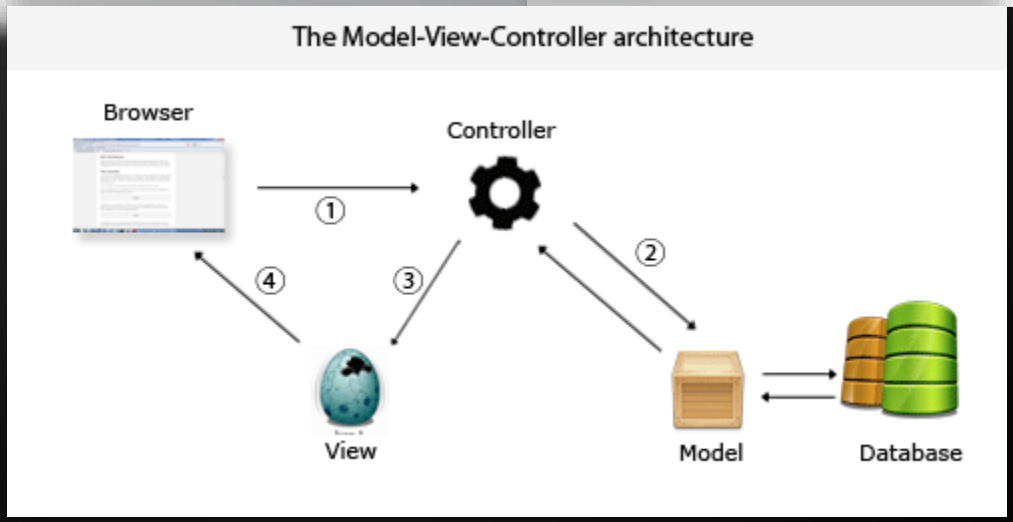


Telerik Academy

Model-View-Controller (MVC)

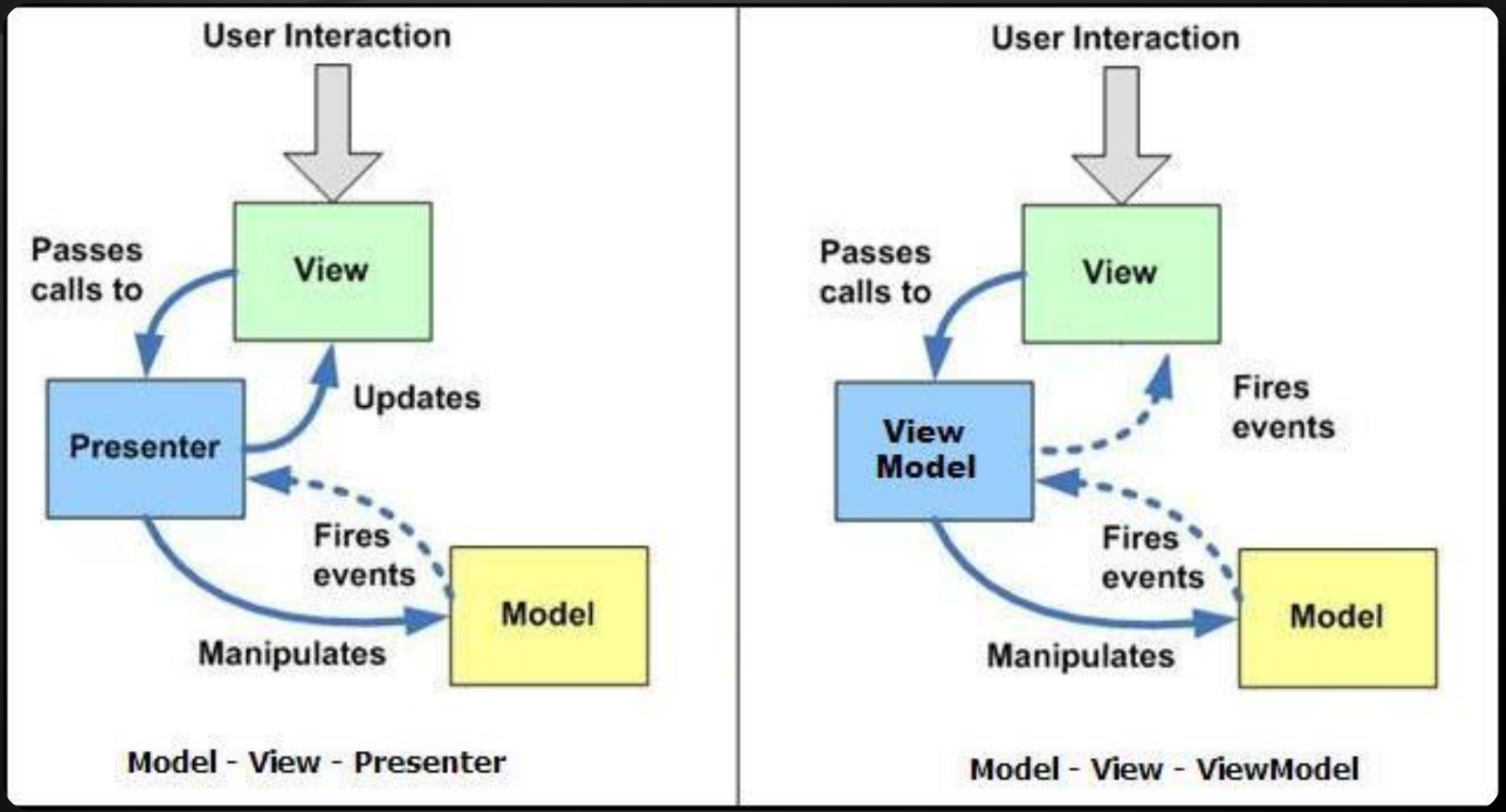






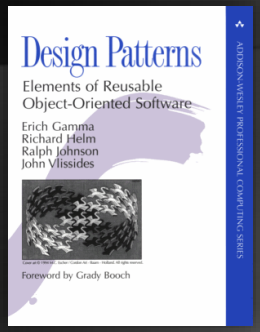
Telerik Academy

MVP vs. MVVM Patterns

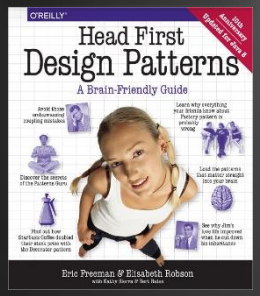


- MVVM is like MVP but leverages the platform's build-in bi-directional data binding mechanisms

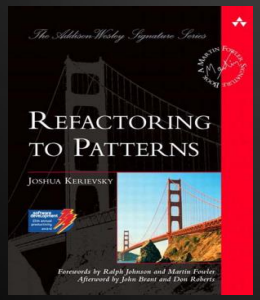
Recommended Books



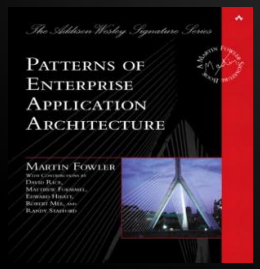
Design Patterns: Elements of Reusable Object-Oriented Software, By The Gang of Four, 1994



Head First Design Patterns, By Eric Freeman, Elisabeth Robson, Bert Bates, Kathy Sierra, 2004



Refactoring to Patterns, By: Joshua Kerievsky, 2004



Patterns of Enterprise Application Architecture, Martin Fowler, 2002

Telerik Academy

Introduction to Design Patterns

http://academy.telerik.com

Telerik Academy

Free Trainings @ Telerik Academy

- C# Programming @ Telerik Academy
  - csharpfundamentals.telerik.com

- Telerik Software Academy
  - academy.telerik.com

- Telerik Academy @ Facebook
  - facebook.com/TelerikAcademy

- Telerik Software Academy Forums
  - forums.academy.telerik.com