Design Patterns

Index

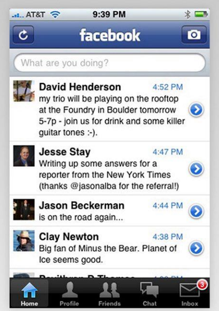
- Design Pattern introduction
- Decorator Pattern
- Strategy Pattern
- Observer Pattern

What are Design Pattern?

In software engineering, a design pattern is a general repeatable solution to a commonly occurring problem in software design. A design pattern is not a finished design that can be transformed directly into code. It is a description or template for how to solve a problem that can be used in many different situations.

Why should we use it?

Design Patterns provide easy to recognize and use OOP solutions to common problems. They're inherently easy to maintain, because many people are familiar with them.


AT&T 9:39 PM

facebook

What are you doing?

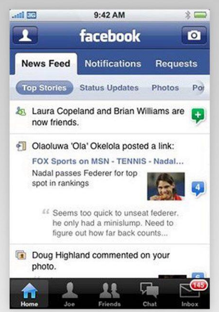
David Henderson 4:52 PM
my trio will be playing on the rooftop at the Foundry in Boulder tomorrow 5-7p - join us for drink and some killer guitar tones :-).

Jesse Stay 4:47 PM
Writing up some answers for a reporter from the New York Times (thanks @jasonalba for the referral!)

Jason Beckerman 4:44 PM
is on the road again...

Clay Newton 4:38 PM
Big fan of Minus the Bear. Planet of Ice seems good.

2008
July


9:42 AM

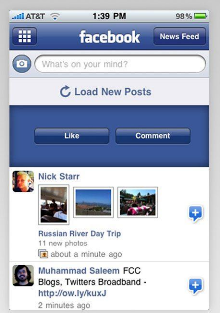
facebook

Laura Copeland and Brian Williams are now friends.

Olaoluwa 'Ola' Okelola posted a link:
FOX Sports on MSN - TENNIS - Nadal...
Nadal passes Federer for top spot in rankings
"Seems too quick to unseat federer. he only had a minislump. Need to figure out how far back counts...

Doug Highland commented on your photo.

2008
September


AT&T 1:39 PM 98 %

facebook

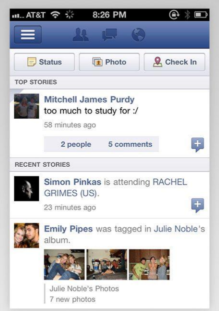
What's on your mind?

Load New Posts

Nick Starr
Russian River Day Trip
11 new photos
about a minute ago

Muhammad Saleem FCC Blogs, Twitters Broadband - http://ow.ly/kuxJ
2 minutes ago

2009
August


AT&T 8:26 PM

TOP STORIES

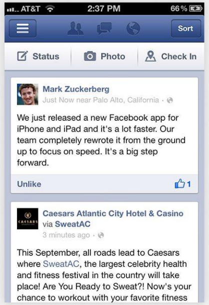
Mitchell James Purdy
too much to study for :/
58 minutes ago
2 people 5 comments

RECENT STORIES

Simon Pinkas is attending RACHEL GRIMES (US).
23 minutes ago

Emily Pipes was tagged in Julie Noble's album.
Julie Noble's Photos
7 new photos

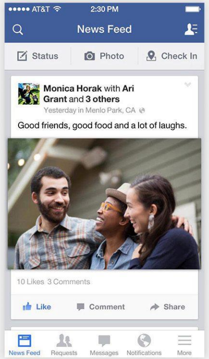
2011
October


AT&T 2:37 PM 66 %

Mark Zuckerberg
Just Now near Palo Alto, California
We just released a new Facebook app for iPhone and iPad and it's a lot faster. Our team completely rewrote it from the ground up to focus on speed. It's a big step forward.
Unlike 1

Caesars Atlantic City Hotel & Casino via SweatAC
3 minutes ago
This September, all roads lead to Caesars where SweatAC, the largest celebrity health and fitness festival in the country will take place! Are You Ready to Sweat?! Now's your chance to workout with your favorite fitness

2012
August

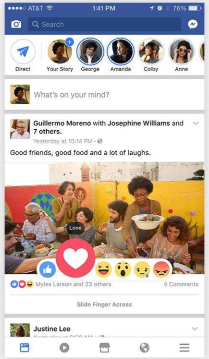

AT&T 2:30 PM

News Feed

Monica Horak with **Ari Grant** and **3 others**
Yesterday in Menlo Park, CA
Good friends, good food and a lot of laughs.

10 Likes 3 Comments

2013
September


AT&T 1:41 PM 76%

What's on your mind?

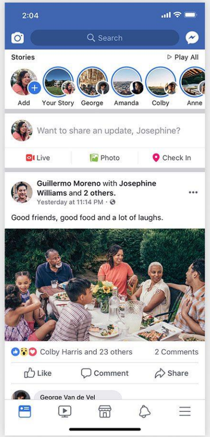
Guillermo Moreno with **Josephine Williams** and **7 others.**
Yesterday at 10:14 PM
Good friends, good food and a lot of laughs.

Myles Larson and 23 others 4 Comments

Slide Finger Across

Justine Lee

2017
March


2:04

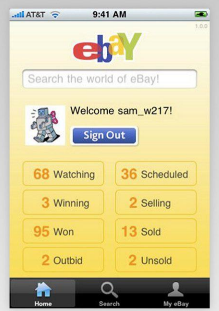
Want to share an update, Josephine?

Guillermo Moreno with **Josephine Williams** and **2 others.**
Yesterday at 11:14 PM
Good friends, good food and a lot of laughs.

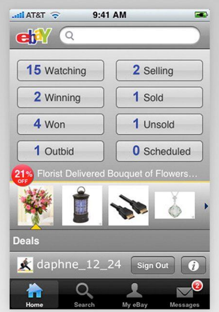
Colby Harris and 23 others 2 Comments

George Van de Vel

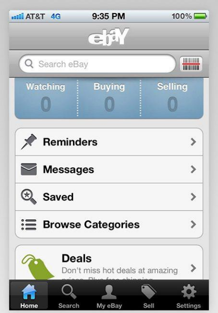
2017
November



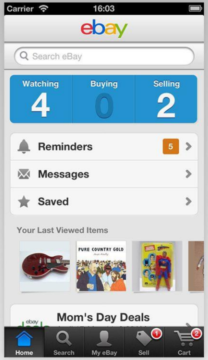
2008
July



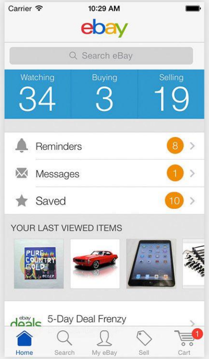
2009
November



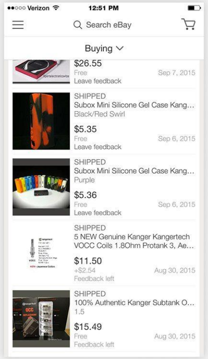
2010
November



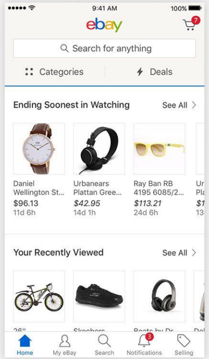
2013
May



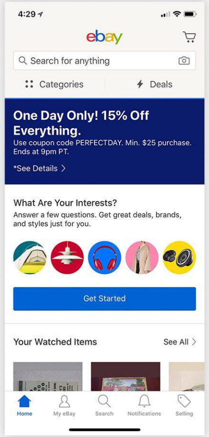
2013
September



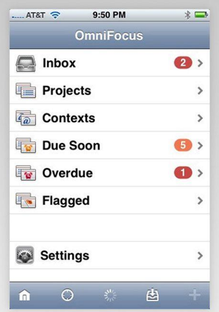
2015
September



2016
May



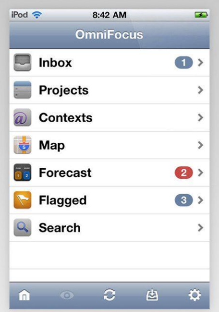
2017
November


..... AT&T 9:50 PM

OmniFocus

Inbox 2
Projects
Contexts
Due Soon 5
Overdue 1
Flagged
Settings

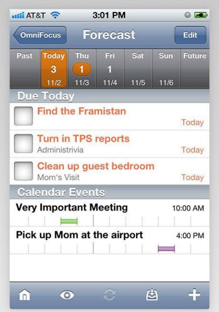
2008
July


iPod 8:42 AM

OmniFocus

Inbox 1
Projects
Contexts
Map
Forecast 2
Flagged 3
Search

2010
June


AT&T 3:01 PM

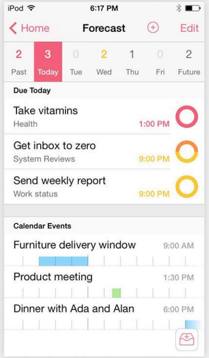
OmniFocus Forecast Edit

Past	Today	Thu	Fri	Sat	Sun	Future
	3	1	1			
	11/2	11/3	11/4	11/5	11/6	

Due Today
Find the Framistan — Today
Turn in TPS reports — Administrivia — Today
Clean up guest bedroom — Mom's Visit — Today

Calendar Events
Very Important Meeting — 10:00 AM
Pick up Mom at the airport — 4:00 PM

2011
June


iPod 6:17 PM

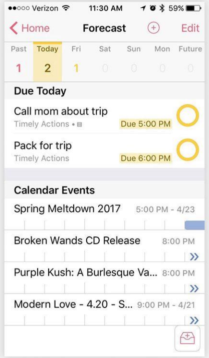
Home Forecast Edit

2	3	0	2	1	0	2
Past	Today	Tue	Wed	Thu	Fri	Future

Due Today
Take vitamins — Health — 1:00 PM
Get inbox to zero — System Reviews — 9:00 PM
Send weekly report — Work status — 9:00 PM

Calendar Events
Furniture delivery window — 9:00 AM
Product meeting — 1:30 PM
Dinner with Ada and Alan — 6:00 PM

2013
September


●●○○○ Verizon 11:30 AM 59%

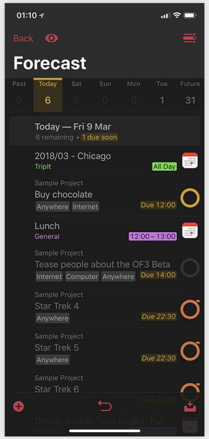
Home Forecast Edit

Past	Today	Fri	Sat	Sun	Mon	Future
1	2	1	0	0	0	0

Due Today
Call mom about trip — Timely Actions — Due 5:00 PM
Pack for trip — Timely Actions — Due 6:00 PM

Calendar Events
Spring Meltdown 2017 — 5:00 PM - 4/23
Broken Wands CD Release — 8:00 PM
Purple Kush: A Burlesque Va... — 8:00 PM
Modern Love - 4.20 - S... — 9:00 PM - 4/21

2015
April


01:10

Back

Forecast

Past	Today	Sat	Sun	Mon	Tue	Future
0	6	0	0	0	1	31

Today — Fri 9 Mar
6 remaining • 1 due soon

2018/03 - Chicago — TripIt — All Day
Sample Project — Buy chocolate — Anywhere Internet — Due 12:00
Lunch — General — 12:00 – 13:00
Sample Project — Tease people about the OF3 Beta — Internet Computer Anywhere — Due 14:00
Sample Project — Star Trek 4 — Anywhere — Due 22:30
Sample Project — Star Trek 5 — Anywhere — Due 22:30
Sample Project — Star Trek 6 — Due 22:30
Dinner, Broken Taco English Pub — General

2018
May

Benefits

Code maintainability: Allows your code to be maintained easier because it is more understandable.

Communication: They help you communicate design goals amongst programmers.

Intention: They show the intent of your code instantly to someone learning the code.

Code re-use: They help you identify common solutions to common problems.

Less code: They allow you to write less code because more of your code can derive common functionality from common base classes.

Tested and sound solutions: Most of the design patterns are tested, proven and

Design Patterns Types

Creational design patterns: These design patterns are all about class instantiation.

Exemples: **Builder**, Abstract Factory and Singleton.

Structural design patterns: These design patterns are all about Class and Object composition.

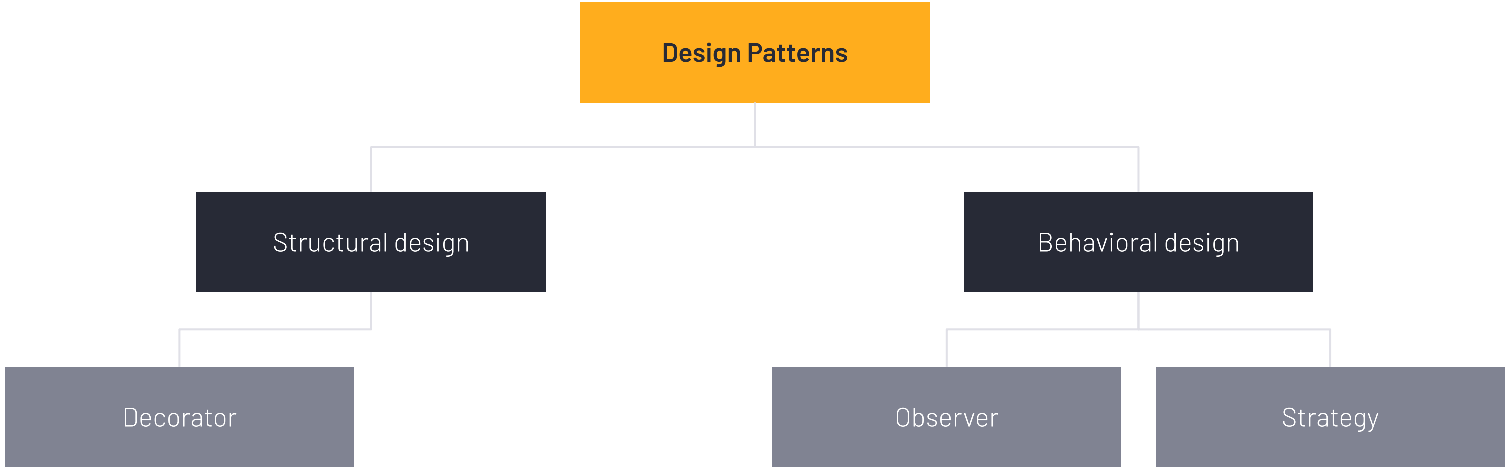
Examples: Adapter, Bridge and **Decorator**.

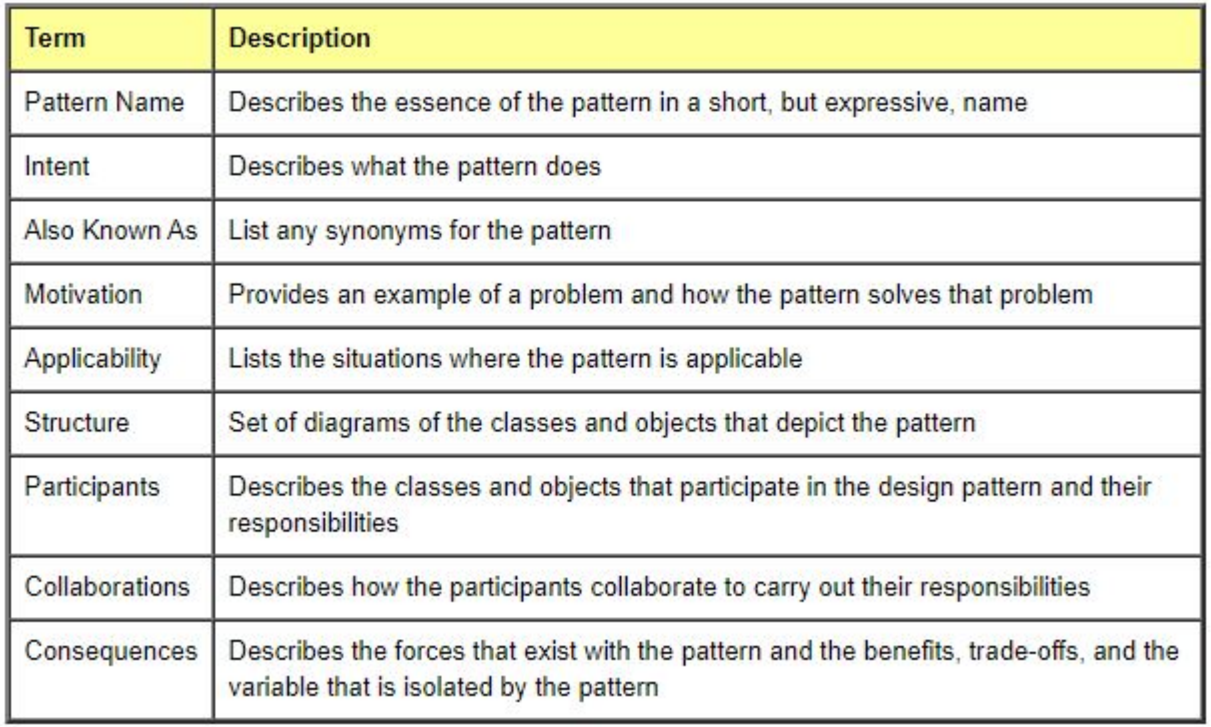
Behavioral design patterns: These design patterns are all about Class's objects communication.

Examples: Mediator, **Observer** and **Strategy**.

See more: https://sourcemaking.com/design_patterns

Designs Patterns

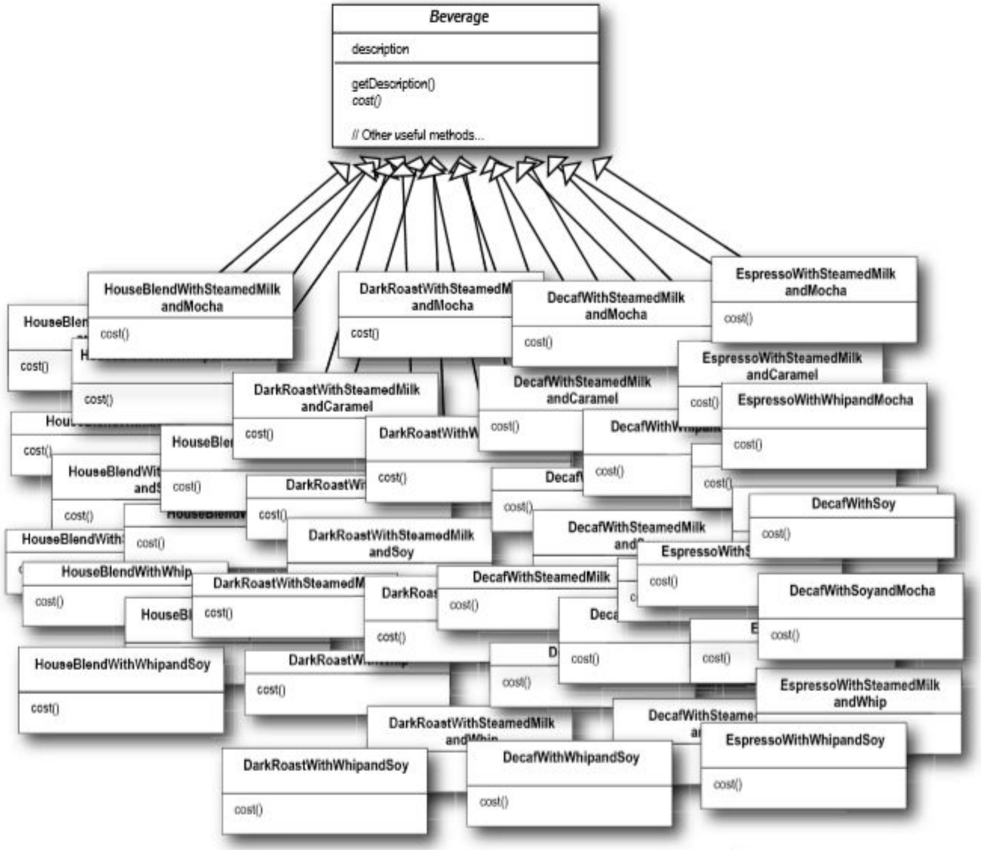




Term	Description
Pattern Name	Describes the essence of the pattern in a short, but expressive, name
Intent	Describes what the pattern does
Also Known As	List any synonyms for the pattern
Motivation	Provides an example of a problem and how the pattern solves that problem
Applicability	Lists the situations where the pattern is applicable
Structure	Set of diagrams of the classes and objects that depict the pattern
Participants	Describes the classes and objects that participate in the design pattern and their responsibilities
Collaborations	Describes how the participants collaborate to carry out their responsibilities
Consequences	Describes the forces that exist with the pattern and the benefits, trade-offs, and the variable that is isolated by the pattern

1. Decorator Pattern

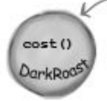
Decoring objects


Beverage
description
getDescription()
cost()
// Other useful methods...
HouseBlendWithSteamedMilk andMocha
DarkRoastWithSteamedMilk andMocha
DecafWithSteamedMilk andMocha
EspressoWithSteamedMilk andMocha
EspressoWithSteamedMilk andCaramel
DarkRoastWithSteamedMilk andCaramel
DecafWithSteamedMilk andCaramel
EspressoWithWhipandMocha
DecafWithSoy
DarkRoastWithSteamedMilk andSoy
DecafWithSteamedMilk
DecafWithSoyandMocha
HouseBlendWithWhip
HouseBlendWithWhipandSoy
EspressoWithSteamedMilk andWhip
DarkRoastWithSteamedMilk andWhip
EspressoWithWhipandSoy
DarkRoastWithWhipandSoy
DecafWithWhipandSoy
cost()

Problem: without decorator

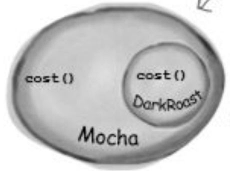
Constructing a drink order with Decorators

❶ **We start with our DarkRoast object.**



Remember that DarkRoast inherits from Beverage and has a cost() method that computes the cost of the drink.

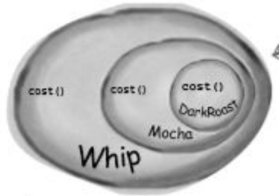
❷ **The customer wants Mocha, so we create a Mocha object and wrap it around the DarkRoast.**



The Mocha object is a decorator. Its type mirrors the object it is decorating, in this case, a Beverage. (By "mirror", we mean it is the same type.)

So, Mocha has a cost() method too, and through polymorphism we can treat any Beverage wrapped in Mocha as a Beverage, too (because Mocha is a subtype of Beverage).

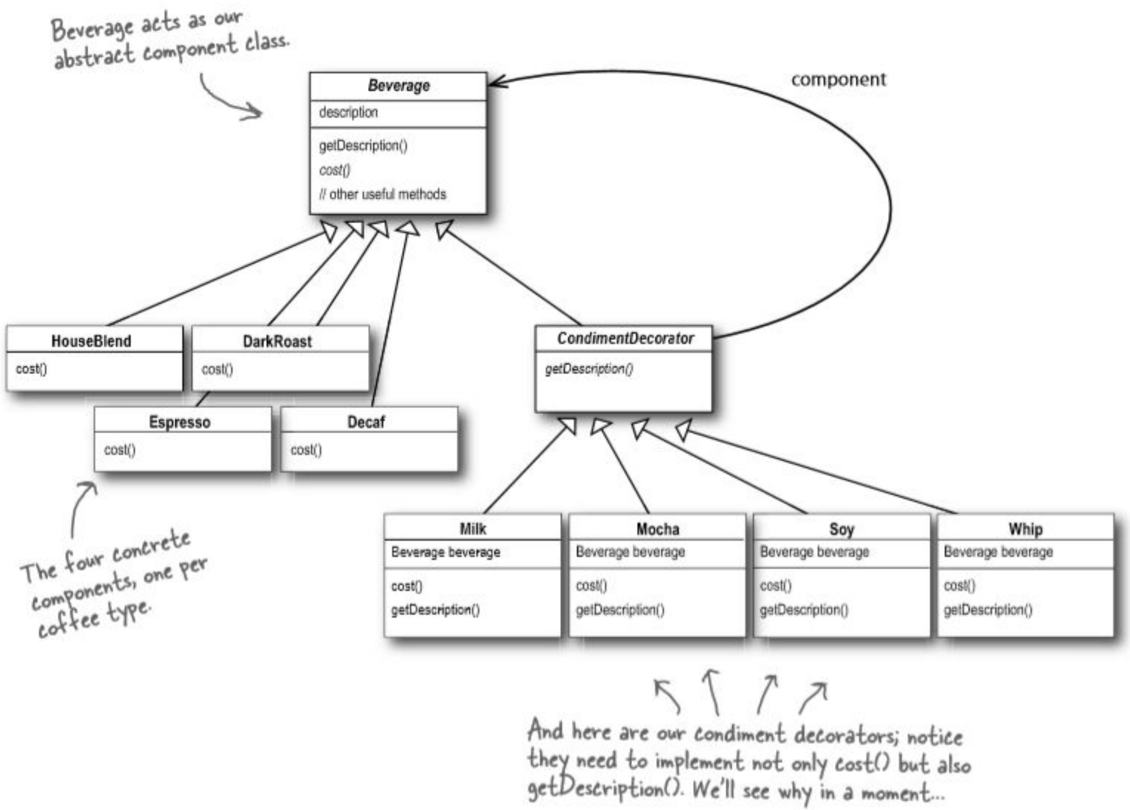
❸ **The customer also wants Whip, so we create a Whip decorator and wrap Mocha with it.**



Whip is a decorator, so it also mirrors DarkRoast's type and includes a cost() method.

So, a DarkRoast wrapped in Mocha and Whip is still a Beverage and we can do anything with it we can do with a DarkRoast, including call its cost() method.

After



With decorator

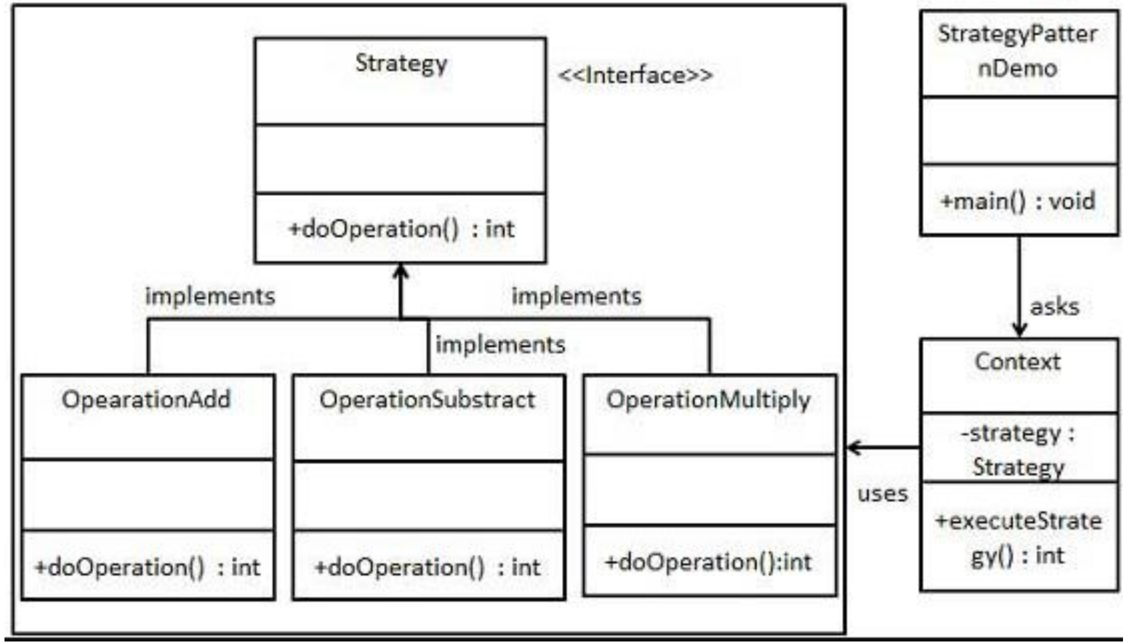
Decorator pattern allows a user to add new functionality to an existing object without altering its structure. This type of design pattern comes under structural pattern as this pattern acts as a wrapper to existing class.

2. Strategy Pattern

Let’s start with the first set of slides

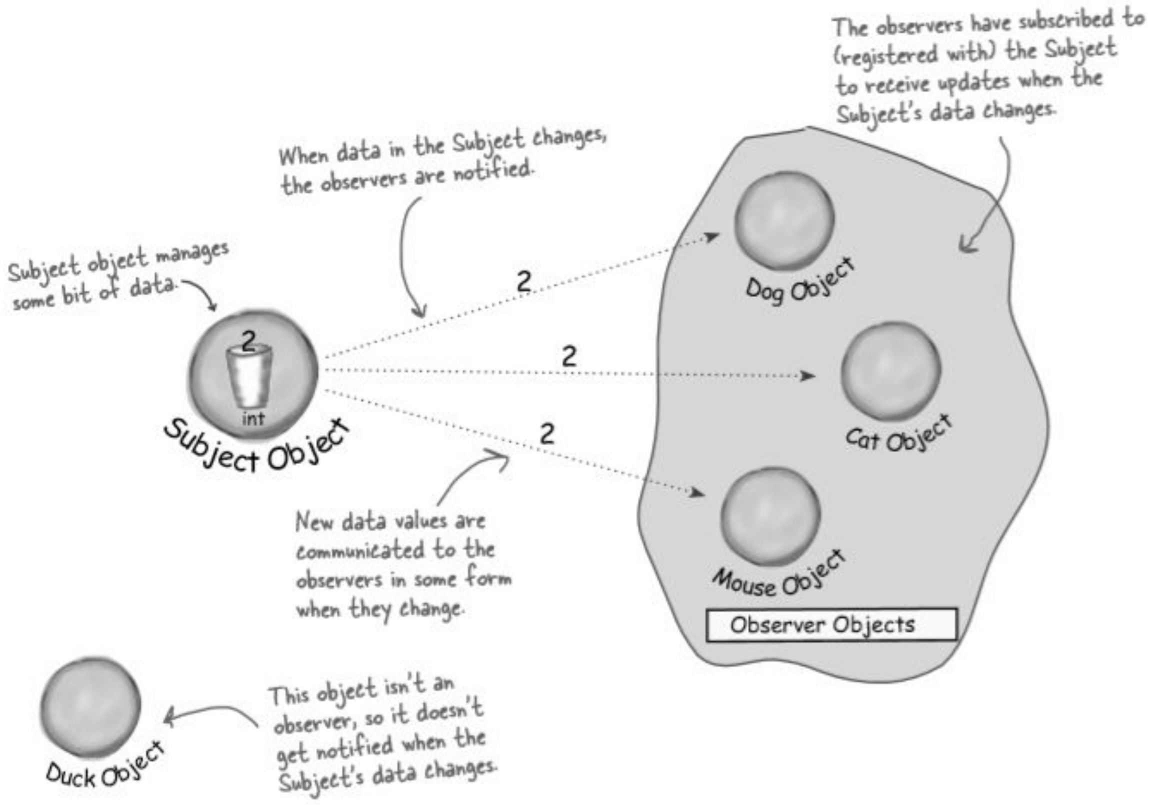
In Strategy pattern, a class behavior or its algorithm can be changed at run time. This type of design pattern comes under behavior pattern.

In Strategy pattern, we create objects which represent various strategies and a context object whose behavior varies as per its strategy object. The strategy object changes the executing algorithm of the context object.


Strategy
<<Interface>>
+doOperation() : int
implements
implements
implements
OpearationAdd
+doOperation() : int
OperationSubstract
+doOperation() : int
OperationMultiply
+doOperation():int
StrategyPatternDemo
+main() : void
asks
Context
-strategy : Strategy
+executeStrategy() : int
uses

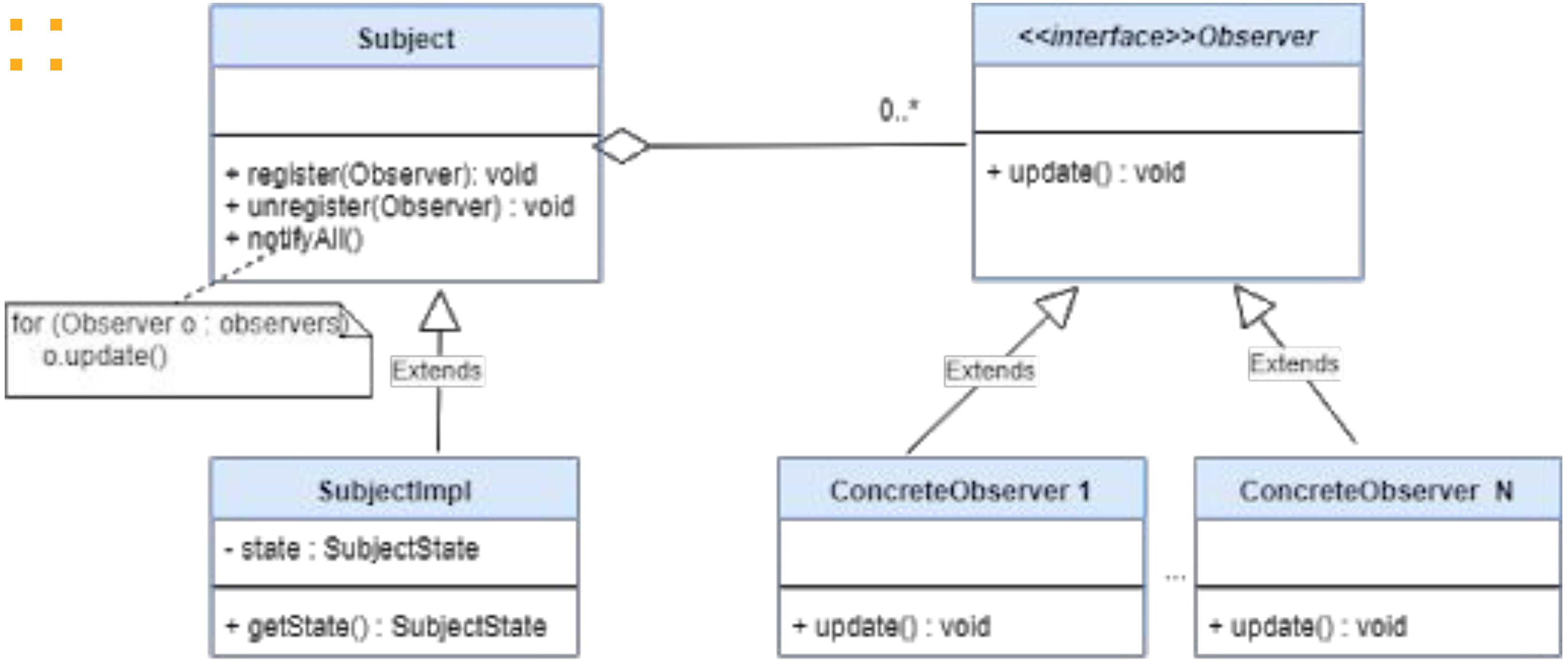
3. Observer Pattern

Keep your objcts in the know


The observers have subscribed to (registered with) the Subject to receive updates when the Subject's data changes.
When data in the Subject changes, the observers are notified.
Subject object manages some bit of data.
2
int
Subject Object
2
2
2
Dog Object
Cat Object
Mouse Object
Observer Objects
New data values are communicated to the observers in some form when they change.
Duck Object
This object isn't an observer, so it doesn't get notified when the Subject's data changes.

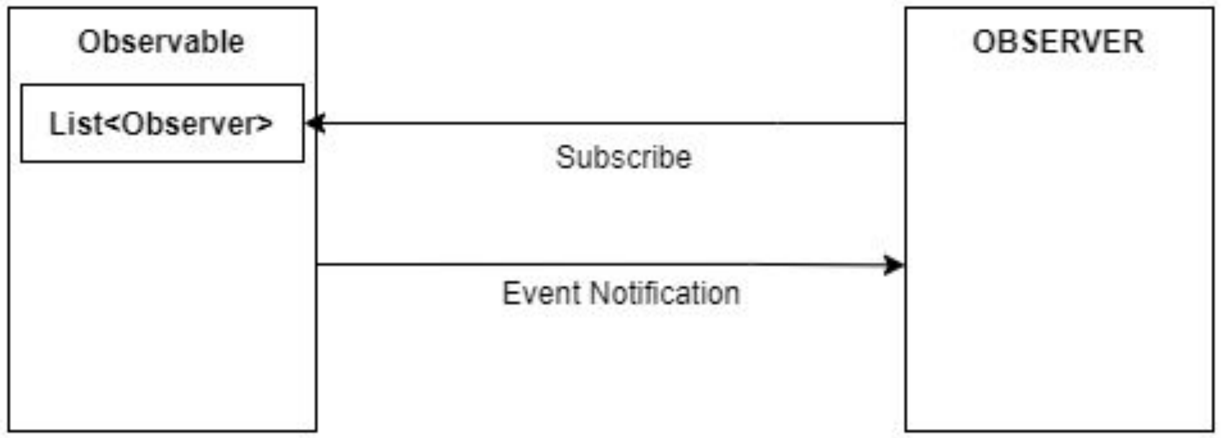
Observer pattern idea

Observer pattern is used when there is one-to-many relationship between objects such as if one object is modified, its depenedent objects are to be notified automatically. Observer pattern falls under behavioral pattern category.


Subject
+ register(Observer): void
+ unregister(Observer) : void
+ notifyAll()
for (Observer o : observers)
o.update()
Extends
SubjectImpl
- state : SubjectState
+ getState() : SubjectState
0..*
<<interface>>Observer
+ update() : void
Extends
Extends
ConcreteObserver 1
+ update() : void
...
ConcreteObserver N
+ update() : void

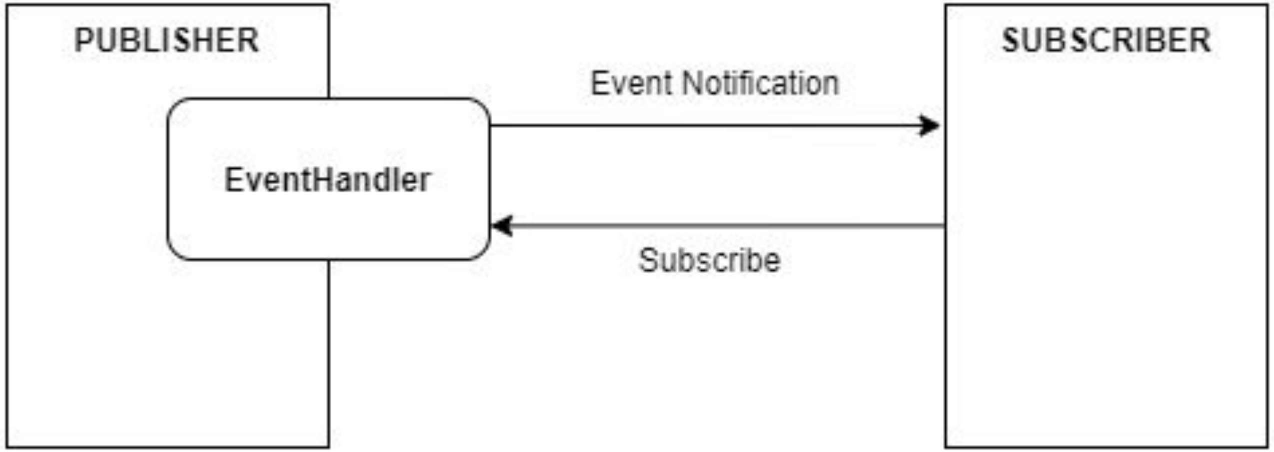
Observer vs Publisher-Subscriber Pattern

In the Observer pattern, Observables (Publishers) must keep track of Observers (Subscribers). The Observable sends events to each Observer.



Observer vs Publisher-Subscriber Pattern

In the Pub-Sub pattern, event notification occurs through the use of an event handler. The publisher does not need to know about subscribers.



Thanks!

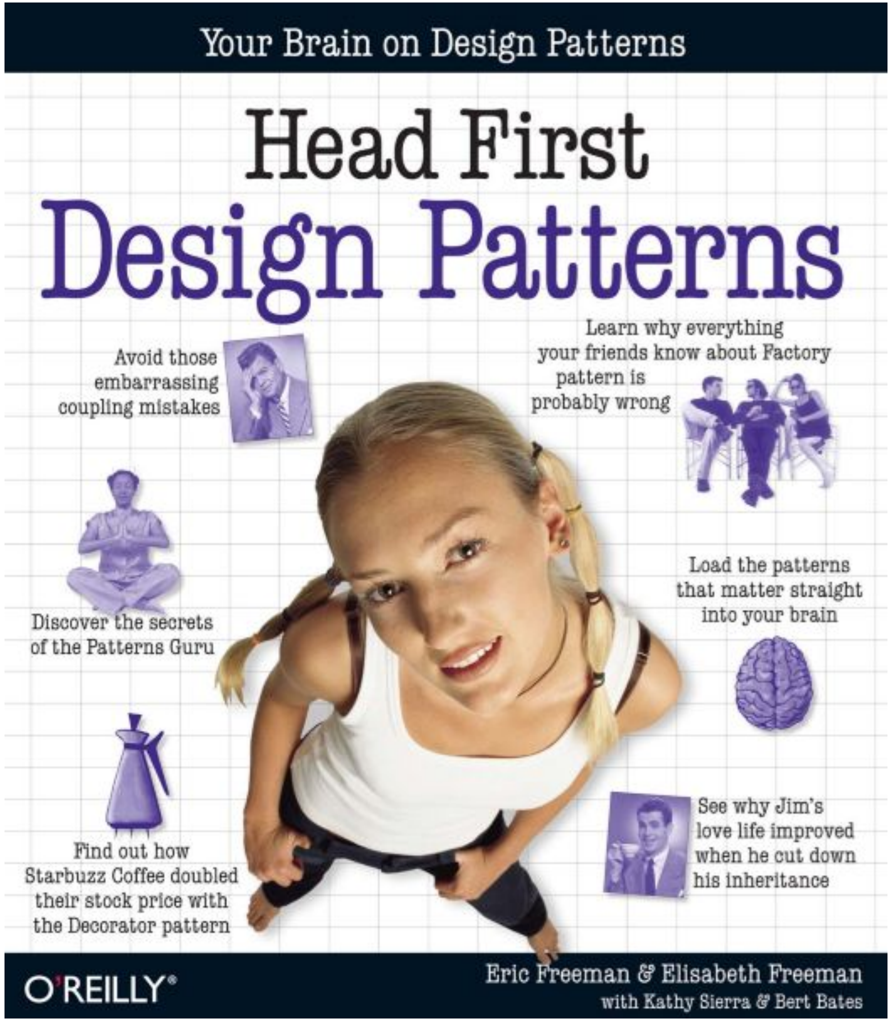
Any questions?

You can find me at:

- @username
- user@mail.me

Credits

- https://refactoring.guru/pt-br/design-patterns
- Head First Design Pattern Eric Freeman and Elisabeth Freeman



Your Brain on Design Patterns

Head First Design Patterns

Avoid those embarrassing coupling mistakes

Learn why everything your friends know about Factory pattern is probably wrong

Discover the secrets of the Patterns Guru

Load the patterns that matter straight into your brain

Find out how Starbuzz Coffee doubled their stock price with the Decorator pattern

See why Jim's love life improved when he cut down his inheritance

O'REILLY®

Eric Freeman & Elisabeth Freeman with Kathy Sierra & Bert Bates