

Patterns in the real world

- A pattern is a solution to a problem in a context.
- Some tips to improve yourself:

Study patterns catalog(we are going to see it soon)

Do your homework.

Take time to reflect and evaluate.

Get your ideas down on paper/code for others.

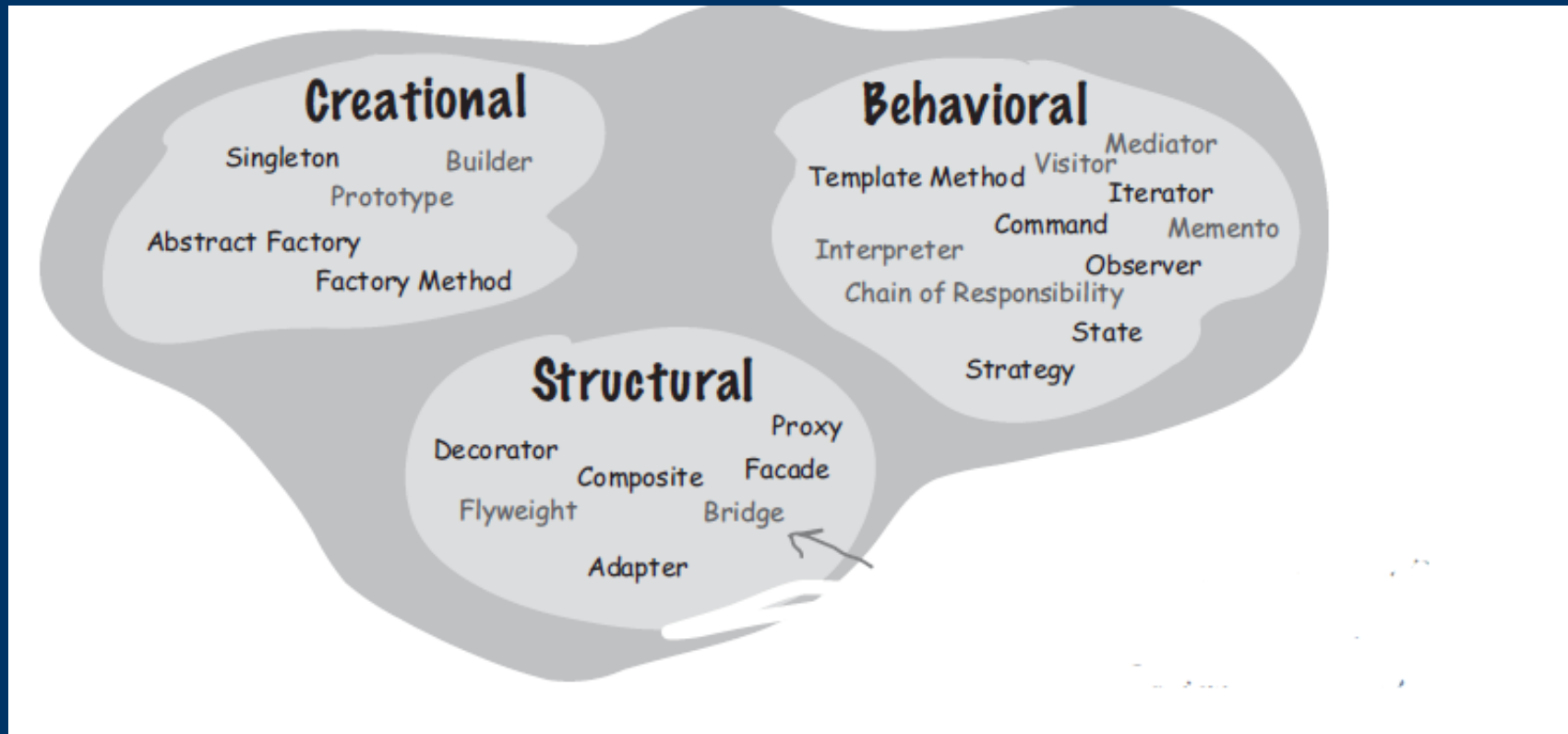
If you are willing to create your own, don't forget the rule of three.

Pattern	Description
Decorator	Wraps an object and provides a different interface to it.
State	Subclasses decide how to implement steps in an algorithm.
Iterator	Subclasses decide which concrete classes to create.
Facade	Ensures one and only object is created.
Strategy	Encapsulates interchangeable behaviors and uses delegation to decide which one to use.
Proxy	Clients treat collections of objects and individual objects uniformly.
Factory Method	Encapsulates state-based behaviors and uses delegation to switch between behaviors.
Adapter	Provides a way to traverse a collection of objects without exposing its implementation.
Observer	Simplifies the interface of a set of classes.
Template Method	Wraps an object to provide new behavior.
Composite	Allows a client to create families of objects without specifying their concrete classes.
Singleton	Allows objects to be notified when state changes.
Abstract Factory	Wraps an object to control access to it.
Command	Encapsulates a request as an object.

Patterns in the real world

- Patterns are grouped into one of a few classification schemas. The most well-known schema was are: Creational, Behavioral and Structural.
 - Creational patterns involve object instantiation and all provide a way to decouple a client from the objects it needs to instantiate.
 - Structural patterns let you compose classes or object into larger structures.
 - Bahavioral is concerned with how classes and object interact and distribute responsibility.
-
-

Patterns in the real world



Patterns in the real world

Creational	Abstract Factory (87)	families of product objects
	Builder (97)	how a composite object gets created
	Factory Method (107)	subclass of object that is instantiated
	Prototype (117)	class of object that is instantiated
	Singleton (127)	the sole instance of a class

Patterns in the real world

Structural	Adapter (139)	interface to an object
	Bridge (151)	implementation of an object
	Composite (163)	structure and composition of an object
	Decorator (175)	responsibilities of an object without subclassing
	Facade (185)	interface to a subsystem
	Flyweight (195)	storage costs of objects
	Proxy (207)	how an object is accessed; its location

Patterns in the real world

Behavioral	Chain of Responsibility (223)	object that can fulfill a request
	Command (233)	when and how a request is fulfilled
	Interpreter (243)	grammar and interpretation of a language
	Iterator (257)	how an aggregate's elements are accessed, traversed
	Mediator (273)	how and which objects interact with each other
	Memento (283)	what private information is stored outside an object, and when
	Observer (293)	number of objects that depend on another object; how the dependent objects stay up to date
	State (305)	states of an object
	Strategy (315)	an algorithm
	Template Method (325)	steps of an algorithm
	Visitor (331)	operations that can be applied to object(s) without changing their class(es)

Patterns in the real world

- Suggestions:

Today are more patterns than in GoF book, learn about them as well.

Go for simplicity and don't became over excited. If you come up with a simpler solution without using a pattern, then go for it.

Patterns are tools not rules.

Patterns in the real world

- There are more patterns:

Application Patterns: are patterns for creating system level architecture, many multi tier architectures fall into this category.

Domain specific patterns: patterns for a specific domain like concurrent systems or real time system.

Business process patterns: describe interaction between business, client, customers and data. And used to communicate decisions.

Organizational patterns: describe structures and practice of human organizations.

User interface: focus on design interactive software programs.

Patterns in the real world

- Anti-patterns: always look like a good solution, but then turn out to be a bad solution when it is applied.