

N Design Patterns You Might Actually Use

Aja Hammerly

~~N~~5 Design Patterns You Might Actually Use

Aja Hammerly

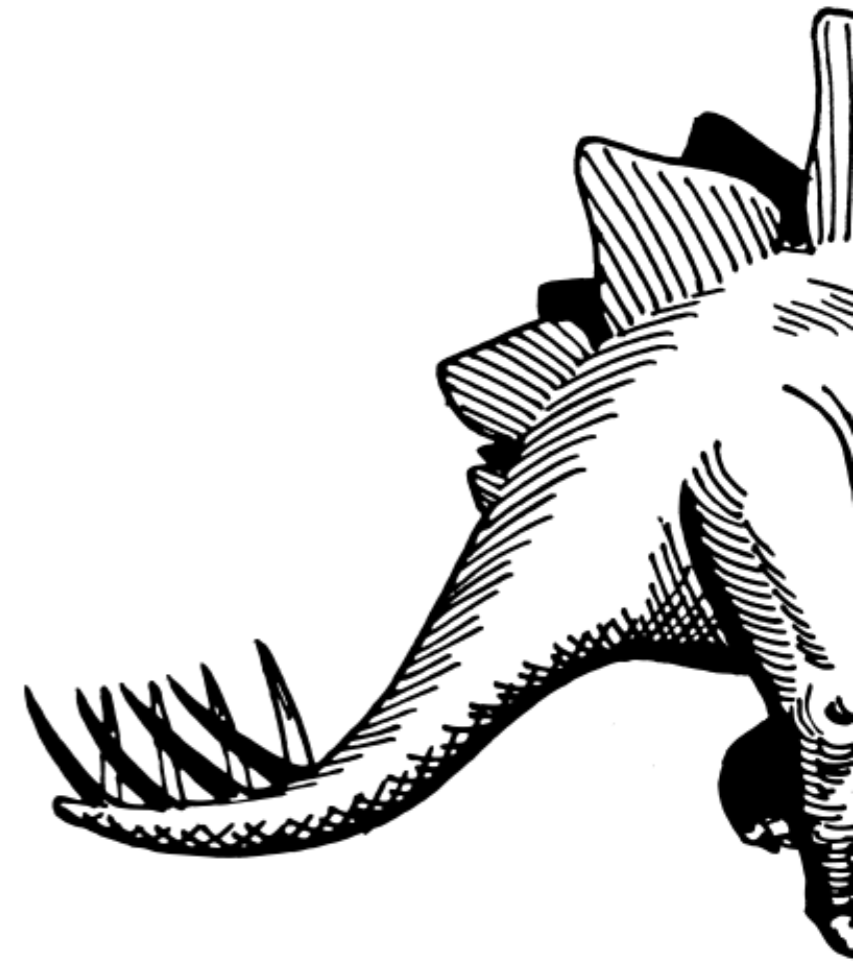
Aja Hammerly

@thagomizer_rb

<http://github.com/thagomizer>

<http://www.thagomizer.com>

THAGOMIZER



Motivations



Singleton?

Factory?

Something Else?



<https://www.flickr.com/photos/laughingsquid/7658733440/>



re5et added a note on May 8, 2013



again, looks like strategies



<https://www.flickr.com/photos/kozumel/4918575268>

Design Patterns

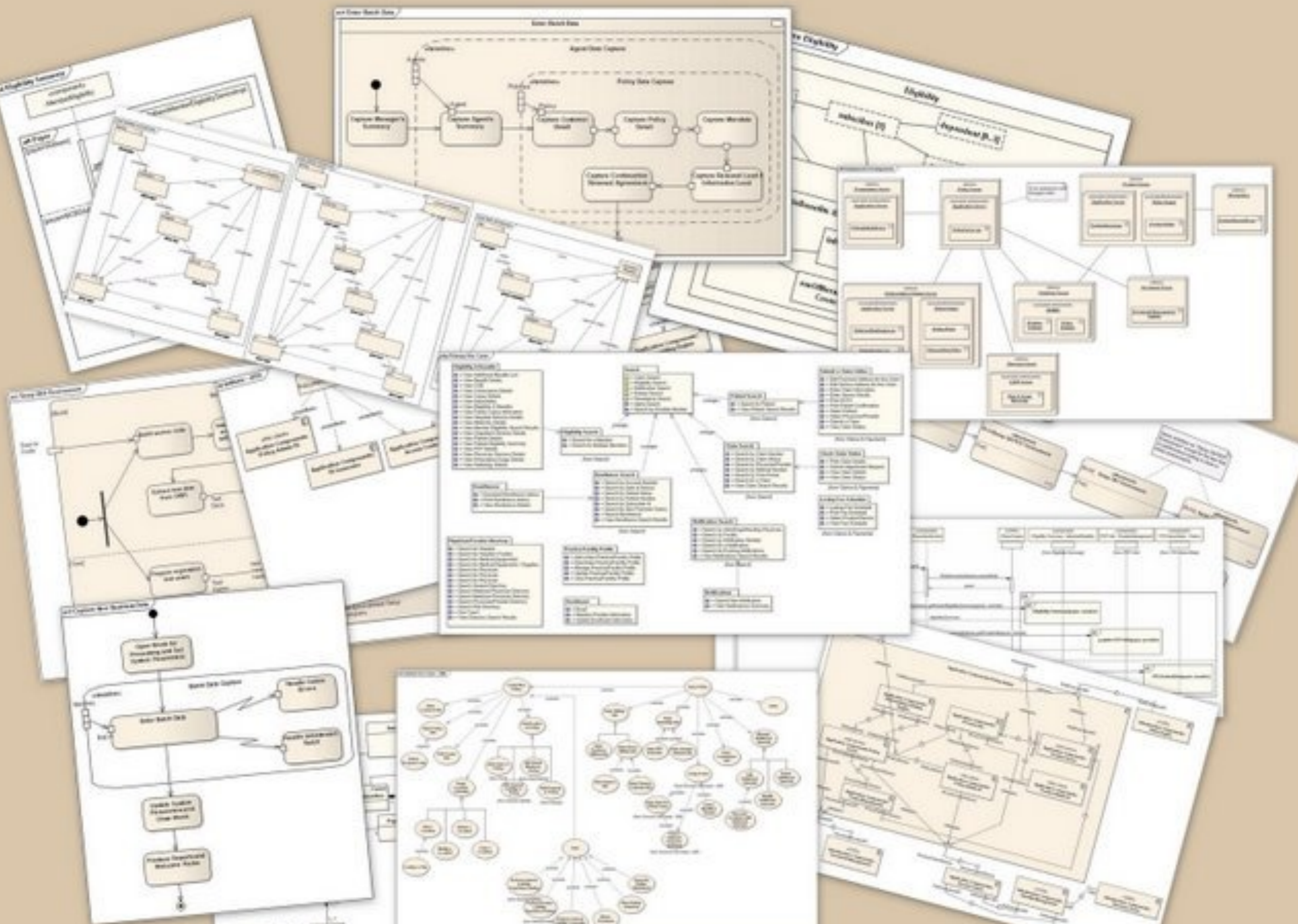
==

Common Vocabulary



<http://www.flickr.com/photos/54397539@N06/5034334027/>





RASPBERRY

IMITATION FLAVOR

Kool-Aid®

INSTANT SOFT DRINK MIX • ADD SUGAR



MAKES
2
QUARTS

K

Kool-Aid®

CUT out GAG TAG; brush off back of TAG; wear on pocket or button.



DIRECTIONS: Empty contents into large pitcher. Add 1 cup sugar. Add COLD water and ice to make 2 quarts.



Fumaric acid, sugar, artificial flavor and color, calcium carbonate, dioctyl sodium sulfosuccinate (DSS).

GENERAL FOODS CORP., WHITE PLAINS, N. Y., U. S. A.
P-8256 NET WT. 4.5 GM.

Setting Expectations

- ~140 Slides in 30 minutes
- Code Medium
- Advanced Beginner - Intermediate

Patterns

- Template
- Strategy
- Composite
- Decorator
- Command

Template

What Is It?

Outline For Similar Tasks

Examples



ETL



Extract Transform Load

Active Record Models

TPS REPORT (TOILET PAPER SUPPLY)



96 ROLLS P/BOX
4 BOXES
1000 SHEETS P/ROLL

384 ROLLS

- 4 SHEETS P/ROLL BEGINNING OF ROLL
- 2 SHEETS P/ROLL END OF ROLL

Reporting

USAGE

32 PERSONS (AVERAGE)
12 SHEETS P/WIPE
2 WIPES P/DAY

$$(32 \cdot 12 \cdot 2) = 768 \text{ SHEETS P/DAY OFFICE USAGE}$$

CAPACITY OF CURRENT INVENTORY

$$\frac{379,392}{768} = 494 \text{ DAYS}$$

PENDING
DIETARY
CHANGES

Recognizing It

```
class PastDueAccountsReport

  def extract
    @data = Accounts.past_due
  end

  def aggregate
    @data = @data.group_by { |a| a.owner }
  end

  def email
    Email.new("accounting@adequate.hq", @data)
  end

  def generate
    extract
    aggregate
    email
  end
end
```

```
class NewUsersReport

  def extract
    @data = Users.new_in_last_24_hours
  end

  def aggregate
    @data = @data.group_by { |u| u.department }
  end

  def email
    Email.new("reports@adequate.hq", @data)
  end

  def generate
    extract
    aggregate
    email
  end
end
```

```
class NewUsersReport

  def extract
    @data = Users.new_in_last_24_hours
  end

  def aggregate
    @data = @data.group_by { |u| u.department }
  end

  def email
    Email.new("reports@adequate.hq", @data)
  end

  def generate
    extract
    aggregate
    email
  end
end
```

```
class ReportTemplate
  def extract; raise "NYI"; end
  def aggregate; raise "NYI"; end
  def email; raise "NYI"; end

  def generate
    extract
    aggregate
    email
  end
end
```



```
class ReportTemplate
  def extract; raise "NYI"; end
  def aggregate; raise "NYI"; end
  def email; raise "NYI"; end

  def generate
    extract
    aggregate
    email
  end
end
```

```
class ReportTemplate
  def extract; raise "NYI"; end
  def aggregate; raise "NYI"; end
  def email; raise "NYI"; end

  def generate
    extract
    aggregate
    email
  end
end
```

```
class NewUsersReport < ReportTemplate

  def extract
    @data = Users.new_in_last_24_hours
  end

  def aggregate
    @data = @data.group_by { |u| u.department }
  end

  def email
    Email.new("reports@adequate.hq", @data)
  end

end
```

```
class PastDueAccountsReport < ReportTemplate

  def extract
    @data = Accounts.past_due
  end

  def aggregate
    @data = @data.group_by { |a| a.owner }
  end

  def email
    Email.new("accounting@adequate.hq", @data)
  end

end
```



Share Common Code

Isolate Differences

Easy To Add New Versions

Force Common Algorithm

Read multiple files

Inheritance

Inheritance



Report A

Extract 1

Transform 1

Load 1

Report B

Extract 1

Transform 2

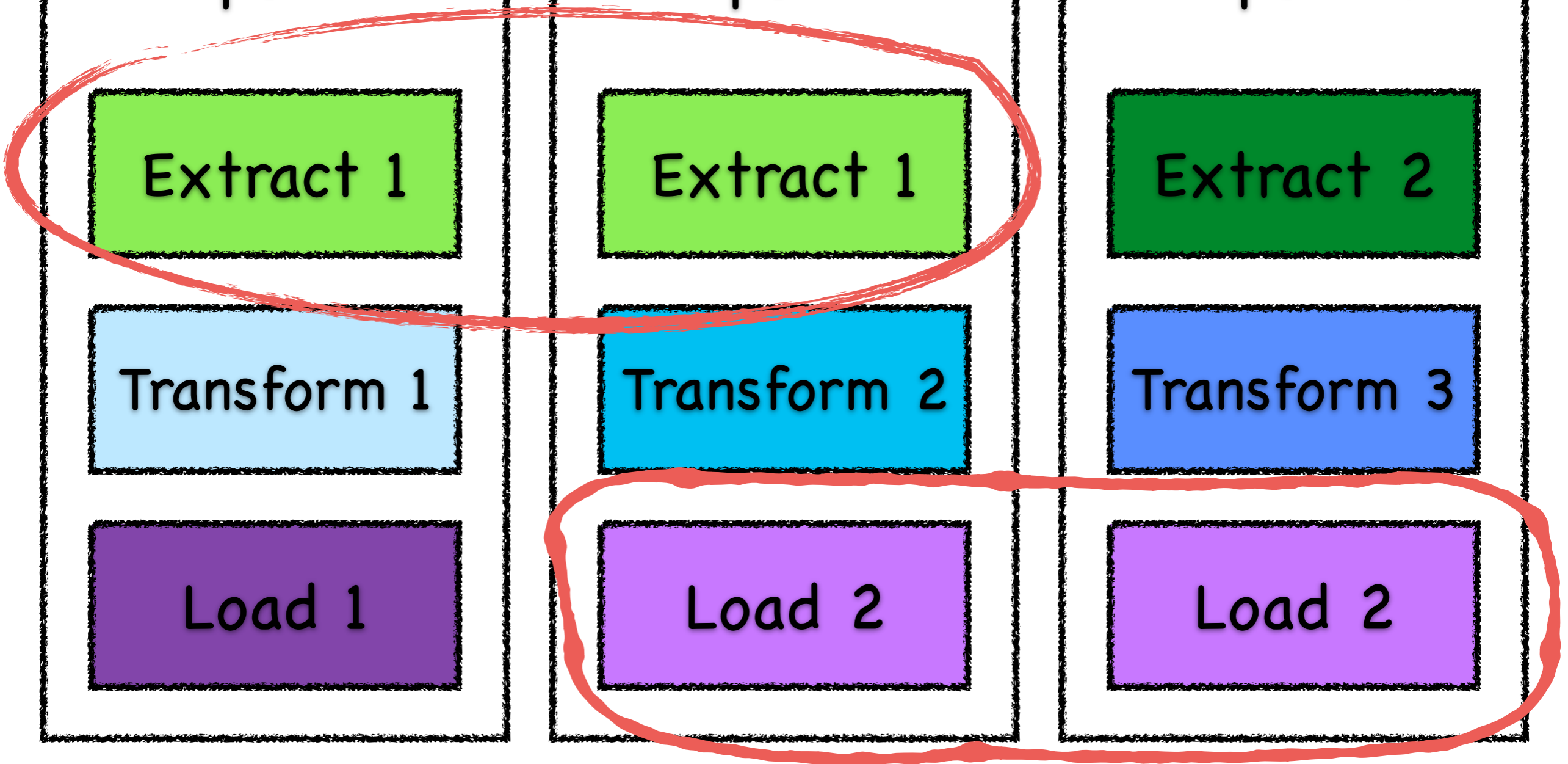
Load 2

Report C

Extract 2

Transform 3

Load 2



Strategy


What Is It?

Polymorphic Algorithm

**Polymorphic
what?**



Change The Algorithm At Runtime

A black dog, possibly a Labrador Retriever, is sitting on a wooden deck. The dog is looking slightly to the left of the camera with its mouth open, showing its pink tongue. It is wearing a red collar with a silver ring. A white speech bubble with a black border is positioned to the left of the dog's head, containing the text "blah blah algorithm?". The background consists of wooden planks.

**blah blah
algorithm?**

Extract Algorithm

Step 1

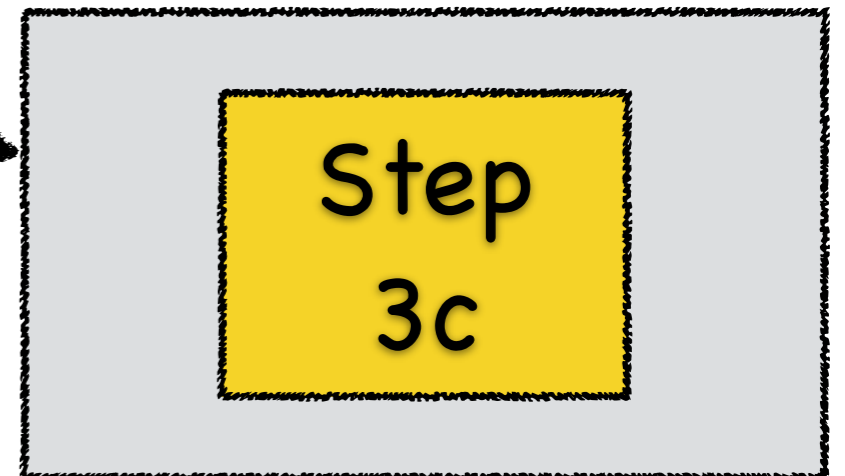
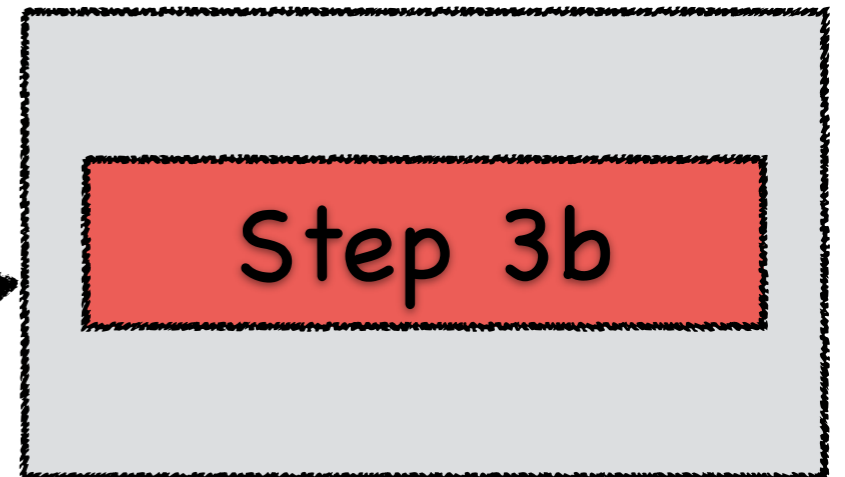
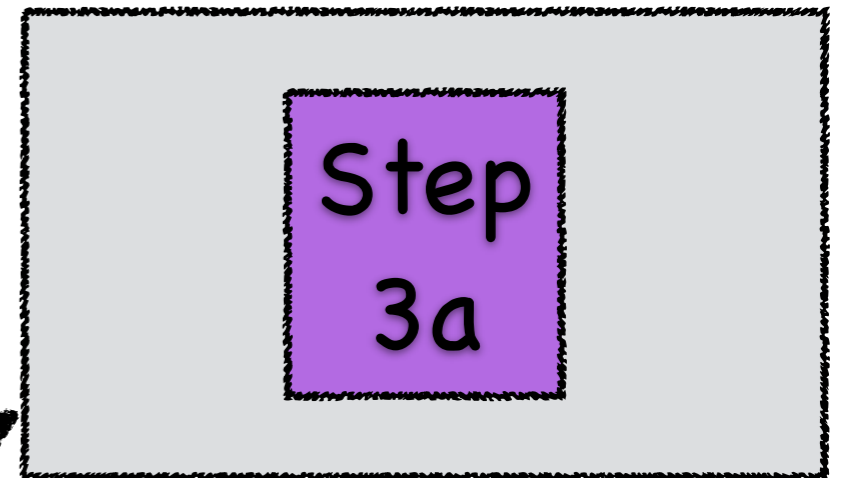
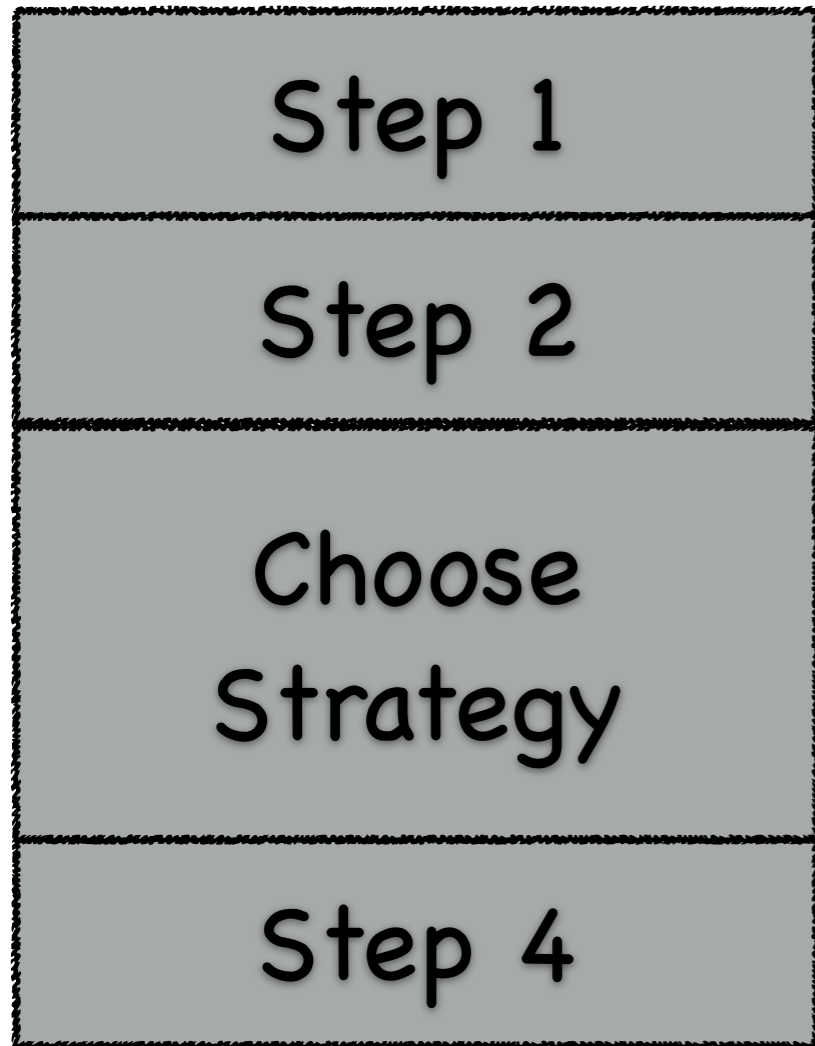
Step 2

Step
3a

Step 3b

Step 3c

Step 4



Examples

100%
Real

Storage Strategy

Stock Market

STOCK	BID	OFFER	LAST	VOL
ALUMINA	5.82	5.83	5.83	6M
AMCOR	8.96	8.97	8.96	3M
AMP	7.10	7.11	7.10	15M
ANZ BANK	18.23	18.23	18.22	3M
BHP BLT	11.41	11.43	11.42	7M
COLES MYER	28.31	28.36	28.33	2HT
CWLTH BANK	4.46	4.47	4.47	2M
FOSTERS	30.97	31.00	30.99	6M
NAT BANK	12.44	12.45	12.45	4M
NEWS CORP	35.08	35.12	35.12	5M
RIO TINTO	21.37	21.40	21.40	7HT
ST. GEORGE	4.88	4.89	4.89	16M
TELSTRA	3.48	3.49	3.49	4M
WES. TRUST	14.06	14.07	14.07	7HT
WESFARMER	16.3	16.3	16.3	5M
WESTFIELD	13.75	13.75	13.75	9200
WESTFLD	11.09	11.09	11.09	5M
WESTPAC	11.09	11.09	11.09	5M
WOODSIDE	13.75	13.75	13.75	9200
WOOLWORTHS	11.09	11.09	11.09	5M

STOCK	BID	OFFER	LAST	VOL
SSH MEDIC	0.105	0.11	0.11	44T
ST FRANCIS	0.021	0.022	0.022	5HT
ST SYNERGY	0.11	0.165	0.165	0
ST. BARBARA	0.085	0.086	0.085	4HT
STADIUM	0.205	0.22	0.205	4T
STARGAMES	1.21	1.24	1.24	4HT
STARPHARMA	0.65	0.66	0.65	1HT
STARTRACK	0.027	0.028	0.028	2HT
STEAMSHIP	1.90	2.10	1.94	0
STERICORP	0.17	0.175	0.175	0
STHN EQUIT	-	-	-	-
STHN PAC.	0.26	0.265	0.265	0
STHN STAR	0.69	0.69	0.69	0
STHN TR	9.75	9.75	9.75	0
STHN TR	0.039	0.039	0.039	0
STHN TR	0.89	0.90	0.90	1M
STHN TR	0.265	0.35	0.265	0
STRAITRES	0.82	0.84	0.83	4HT
STRARCH	-	-	0.10	0

STOCK	BID	OFFER	LAST	VOL
STRAT MINS	0.235	0.24	0.24	2HT
STRAT POOL	0.245	0.25	0.25	0
STRATA	0.022	0.023	0.023	4HT
STRATATEL	0.069	0.08	0.068	0
STRATHFLD	0.13	0.135	0.135	2HT
STRIKER	0.065	0.066	0.066	3M
STRUCTURAL	0.485	0.50	0.50	0
STUART	0.72	0.73	0.73	HT
STUART	3.35	3.38	3.38	HT
STUART	0.057	0.058	0.058	5M
STUART	0.2	0.2	0.2	HT
STUART	0.09	0.09	0.09	HT
STUART	12.4	12.4	12.4	HT
STUART	0.065	0.065	0.065	HT
STUART	0.155	0.155	0.155	HT
STUART	1.18	1.18	1.18	HT
SUNRAY.TV.	11.11	15.00	11.00	0
SUNSHINE G	0.275	0.285	0.275	14T
SUNTECH	0.031	0.033	0.031	2HT
SUNVEST	0.65	-	0.65	0

STOCK	BID	OFFER	LAST	VOL
ASX ALL ORDINARIES	16	16	16	3283.0
ASX 200	16	16	16	16
ASX 300	16	16	16	16
ASX MIDCAP 50	16	16	16	16
ASX SMALL ORDS	16	16	16	16
ASX ENERGY	16	16	16	16
ASX MATERIALS	16	16	16	16
ASX INDUSTRIALS	16	16	16	16
ASX CONS DIS	16	16	16	16
ASX CONS ST	16	16	16	16
ASX FIN	16	16	16	16
ASX UTILITIES	16	16	16	16

28400	WESFARMER 28.13	16500	WESTFIELD 14.07	22100	WOOLWORTHS 11.09	17200	ALUMINA 5.83	48600	BHP BLT 11.42	62200	RIO TINTO 35.12	5500	WOODSIDE 13.75	9200
5:24pm	MELBOURNE	3283.0	+16.8	4:2pm	TOKYO	1x	8:24am	LONDON	4311.0	-2.9	3:24am	NEW YORK	9674.68	-5.33

A close-up photograph of a calculator keypad. The keys are white with black text. The central focus is on the 'TAXI' key, which is slightly larger and more prominent than the others. To its left is a key labeled 'C/A C', and above it is a key labeled 'M/EX'. Below the 'TAXI' key is a key labeled 'RATE'. The background is a soft, out-of-focus light blue. The text 'Sales Tax' is overlaid in a large, bold, black font across the center of the image.

Sales Tax

<http://www.flickr.com/photos/phillip/345829246/>

Recognizing It

```
def handle(a, b, c)
  # step 1
  normalize_inputs a, b, c

  # step 2
  reticulate_splines

  # step 3
  if a == 'x'
    results = x_request a, b
  elsif a == 'y'
    results y_request a
  elsif a == 'z' and b == '123'
    results z_request a, b, c
  end

  # step 4
  present results
end
```

```
def handle(a, b, c)
  # step 1
  normalize_inputs a, b, c

  # step 2
  reticulate_splines

  # step 3
  if a == 'x'
    results = x_request a, b
  elsif a == 'y'
    results y_request a
  elsif a == 'z' and b == '123'
    results z_request a, b, c
  end

  # step 4
  present results
end
```

```
def initialize
  @strategies = {}
  @strategies['x'] = XStrategy.new
  @strategies['y'] = YStrategy.new
  @strategies['z'] = ZStrategy.new
end
```

```
def handle a, b, c
  # step 1
  normalize_inputs a, b, c

  # step 2
  reticulate_splines

  # step 3
  @strategies[a].run a, b, c

  # step 4
  present_results
end
```



```
def initialize
  @strategies = {}
  @strategies['x'] = XStrategy.new
  @strategies['y'] = YStrategy.new
  @strategies['z'] = ZStrategy.new
end
```

```
def handle a, b, c
  # step 1
  normalize_inputs a, b, c

  # step 2
  reticulate_splines

  # step 3
  @strategies[a].run a, b, c

  # step 4
  present_results
end
```

```
def initialize()  
  @strategies = {}  
  @strategies['x'] = XStrategy.new  
  @strategies['y'] = YStrategy.new  
  @strategies['z'] = ZStrategy.new  
  @strategies['aa'] = AAStrategy.new  
end
```

```
def handle(a, b, c)  
  # step 1  
  normalize_inputs a, b, c  
  
  # step 2  
  reticulate_splines  
  
  # step 3  
  @strategies[a].run(a, b, c)  
  
  # step 4  
  present_results  
end
```

```
def add_strategy(key, strategy)
  @strategies[key] = strategy
end
```

```
add_strategy 'aa', AAStrategy.new
```

A large, light-skinned thumbs-up emoji is centered on the page. It is positioned behind the text, with its thumb pointing upwards and its fingers curled. The emoji has a soft shadow and a slight gradient, giving it a three-dimensional appearance.

Share Common Code

Isolate Differences

Easily Add New Strategies

Force Common Interface

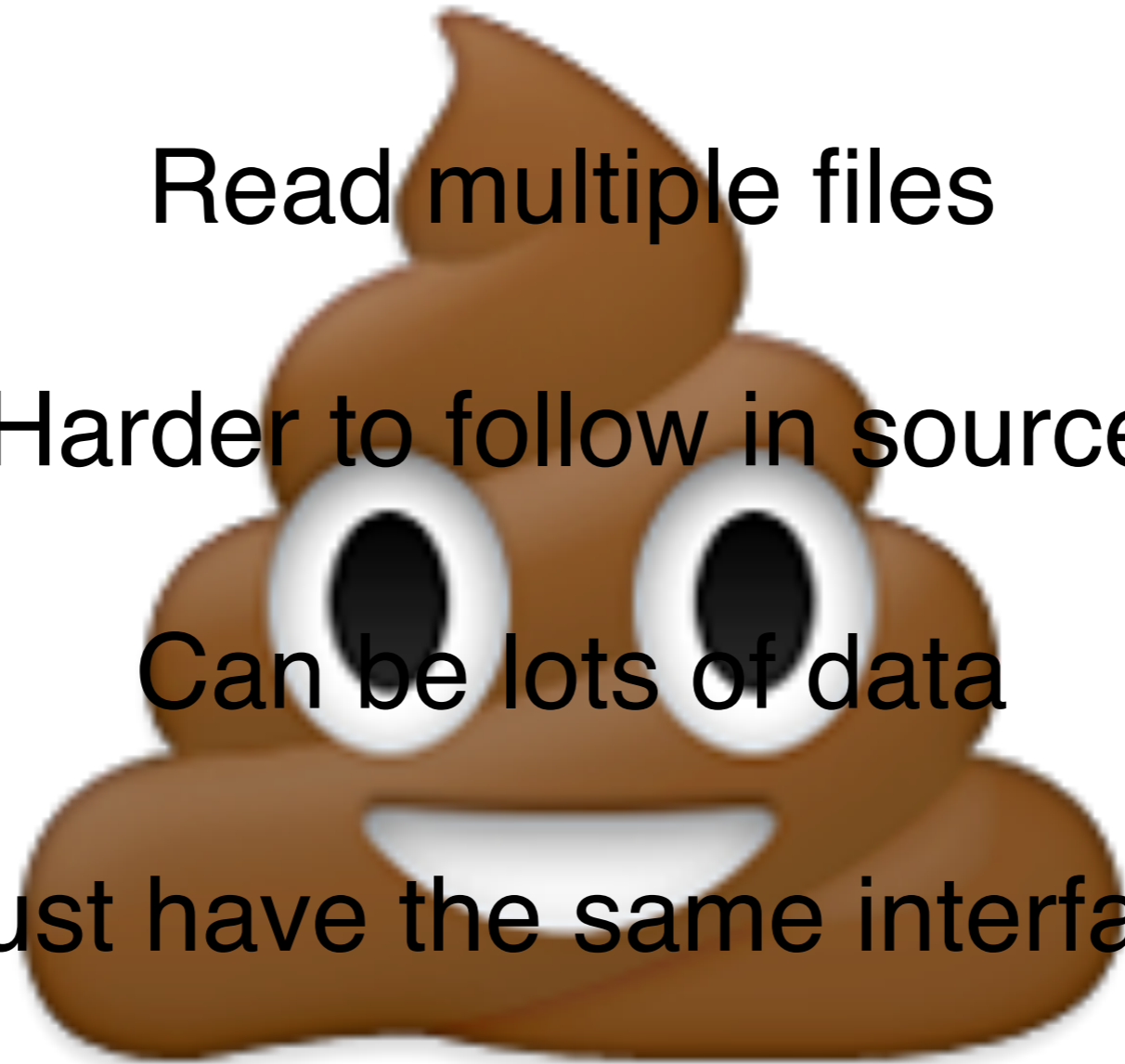
Avoid Inheritance

Read multiple files

Harder to follow in source

Can be lots of data

Must have the same interface



Composite

What Is It?

Common Interface For One or Many

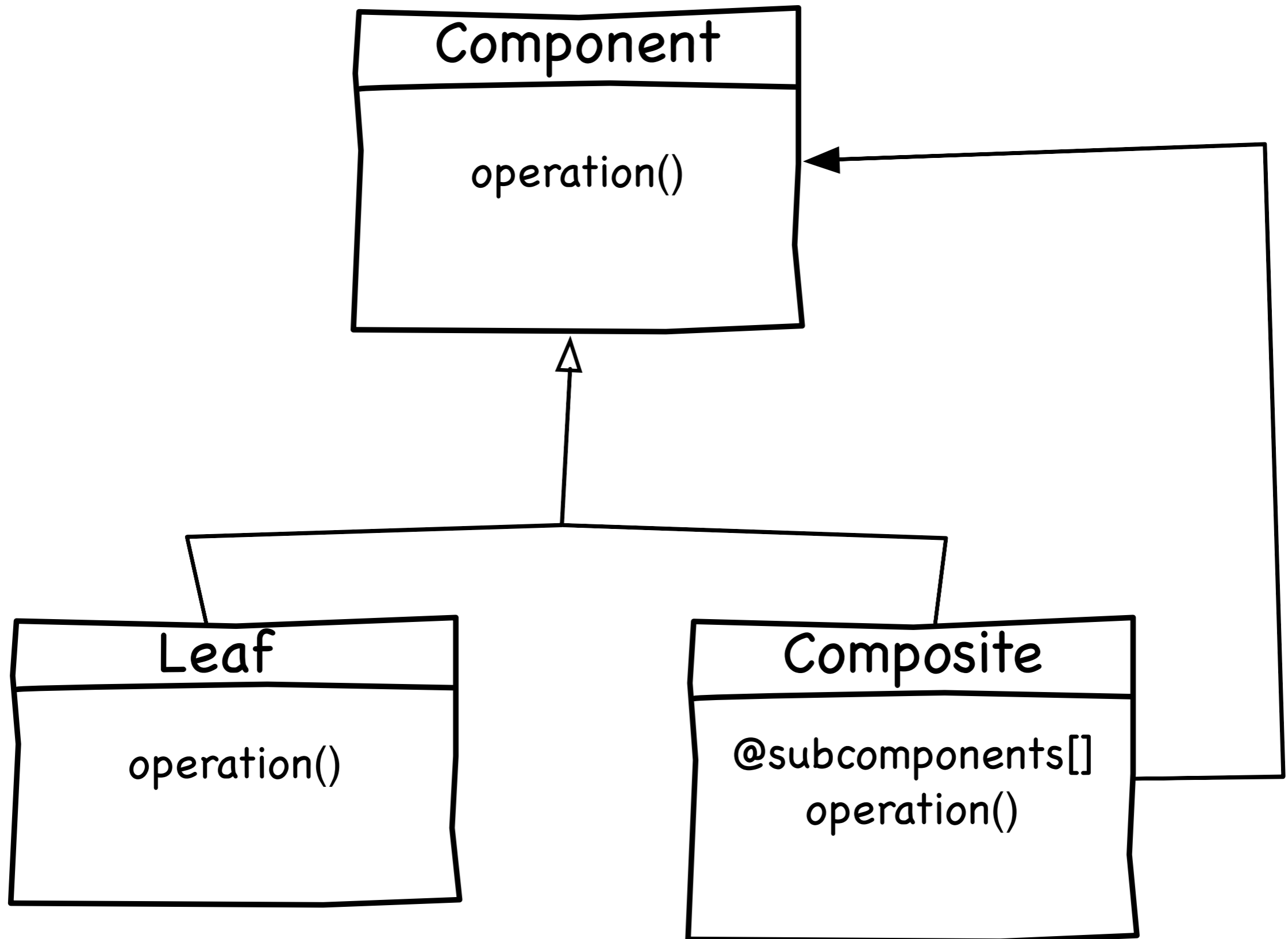


**Common
Interface?**

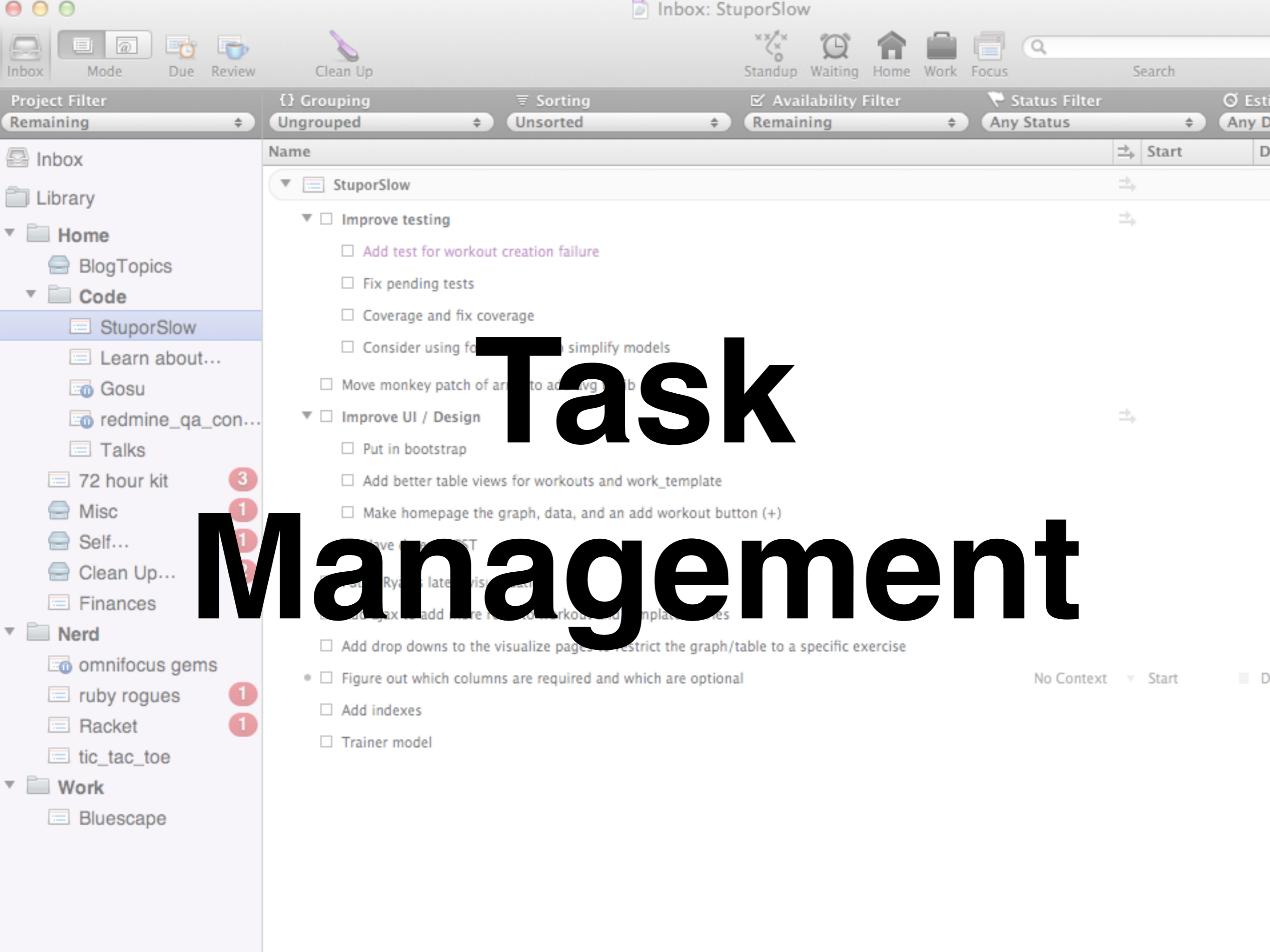
<https://www.flickr.com/photos/joi/2569341936>

Let Consumers
Ignore Quantity

Vocabulary



Examples



Task Management

Inbox Mode Due Review Clean Up

Standup Waiting Home Work Focus Search

Project Filter Remaining Grouping Ungrouped Sorting Unsorted Availability Filter Remaining Status Filter Any Status

- Inbox
- Library
- Home
 - BlogTopics
 - Code
 - StuporSlow
 - Learn about...
 - Gosu
 - redmine_qa_con...
 - Talks
 - 72 hour kit (3)
 - Misc (1)
 - Self... (1)
 - Clean Up... (2)
 - Finances
- Nerd
 - omnifocus gems
 - ruby rogues (1)
 - Racket (1)
 - tic_tac_toe
- Work
 - Bluescape

- | Name | Start | D |
|--|------------|-------|
| StuporSlow | | |
| Improve testing | | |
| Add test for workout creation failure | | |
| Fix pending tests | | |
| Coverage and fix coverage | | |
| Consider using fo... simplify models | | |
| Move monkey patch of ar... to ad... avg... | | |
| Improve UI / Design | | |
| Put in bootstrap | | |
| Add better table views for workouts and work_template | | |
| Make homepage the graph, data, and an add workout button (+) | | |
| Add drop downs to the visualize pages... restrict the graph/table to a specific exercise | | |
| Figure out which columns are required and which are optional | No Context | Start |
| Add indexes | | |
| Trainer model | | |

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">

<html>

<head>

<meta name="TITLE" content="..."/>

<meta name="KEYWORDS" content="..."/>

<meta name="DESCRIPTION" content="..."/>

<link rel="stylesheet" type="text/css" href="..."/>

<script language="javascript" type="text/javascript">

HTML

</script>

</head>

<body bgcolor="#ffffff" width="100%">

90%
Real

Test Scores

Recognizing It

```
def report_scores x
  case x
  when Student
    x.score
  when Classroom
    x.students.map(&:score).avg
  when School
    x.classes.map(&:students).flatten.map(&:score).avg
  end
end
```

```
def report_scores x
  case x
  when Student
    x.score
  when Classroom
    x.students.map(&:score).avg
  when School
    x.classes.map(&:students).flatten.map(&:score).avg
  end
end
```

```
def report_scores x
  case x
  when Student
    x.score
  when Classroom
    x.students.map(&:score).avg
  when School
    x.classes.map(&:students).flatten.map(&:score).avg
  end
end
```

```
def report_scores x
  case x
  when Student
    x.score
  when Classroom
    x.students.map(&:score).avg
  when School
    x.classes.map(&:students).flatten.map(&:score).avg
  end
end
```

Composite

```
class Composite
  def score
    # Some implementation
  end
end
```



```
class Student  
  attr_accessor :score  
end
```

```
class Classroom  
  attr_accessor :students  
  
  def score  
    students.map(&:score).avg  
  end  
end
```

```
class School
  attr_accessor :classes

  def score
    classes.map(&:score).avg
  end
end
```

```
def report_scores(x)
  x.score
end
```

A large, light-skinned thumbs up emoji is centered on the page. It is a simple, cartoonish representation of a hand with the thumb pointing upwards. The emoji is slightly blurred and has a soft shadow beneath it.

Common Interface

Arbitrary Depth

Insert In the Middle

No Messy ifs

Harder to follow in source

Must remember hierarchy



Decorator

What Is It?

**Extend an Instance of
an Object**

**an instance of
food?**



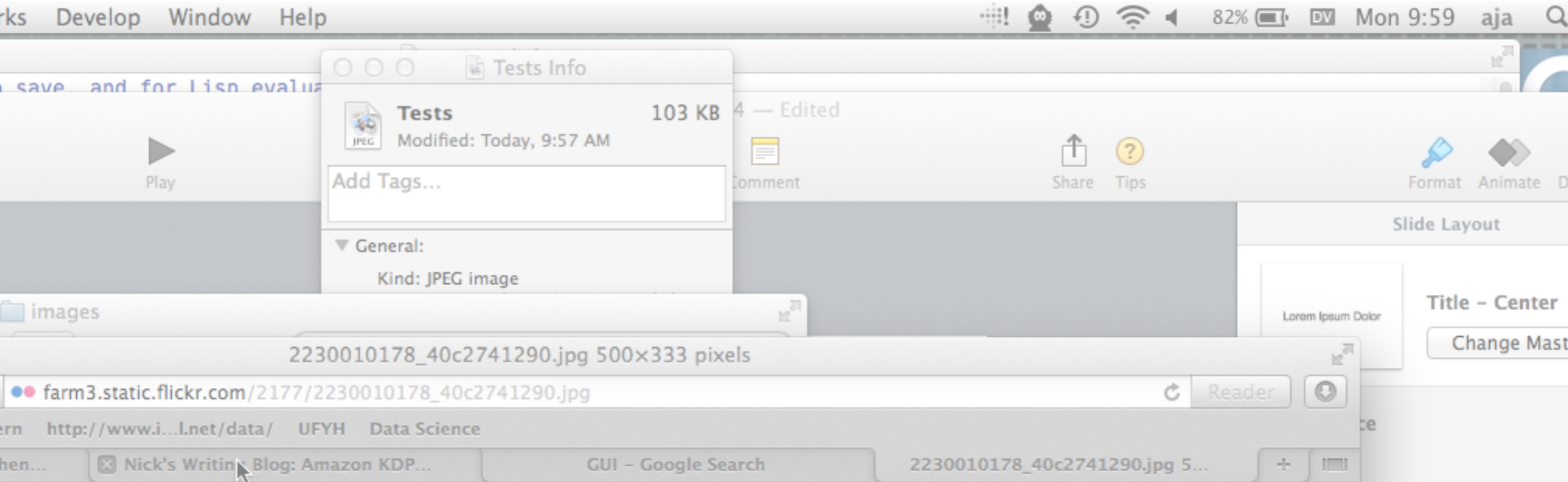
<http://laura-c-f.deviantart.com/art/Confused-Dog-286725830>

Wrapper Adding Functionality



My Object

Examples



GUI Elements



BUY ONE

50%

2nd item

excludes green tagged incred
value every day product

Calculating Discounts



RPG Characters



Recognizing It

```
class Character
  attr_accessor :health, :damage, :xp

  def alive?
    health > 0
  end

  def attack enemy
    enemy.health -= self.damage
  end
end
```

POW!!!

```
require 'forwardable'

class DragonMightDecorator
  extend Forwardable

  def_delegators :@character, :health, :xp

  def initialize(character)
    @character = character
  end

  def damage
    @character.damage * 1.5
  end
end
```

```
require 'forwardable'

class DragonMightDecorator
  extend Forwardable

  def_delegators :@character, :health, :xp

  def initialize(character)
    @character = character
  end

  def damage
    @character.damage * 1.5
  end
end
```

```
require 'forwardable'

class DragonMightDecorator
  extend Forwardable

  def_delegators :@character, :health, :xp

  def initialize(character)
    @character = character
  end

  def damage
    @character.damage * 1.5
  end
end
```

```
require 'forwardable'

class DragonMightDecorator
  extend Forwardable

  def_delegators :@character, :health, :xp

  def initialize(character)
    @character = character
  end

  def damage
    @character.damage * 1.5
  end
end
```

```
> c = Character.new
> c.health = 100; c.damage = 10;

> d = DragonMightDecorator.new(c)
> d.damage
15.0

> d.health
100
```


Change Behavior Dynamically

Customize an Instance

Support Nesting



Offload Pain To Consumers

Multiple Files

Many Decorators -> Unpredictable



Command

What Is It?

Objectified Closure



**What Is
Closure?**

Object Containing Algorithm & Context

Examples



Delayed Job

Drawing Applications

100%
Real



Active Record Migrations

Recognizing It

Undo

```
class CreateRecipes < ActiveRecord::Migration
  def self.up
    create_table :recipes do |t|
      t.column :name, :string
      t.timestamps
    end
  end

  def self.down
    drop_table :recipes
  end
end
```

```
class CreateRecipes < ActiveRecord::Migration
  def self.up
    create_table :recipes do |t|
      t.column :name, :string
      t.timestamps
    end
  end

  def self.down
    drop_table :recipes
  end
end
```

Delay

Delegation



Undo & Redo

Offload / Queue / Distribute / Delegate

Simple To Understand

Encourages Good Design



Overused

YAGNI

So What?

You Are Using
Patterns

Common Vocabulary

Cheat Sheet

Several Objects That
Are Mostly The
Same?

Template

Process The Same
Except *Step 3*?

Strategy

Hierarchical Objects?

Composite

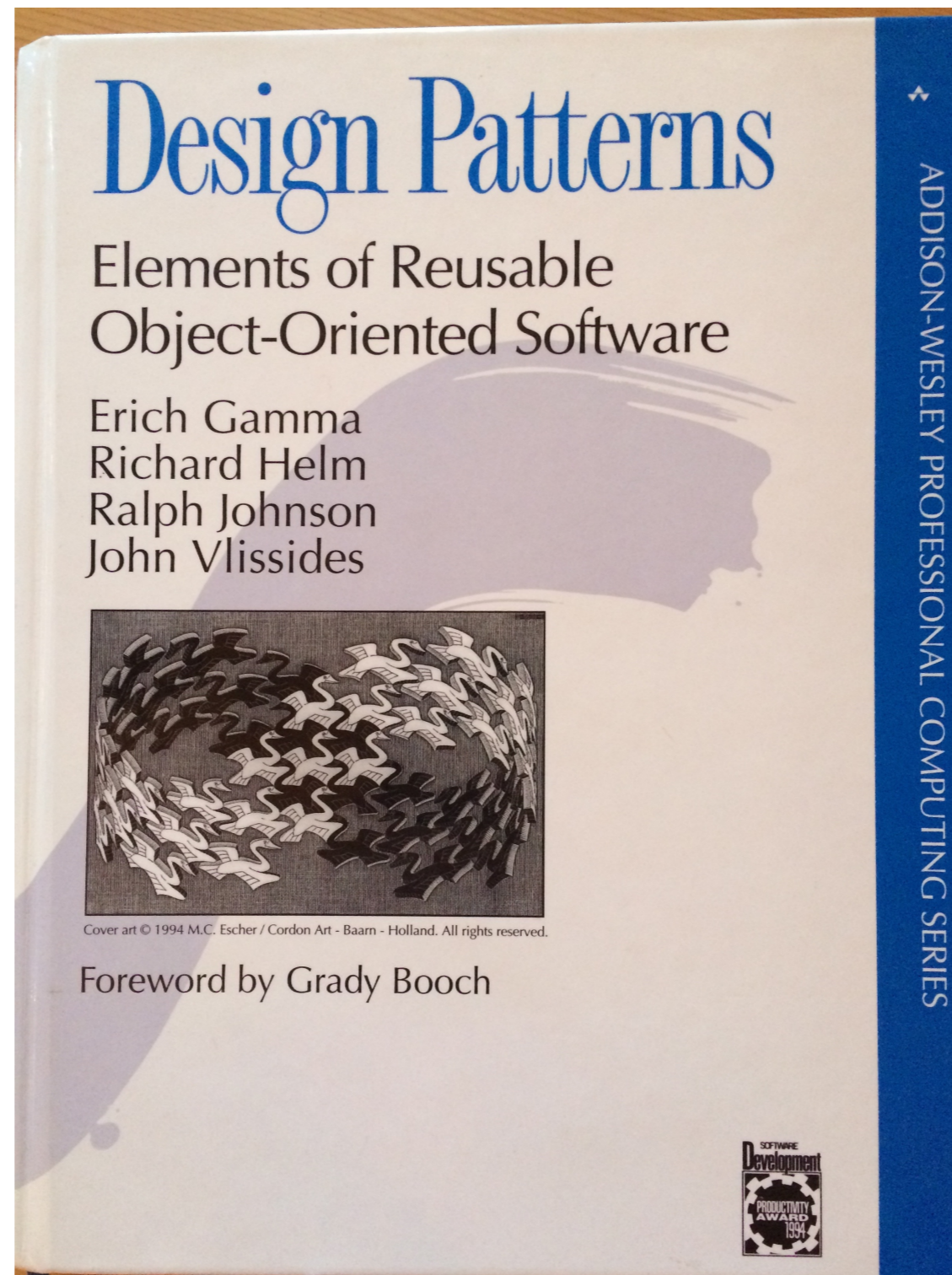
Sometimes Need
Enhancements?

Decorator

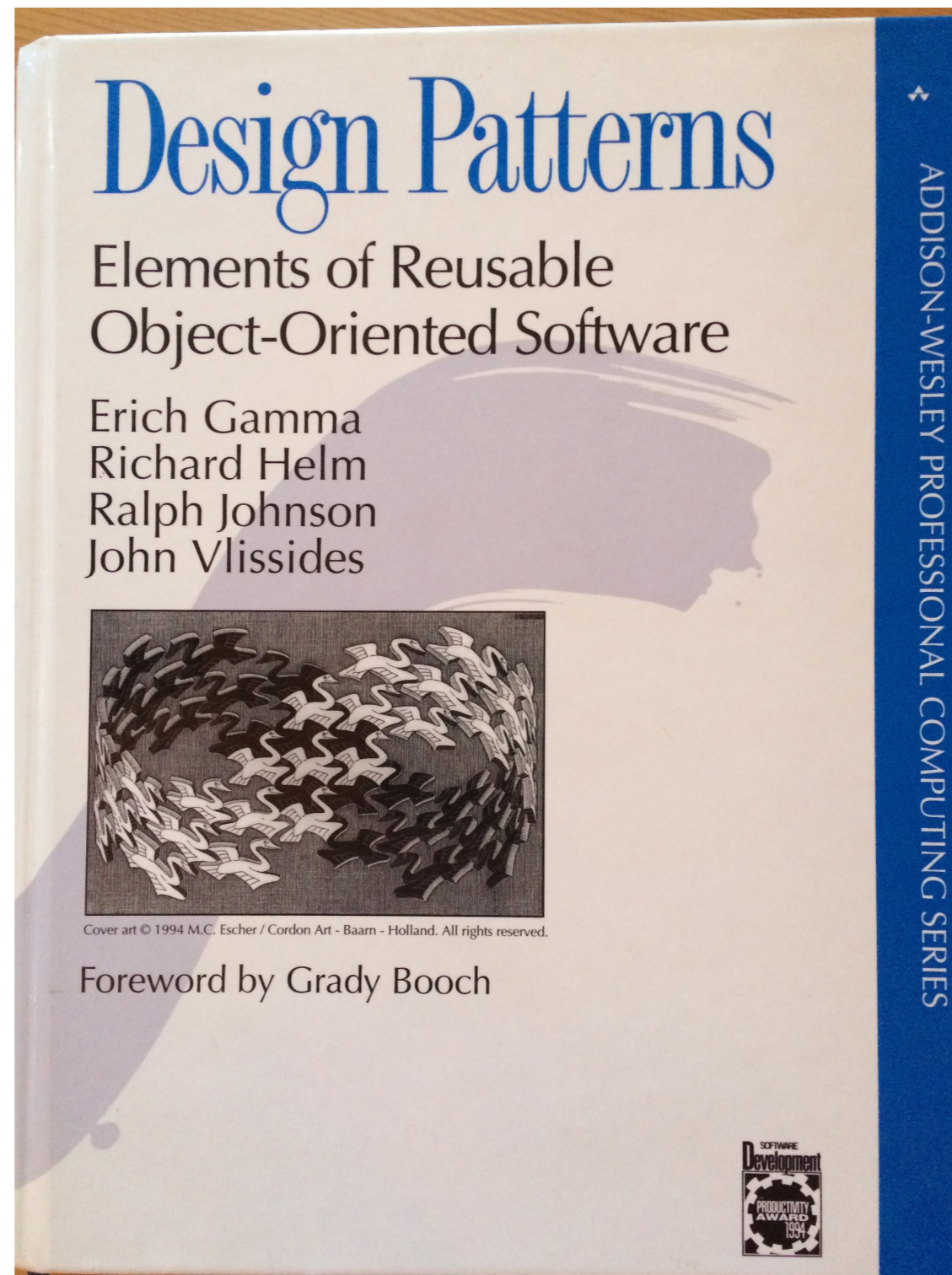
Undo or Delay or
Delegate?

Command

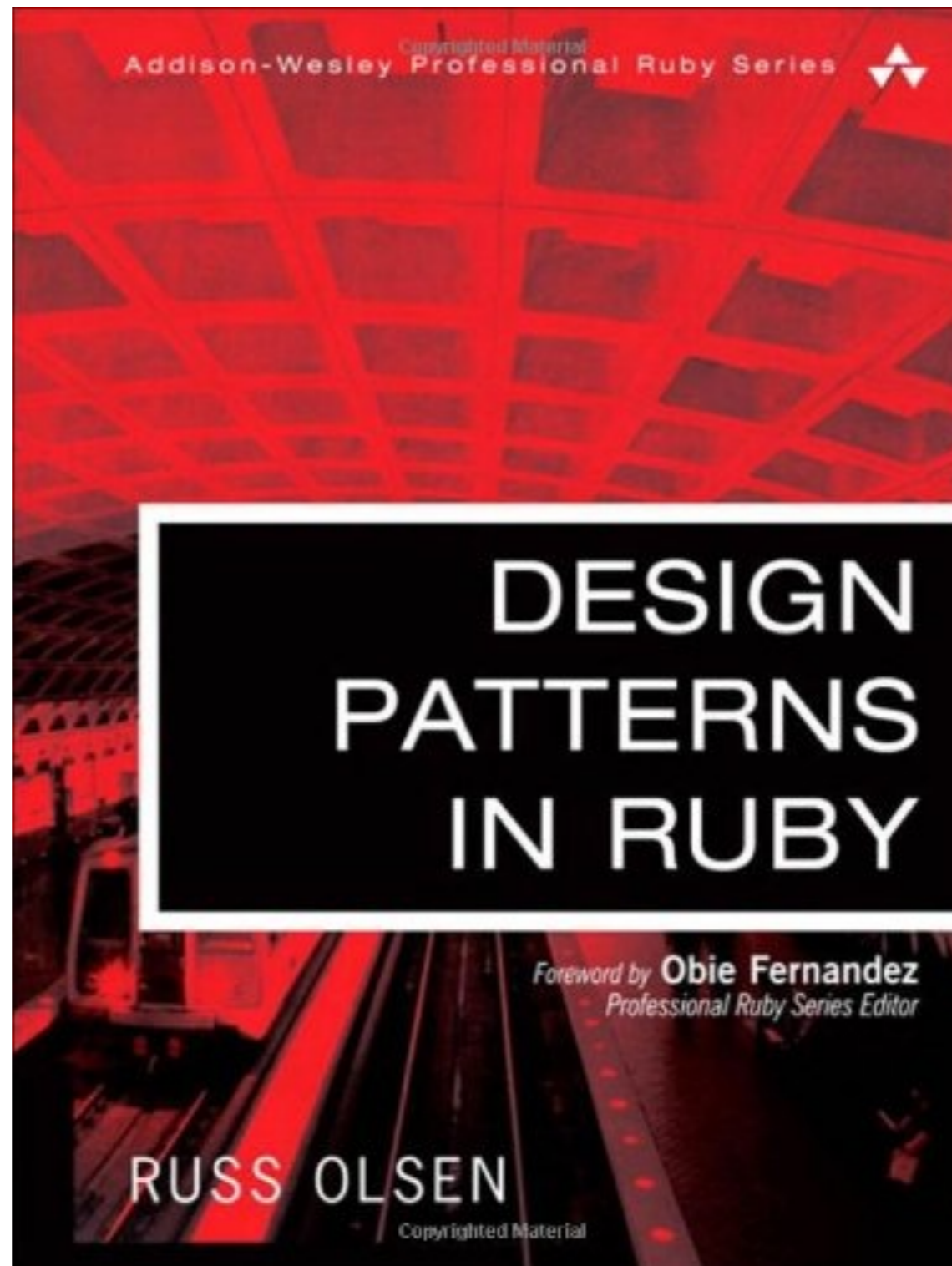
[Learn More](#)



Design Patterns



Gang Of Four




Design Patterns In Ruby

Pattern Index

c2.com/cgi-bin/wiki?PatternIndex

Last-Minute ...ree pattern http://www.i...l.net/data/ UFYH Data Science



Pattern Index

The book [Pattern Languages Of Program Design](http://hillside.net/patterns/books/Details/070.htm) <http://hillside.net/patterns/books/Details/070.htm> is an eclectic collection of patterns. Many of them are accessible on line.

See [CategoryPattern](#) index.

The [GangOfFour](#) wrote up 23 [DesignPatterns](#). These patterns elaborate on their work:

- [ExtensionObjects](#)
- [HiddenManagers?](#)

They are also interested in [UsesOfGofPatterns](#).

Some pattern languages that are useful for large business applications.

- [CrossingChasms](#)
- [RelationalDatabaseAccessLayer](#)
- [ErrorHandling](#)
- [DecouplingObjects](#) ([DecouplingOfObjectOrientedSystems](#))
- [FrameworkConstruction](#)

Other sets of patterns (often pattern languages).

- [UserInterfacePatterns](#)
- [MessagingPattern](#)
- [SystemOfNames](#)
- [TransactionsAndAccounts](#)
- [DebuggingPatternLanguage](#)
- [TestingPatterns](#)
- [OrganizationalPatterns](#)
- [FunctionalPatternSystemForObjectOrientedDesign](#)
- A pattern language for [ObjectBasedProgramming](#) in a procedural language.
- [PatternsForEffectiveMeetings](#)
- [ComponentDesignPatterns](#)

Original Wiki

Thanks