

The Command Pattern

Design Patterns



Motivating Example

Command
Pattern

- A command-line order management system
- Existing orders may be edited, and a log of all changes must be kept

Intent

Command
Pattern

- Represent an action as an object
- Decouple clients that execute the command from the details and dependencies of the command logic
- Enables delayed execution
 - Can queue commands for later execution
 - If command objects are also persistent, can delay across process restarts

Also Known As

- Action, Transaction

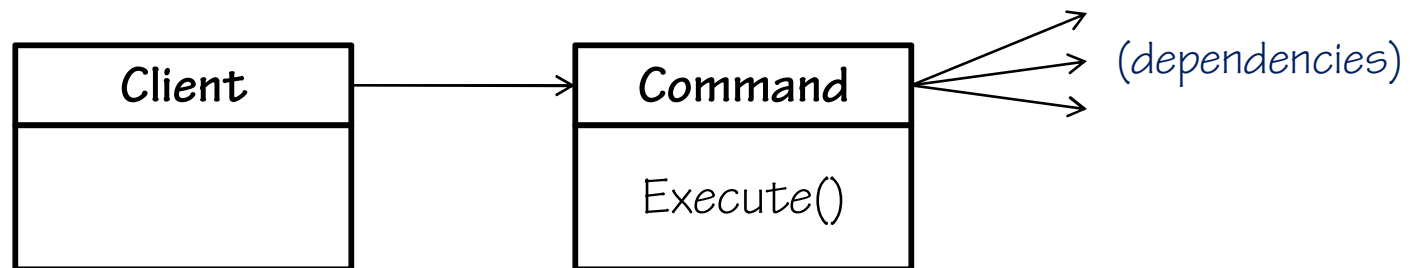
Command
Pattern

Applicability

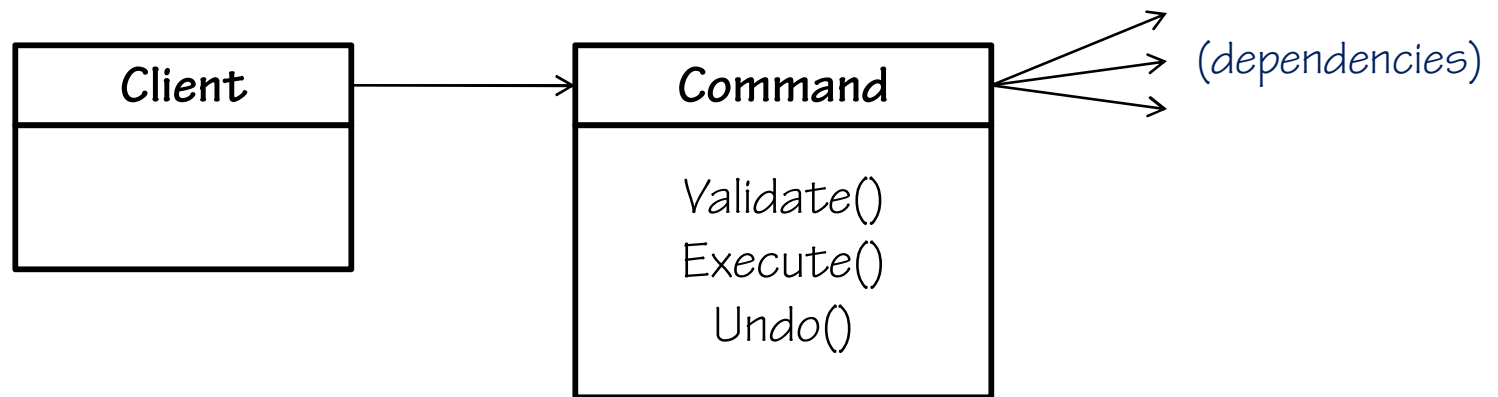
Command
Pattern

- Logging
- Validation
- Undo

Structure



Taking it Further



Consequences

Command
Pattern

- **Commands must be completely self contained**
 - The client doesn't pass in any arguments
- **Easy to add new commands**
 - Just add a new class (open/closed principal)

Implementation Example

Command
Pattern

- A command-line order management system
- Existing orders may be edited, and a log of all changes must be kept

Related Patterns

Command
Pattern

- **Factory Pattern**
 - Factories are often useful to construct command objects
- **Null Object**
 - Often times returning a “null command” can be useful instead of returning null
- **Composite**
 - A *composite command* can be useful
 - Construct it with several “child” commands
 - Execute() on the composite will call Execute() on the child commands

Summary

Command Pattern

- **Consider the command pattern**
 - When want to decouple the client that executes the command from the command logic and its dependencies
 - When you're building a command-line application
 - When you're implementing Validation
 - When you're implementing Undo