



Introduction to Design Patterns

Micky

Growth Session #26 - July 25-26 2019

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

- Introduce what is “Design Pattern”
- Lots of examples and illustrations make it easier to understand
- All of example codes are in Java

Intro

What is a design pattern ?

- Guideline or practice to solve problems that can occur in software development.

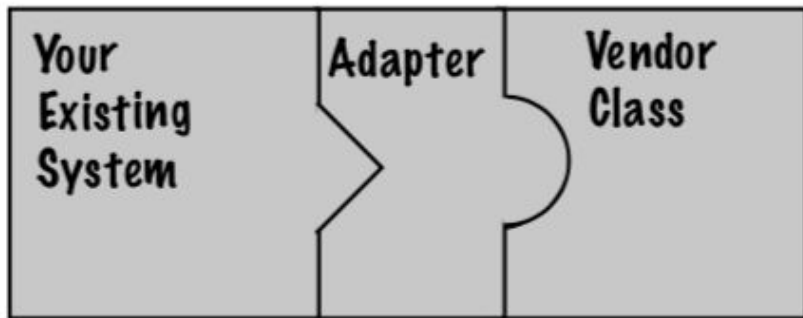
Why we should know about it ?

- Improve quality of design and code structure. Makes it easier to reuse, extend and maintain code.
- Help in communication between developers.



Adapter pattern

“ **The Adapter Pattern** converts the interface of a class into another interface the clients expect.”



- Adapter lets classes work together that couldn't otherwise because of incompatible interfaces.
- Encapsulates the change so we don't have to modified the client class when there are changes in vendor class

Adapter pattern - Example



DASHBOARD



M

MIA - Web

Roadmap

Unplanned Epics



TIME

Sprint **26**

5 TEAM MEMBERS



An, Yutthana, Raju, Hoang, Terone

LINK

[Pivotal Tracker](#)

Feature - Advertiser Can Authenticate (Email)

Release - 0.5.0

Release - 0.6.0

Release - 0.7.0

Release - 0.8.0

Release - 0.9.0

Release - 0.10.0

Adapter pattern - Prepare adapter

```
module Athena
  module PivotalTracker
    class Client
      attr_reader :client

      def initialize
        @client ||= TrackerApi::Client.new(token: ENV.fetch('PIVOTAL_TRACKER_API_TOKEN'))
      end

      def project(project_id)
        client.project(project_id)
      rescue TrackerApi::Errors::ClientError => exception
        handle_exception(exception)
      end

      # ... Some other methods here

      def epics(project)
        project.epics
      end

      # ...Also Some other methods here too

    end
  end
end
```

- Use gem “**TrackerApi**” to fetch data from Pivotal Tracker.
- Initialize client with Pivotal Tracker API Token.
- The method **project** fetches data of a project by a given project ID.
- The method **epics** fetches all epics in a project.

Adapter pattern - Using adapter

```
module AthenaClient
  module Integration
    module PivotalTracker
      class EpicSynchronization < Integration::EpicSynchronizationBase
        def call!
          project = client.project(integration_id)
          epics = client.epics(project)

          # Synchronized fetched epics in Athena
        end

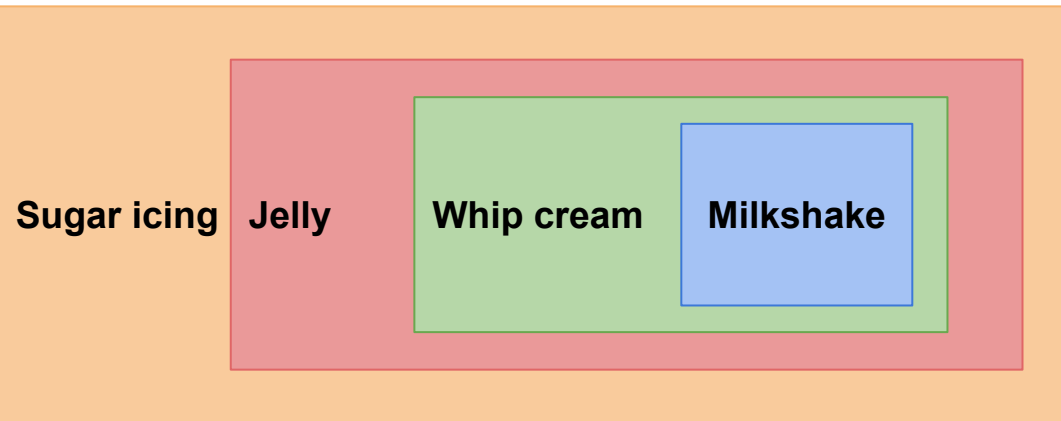
        private

        attr_reader :project_id

        def client
          @client ||= Athena::PivotalTracker::Client.new
        end
      end
    end
  end
end
```

Decorator pattern

“ **The Decorator Pattern** attaches additional responsibilities to an object dynamically.”



- Decorators provide a flexible alternative to subclassing for extending functionality.
- Class can be wrapped with any numbers of decorators.

Decorator pattern - Example

```
class CommentDecorator < SimpleDelegator
  def display_username_for(current_user = nil)
    return "Braive" if user.admin_role?
    return "You" if patient_owner?(current_user)
    return user.username if current_user.present? && current_user.admin_role?

    "Anonymous"
  end

  # Other code
end
```

```
class Comment < ApplicationRecord

  # Some code goes here...

  delegate :display_username_for, :patient_owner?, to: :comment_decorator

  def comment_decorator
    @comment_decorator = CommentDecorator.new(self)
  end

  # Other code goes here...

end
```

Decorator pattern - Example

```
tr.comment class=["depth-#{comment.depth}", ('is-active' if is_active)]
  - unless comment.root?
    td.icon-container
      = image_tag '/images/mailer/icon-reply.png', class: 'icon icon-reply'
    td
      - if comment.deleted_at?
        .comment-content
          = t('shared.comment.deleted_notice')
      - else
        .comment-content
          h4 = comment.display_username_for(user)
          p = comment.body
        .comment-footer
          = l(comment.created_at, format: :date)
```

Anonymous

Help me goddam it!

18 January 2018 [Reply](#)



Braive

What do you want?

31 January 2018 [Reply](#)



This comment has been removed by its author.



Maybe someone has already
solved your problems

Thanks!

Contact Nimble

nimblehq.co

hello@nimblehq.co

Bangkok

399 Interchange 21 Sukhumvit Road, Unit
#2402-03, Klong Toei, Wattana, Bangkok
10110, Thailand

Singapore

28C Stanley St, Singapore 068737

Hong Kong

20th Floor, Central Tower
28 Queen's Road, Central, Hong Kong

