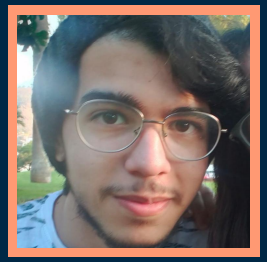
Design Patterns

Diego Herrera Mendoza
Roberto Carrazana Pernía

Index

About us



Diego Herrera Mendoza

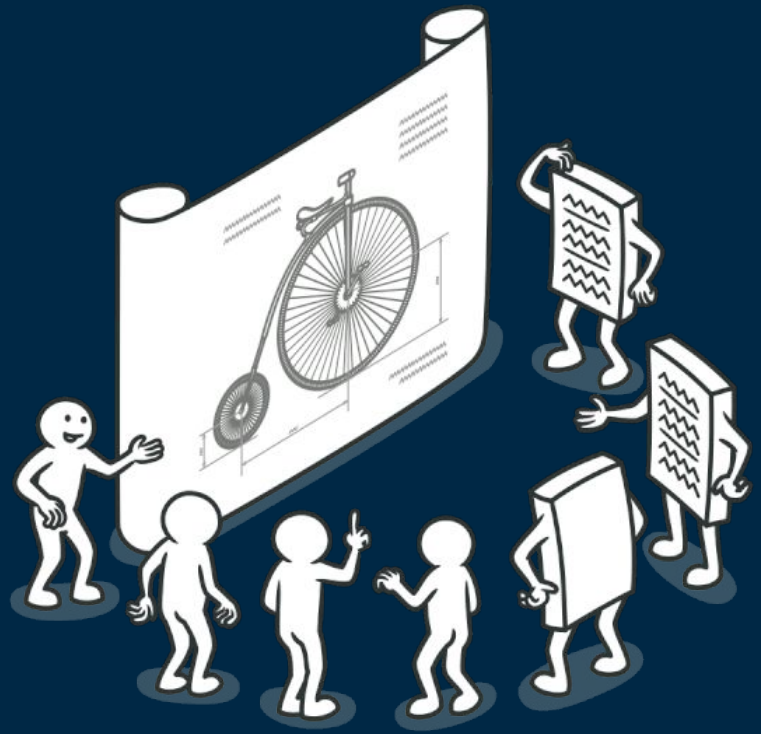
diego.herrera.26@ull.edu.es

Roberto Carrazana Pernía

roberto.carrazana.12@ull.edu.es

What is a design pattern?

Typical solutions to commonly occurring problems in software design



Why should we use them?

- Tried and tested solutions
- They define a common language

Be careful!

Unjustified use:

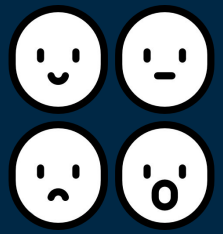
“If all you have is a hammer, everything looks like a nail.”

Classification

C — Creational

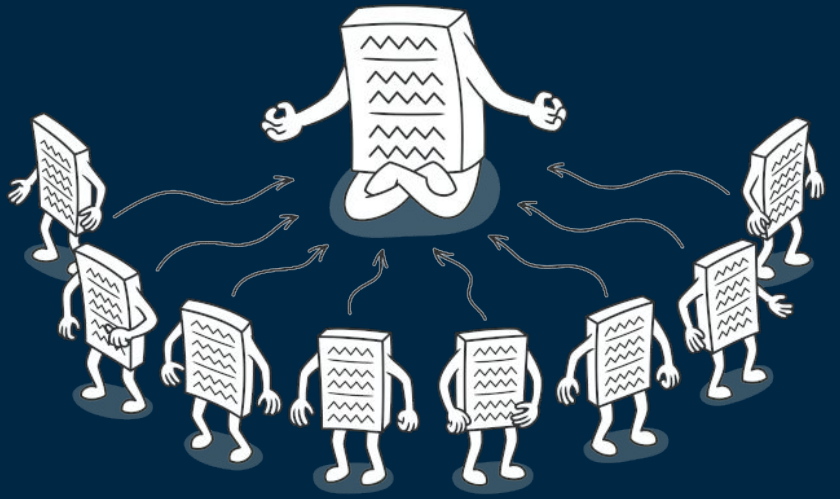
S — Structural

B — Behavioral

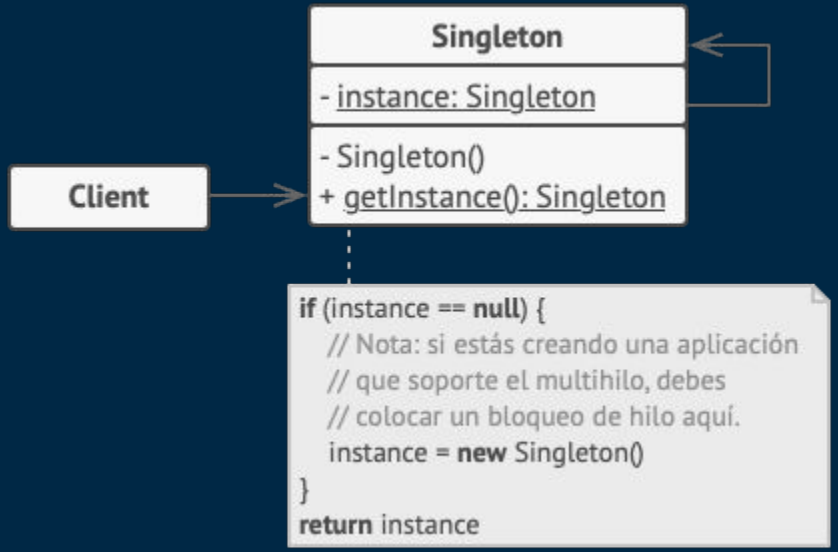


Creationals – Singleton

Ensure that a class has only one instance, while providing a global access point to this instance.



Creationals – Singleton – UML

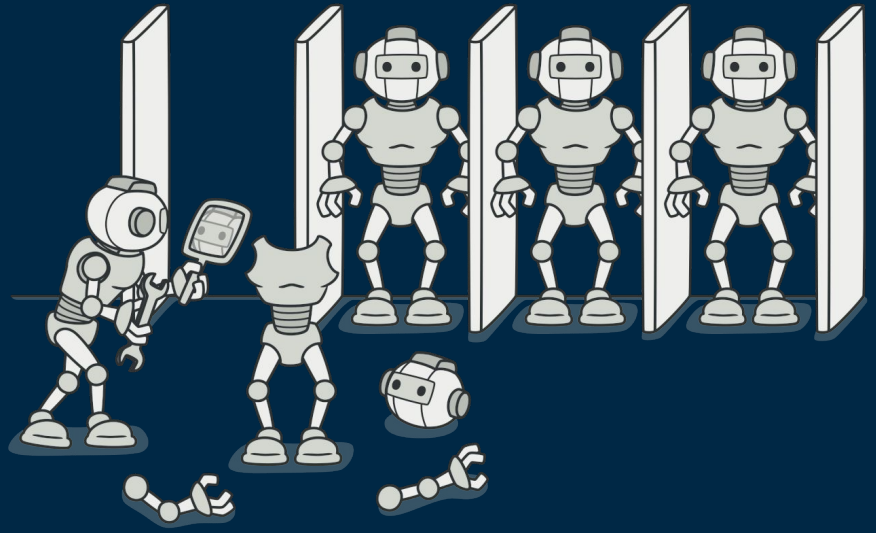


Singleton – Advantages and Disadvantages

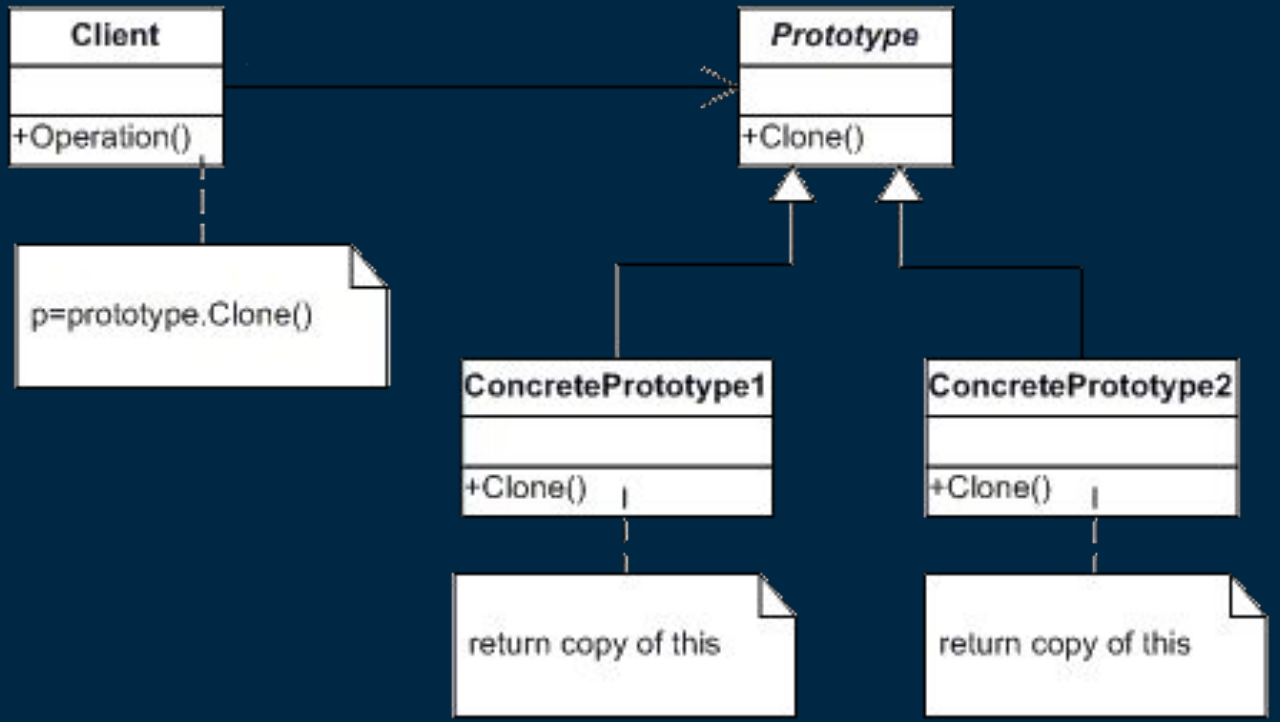
Advantages	Disadvantages
Ensures that the class has only one instance.	Violates the Single Responsibility Principle.
Global access to that instance.	Requires special treatment in a multithreaded environment.
Instance initialized only when requested.	Could mask a bad design.

Creationals – Prototype

Specifies the kind of objects to create using a prototypical instance, and create new objects by copying this prototype.



Creationals – Prototype – UML

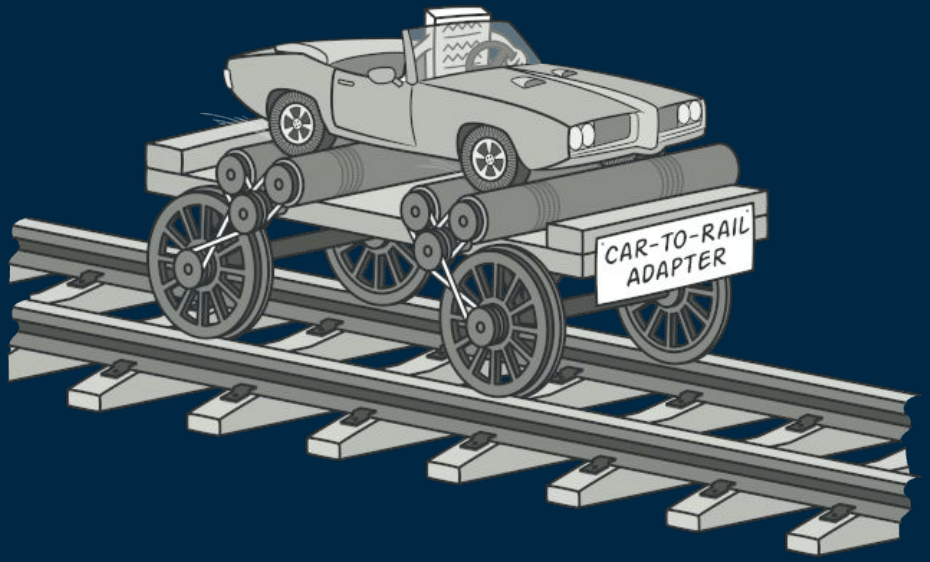


Prototype – Advantages and Disadvantages

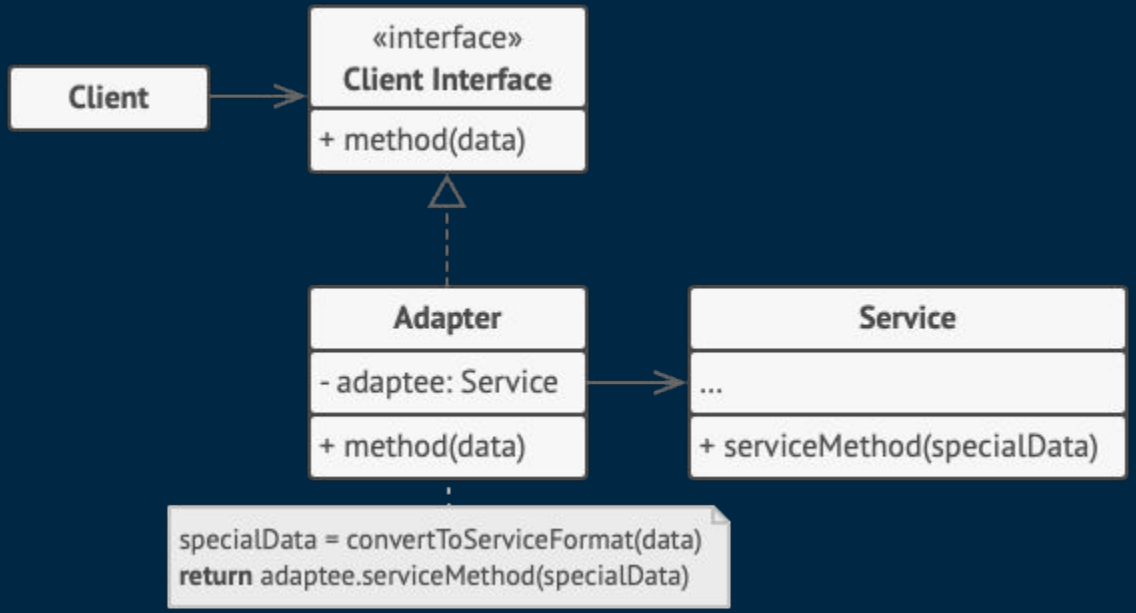
Advantages	Disadvantages
Get rid of repeated initialization code	Circular references might be very tricky.
Make complex objects more conveniently.	

Structurals – Adapter

Allows objects with incompatible interfaces to collaborate.



Structurals – Adapter – UML



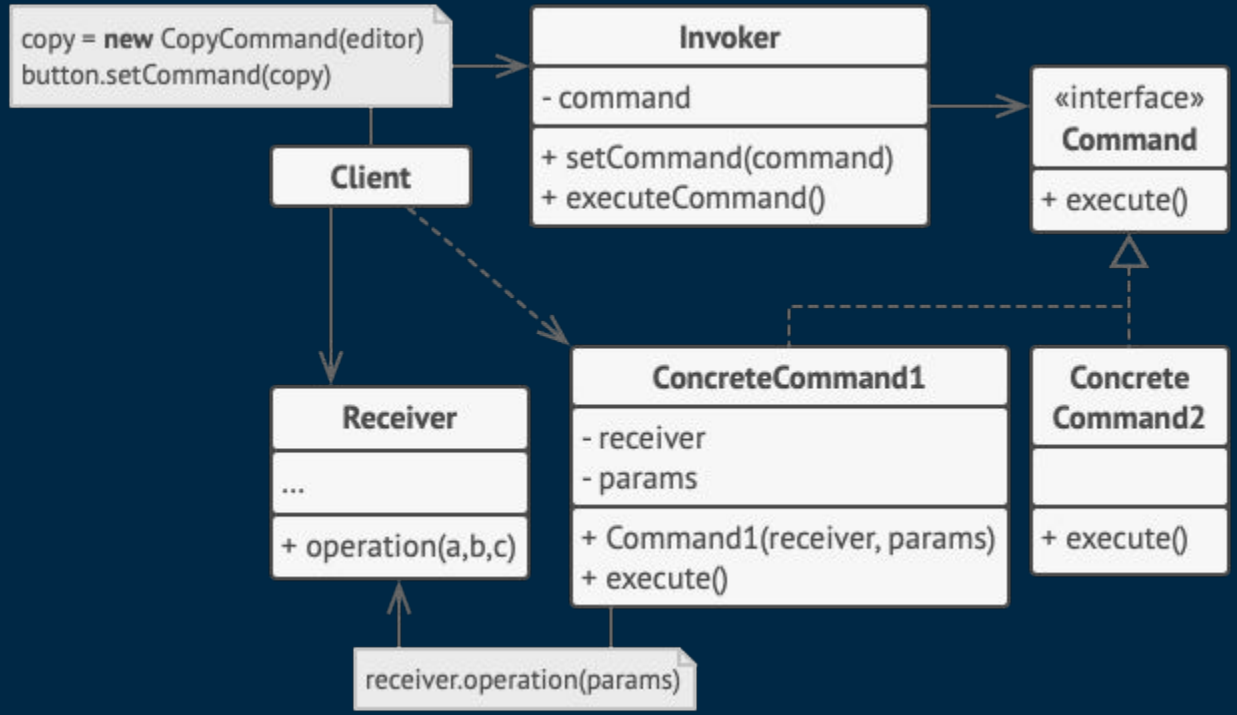
Adapter – Advantages and Disadvantages

Advantages	Disadvantages
Single Responsibility Principle.	Increases overall complexity
Open/Closed Principle.	

Behavioral – Command

Encapsulates a request as an object, thereby letting you parameterize clients with different requests, queue or log requests, and support undoable operations.

Behavioral – Command – UML

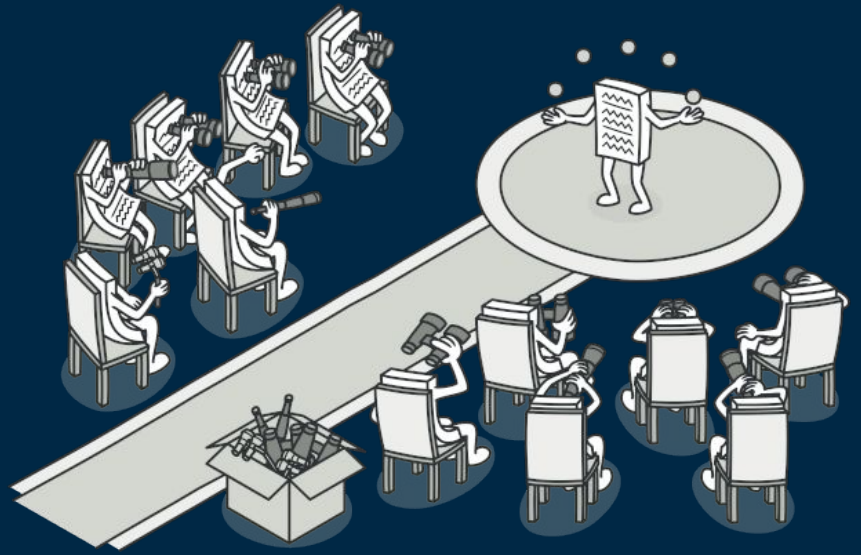


Command – Advantages and Disadvantages

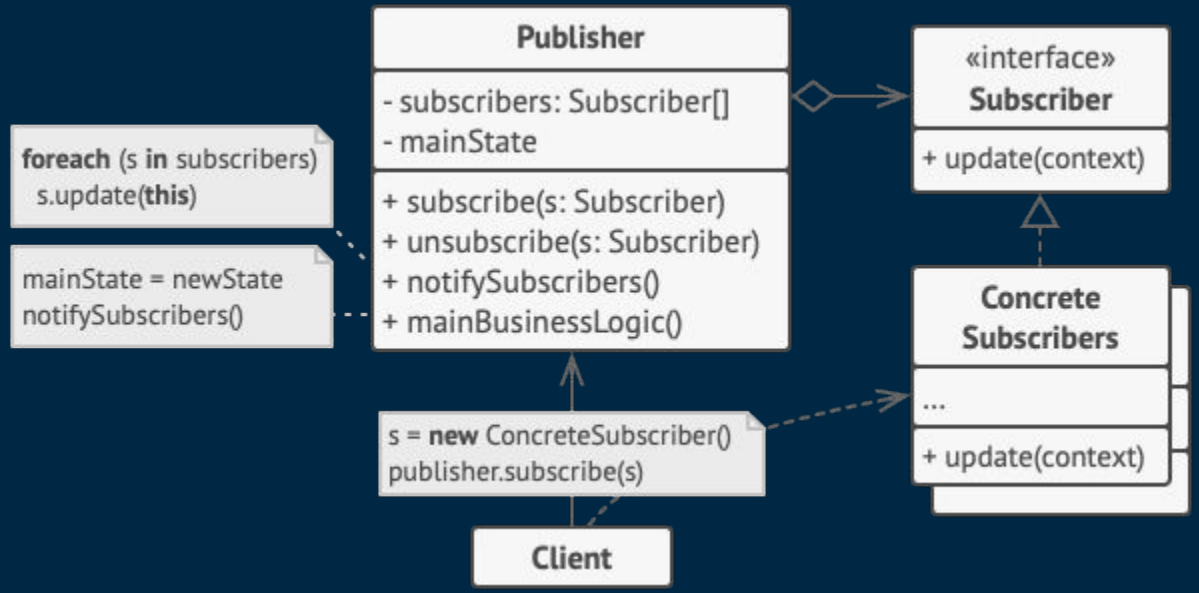
Advantages	Disadvantages
Single Responsibility Principle.	New layer between senders/receivers.
Open/Closed Principle.	
Undo/redo	
Simple commands into a complex one.	

Behavioral – Observer

Define a subscription mechanism to notify multiple objects about any events that happen to the object they're observing.



Behavioral – Observer – UML



Observer – Advantages and Disadvantages

Advantages	Disadvantages
Can establish relations at runtime.	Notifications in random order
Open/Closed Principle.	

References

https://refactoring.guru/es – Guide to refactoring, design patterns, SOLID...

https://dofactory.com/net/design-patterns – Design patterns with examples

https://java-design-patterns.com/patterns/ – Some design patterns in Java

https://methodpoet.com/disadvantages-of-singleton-pattern/ – Singleton problems and how to deal with them

https://www.youtube.com/watch?v=tv-_1er1mWI&ab_channel=Fireship – 10 design patterns in 10 minutes

Thank you for your attention!

Repo:
https://github.com/ULL-ESIT-PAI-2022-2023/2022-2023-pai-design-patterns-design-patterns-pai.git

Any questions?