

Instructions



How is it, to be a final year student???

0 responses



Do you have problems with practice?

Agenda

- Recall what was learned
- Look at current topic practice
- Check current topic knowledge

Total Recall



Is shallow copy allowed for immutable object?

0 

yes

0 

sometimes

0 

no

When do we use singleton pattern?

0

For every service

0

When need only one instance
for the object

0

To reduce boilerplate code

Factory method is a ... pattern.



structural



behavioral



creational



architectural

Which pattern "Allows a client to create families of objects without specifying their concrete classes"?

0 

Prototype

0 

Factory Method

0 

Abstract Factory



Discuss current topic



Polar bears have black skin

Have you looked at todays' materials?

0



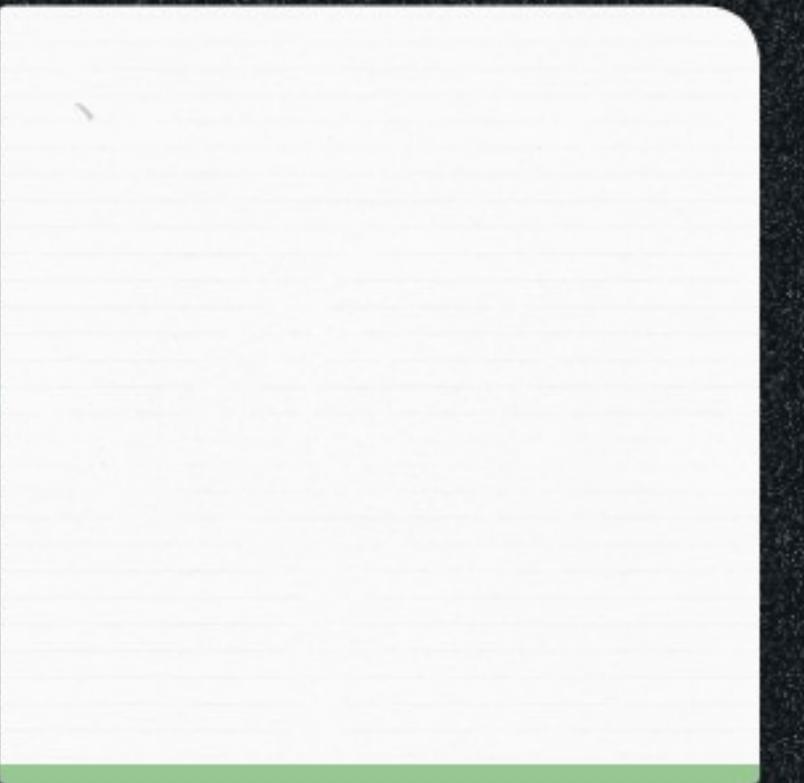
yes

0



no

0



partially

0



i'm the author

Basic Patterns: Facade

After completing this lesson, you will be able to:

- Explain the feasibility of using the facade pattern to reduce dependencies in a complex class system and make it easier
- Implement the facade pattern to hide system complexity and provide customers with a single point of entry to access its

What is a facade in programming architecture?

- A long columned porch that serves as a hallway ✖
- An open garden surrounded by an enclosed portico ✖
- A design pattern that provides a simplified interface for a complex subsystem ✖
- An external wall of a building, often the front wall with the entrance ✖

An object structural pattern that provides a unified interface to a set of interfaces in a subsystem:

0 

Abstract Factory

0 

Composition

0 

Facade

Which of the following is true for facade pattern?

- a) It is a broker pattern that eases interaction between a client and a sub-system of suppliers by providing a simpler interface to the sub-system ✖
- b) Systems may have many classes with complex interfaces, but often clients need basic services that can be supplied through simple interfaces ✖
- c) Broker class, called a façade, provides basic, simplified services to clients by taking upon itself the job of dealing with a complex sub-system ✖
- d) All of the mentioned ✓

Which of the following is for Facade Pattern?

- a) Interaction is begun by one of the collaborators or a client object outside the collaboration ✖
- b) The object directs the collaboration, with all communication going between it and the colleagues in the interaction ✖
- c) None of the mentioned ✓
- d) All of the mentioned ✖

Pros of Facade pattern

0 ✓

You can isolate your code from the complexity of a subsystem.

0 ✗

A facade can become a god object coupled to all classes of an app.

0 ✗

You can create platform-independent classes and apps.

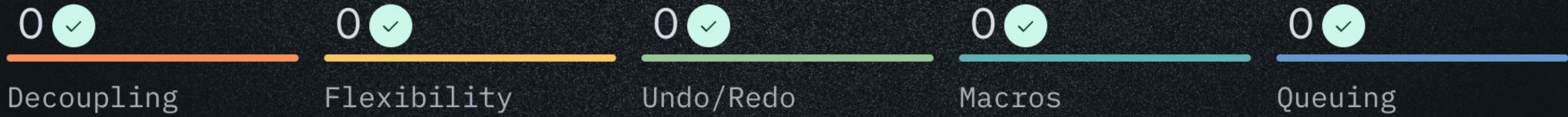
10 mins break

Convenient Patterns: Command

After completing this lesson, you will be able to:

- Explain benefits of using the command pattern to pass request objects as arguments and enqueue, log, and cancel request
- Use the command pattern to encapsulate a request in an object and delegate it to other objects for execution

Benefits of Command pattern



Which of the several aspects of the Command pattern may be varied or elaborated to help achieve particular design goals

- a) An invoker may accept registration of more than one command
- b) The invoker may offer operations for un-registering one or more commands
- c) An invoker may offer several kinds of registrations for different kinds of events
- d) All of the mentioned

Which of the following are consequences for command pattern?

0

- a) Reactor patterns provide event-driven design
models



0

- b) They decouple clients and targets



0

- c) Encapsulate reactions to event



0

- d) None of the mentioned



Which of the following are the consequences for Command pattern?

- a) Event-driven design is facilitated
- b) Making event-handling code easy to find, change, and reuse
- c) The invoker is loosely coupled to both the command and client classes, making them easy to change
- d) All of the mentioned



Q&A part

0 questions
0 upvotes