

Design Patterns

Diego Herrera Mendoza
Roberto Carrazana Pernía

Index

About us → 3

What is a design pattern? → 4

Why we should use them? → 5

Be careful! → 6

Classification → 7

Creationals - Singleton → 8

Creationals - Singleton - UML → 9

Creationals - Prototype → 10

Creationals - Prototype - UML → 11

Structurals - Adapter → 12

Structurals - Adapter- UML → 13

Behavioral - Commander → 14

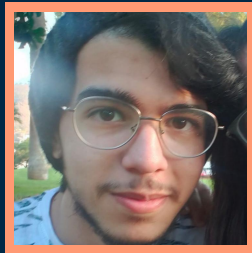
Behavioral - Commander - UML → 15

Behavioral - Observer → 16

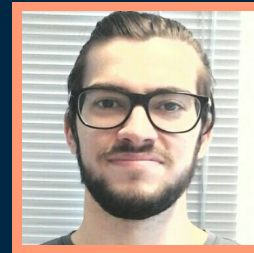
Behavioral - Observer - UML → 17

References → 18

About us



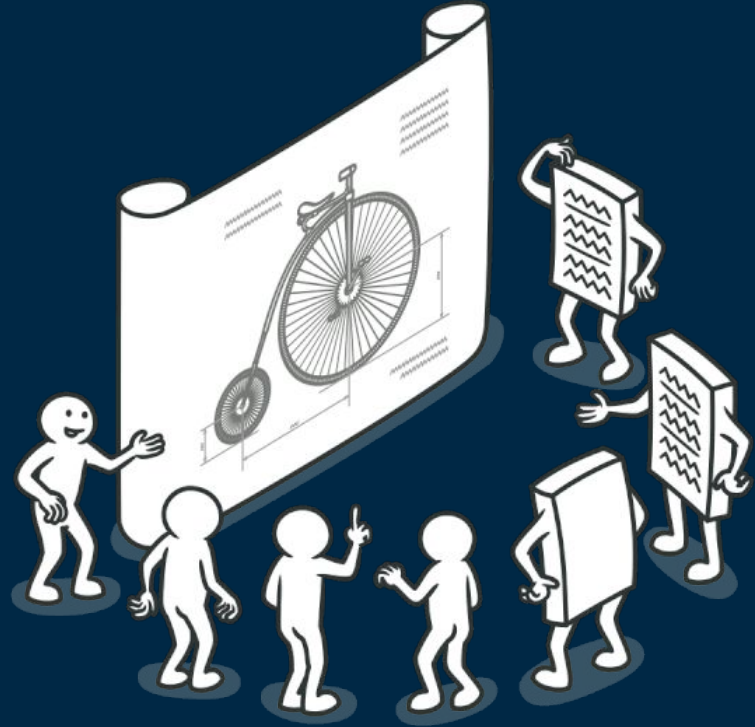
Diego Herrera Mendoza
diego.herrera.26@ull.edu.es



Roberto Carrazana Pernía
roberto.carrazana.12@ull.edu.es

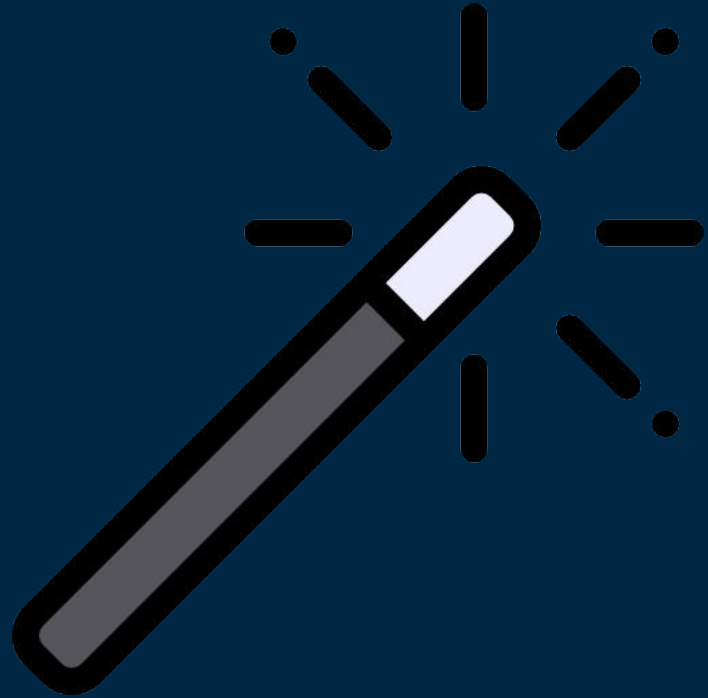
What is a design pattern?

Typical solutions to
commonly occurring
problems in software
design



This is not magic!

We can't choose a
pattern and paste it in
our program as
functions or libraries.



Patterns and algorithms

Patterns	Algorithms
High level description	Group of actions with a clear objective
The user adapts the implementation to his code	The implementation is mostly the same



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



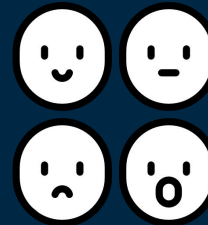
Creational



Structural

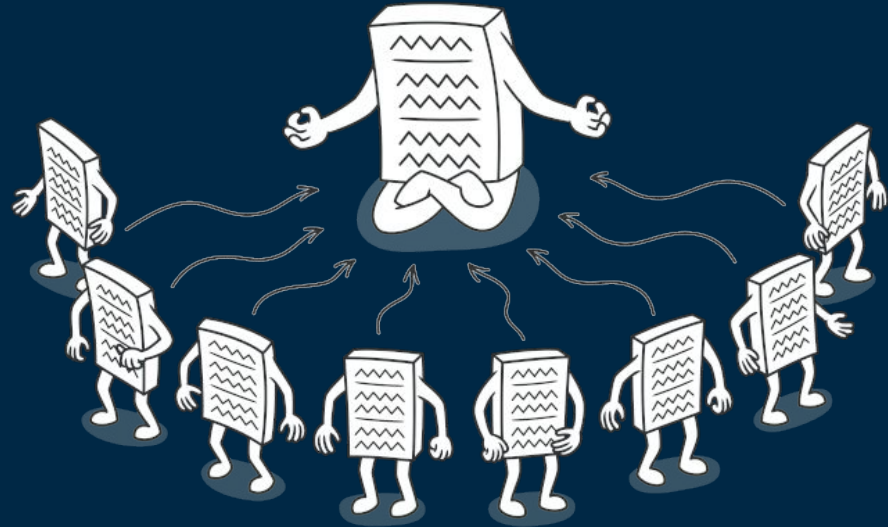


Behavioral



Creationals – Singleton

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



Frequency
of use

1

2

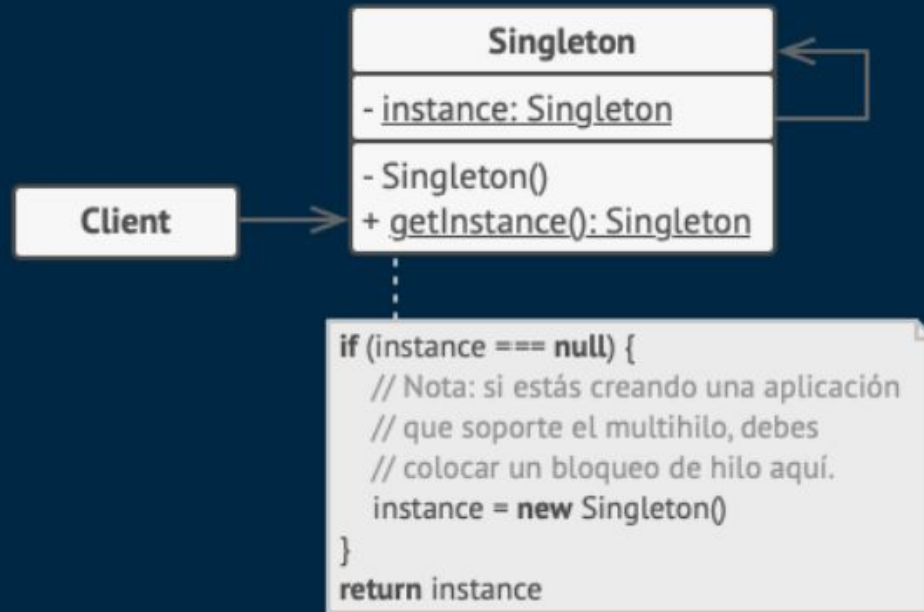
3

4

5



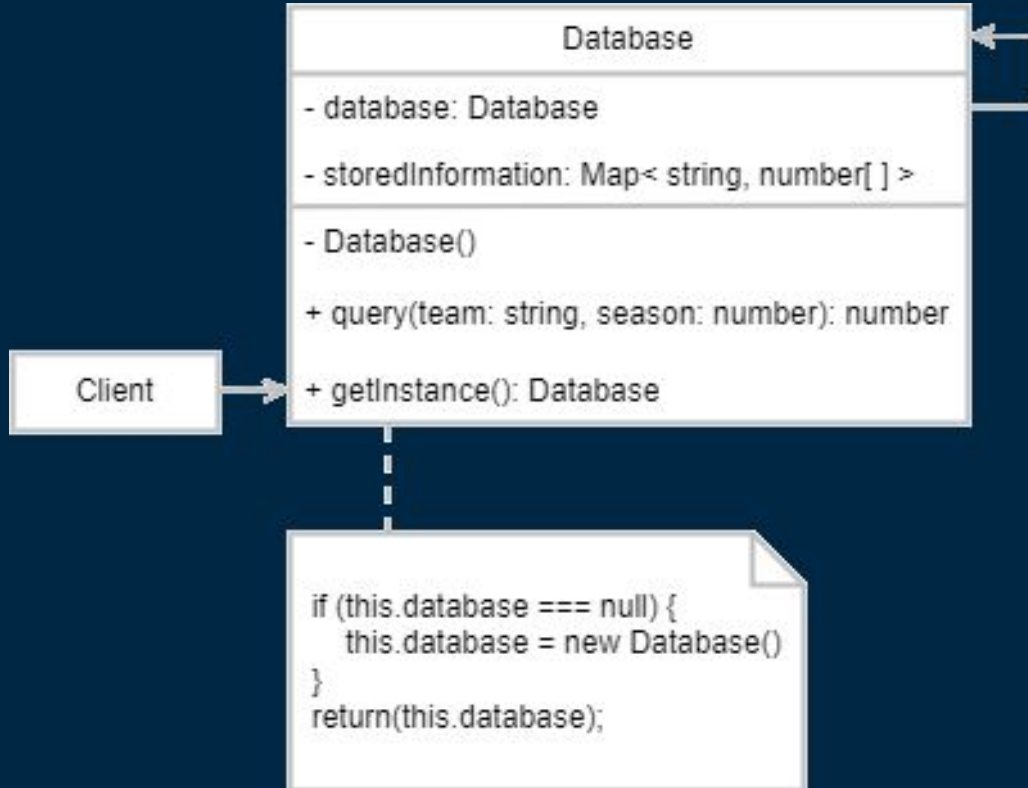
Creationals – Singleton – UML



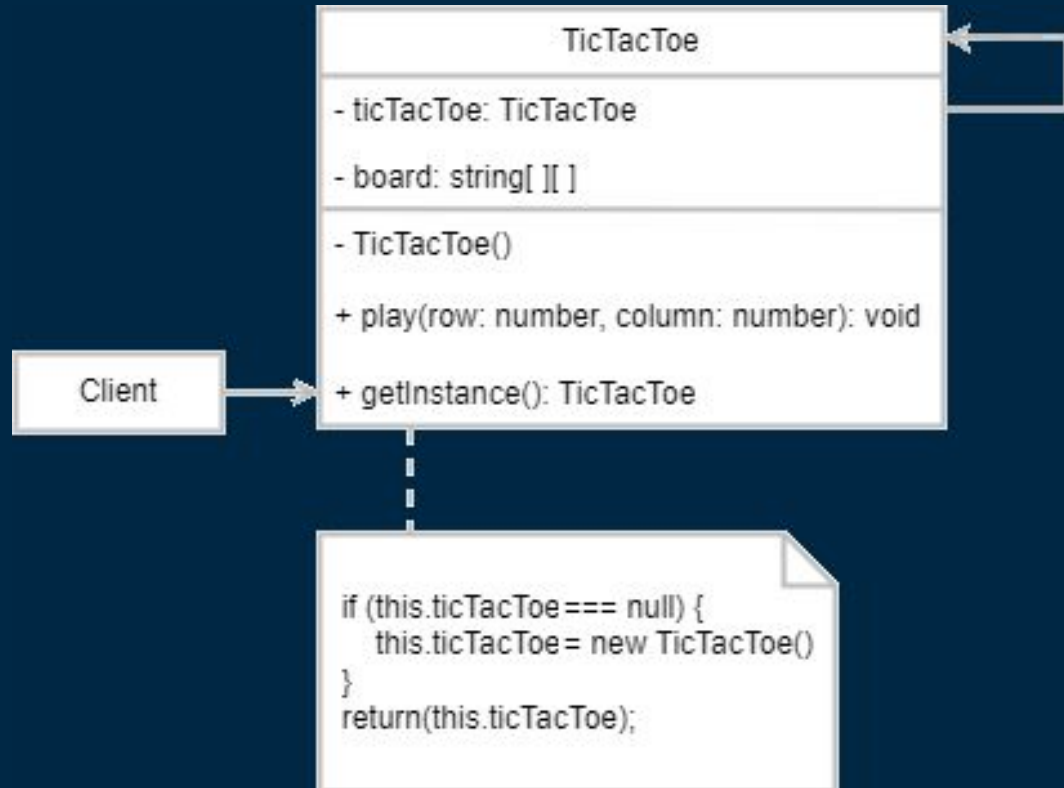
Singleton – Advantages and Disadvantages

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 – Singleton – Database – UML

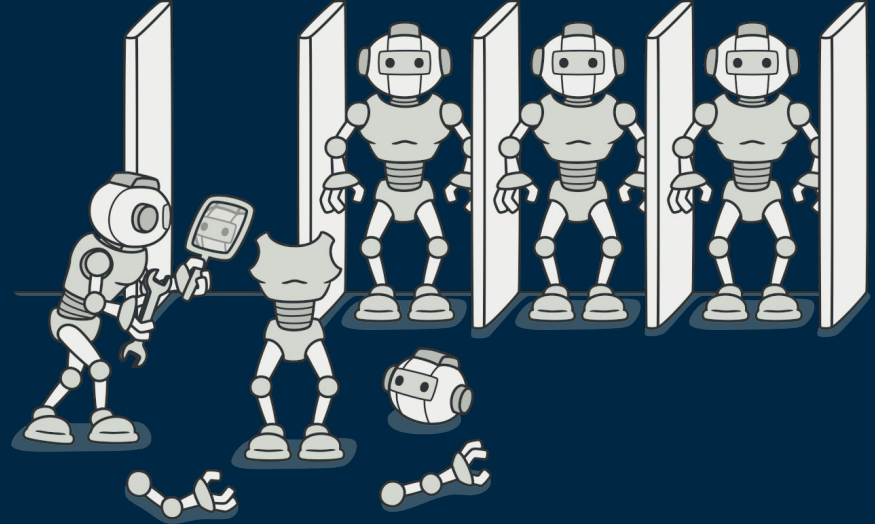


Creationals - Singleton - TicTacToe - UML

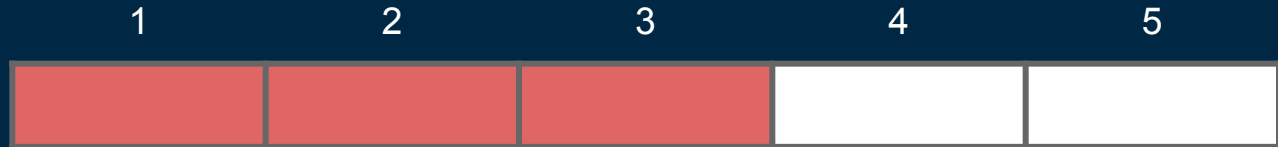


Creationals – Prototype

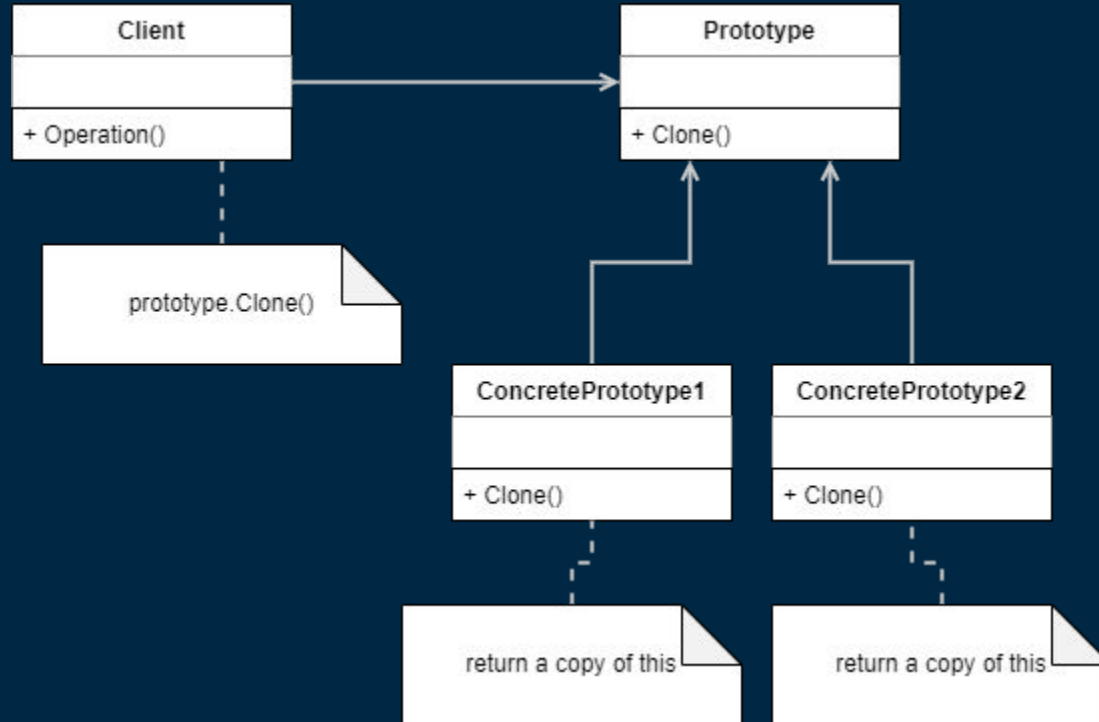
Specifies the kind of objects to create using a prototypical instance, and create new objects by copying this prototype.



Frequency
of use



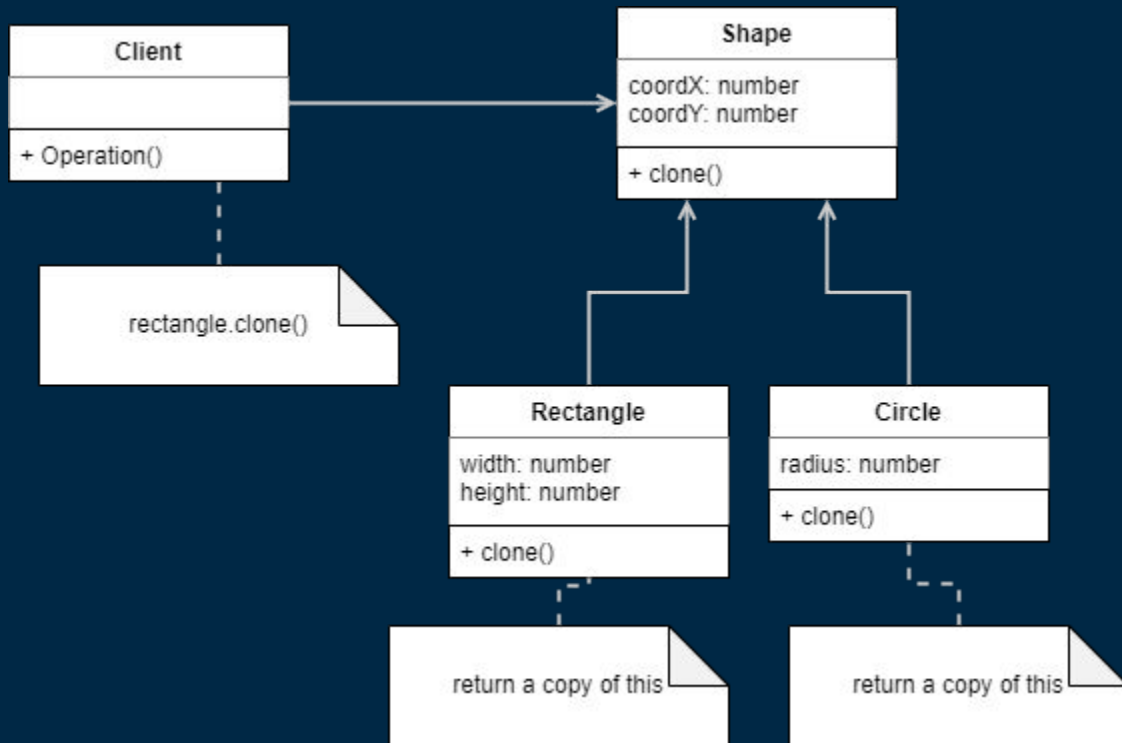
Creationals - Prototype - UML



Prototype – Advantages and Disadvantages

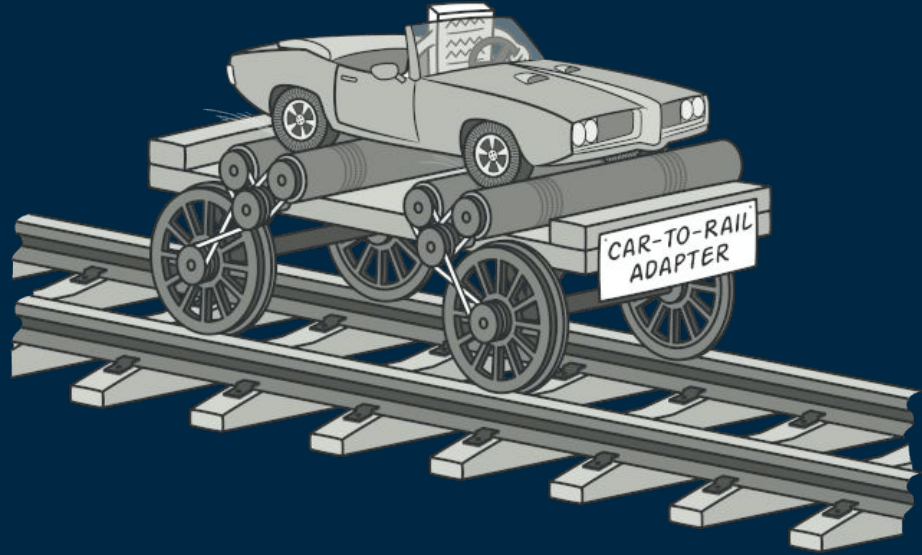
Advantages	Disadvantages
Get rid of repeated initialization code	Circular references might be very tricky.
Make complex objects more conveniently.	

Creationals - Prototype - UML



Structurals – Adapter

Allows objects with incompatible interfaces to collaborate.



Frequency
of use

1

2

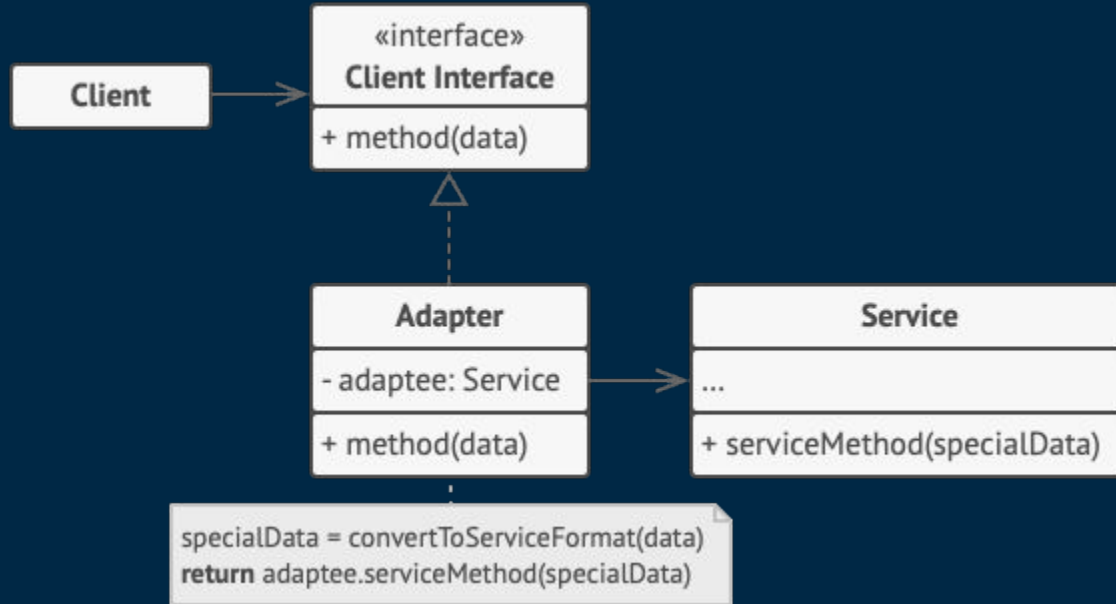
3

4

5



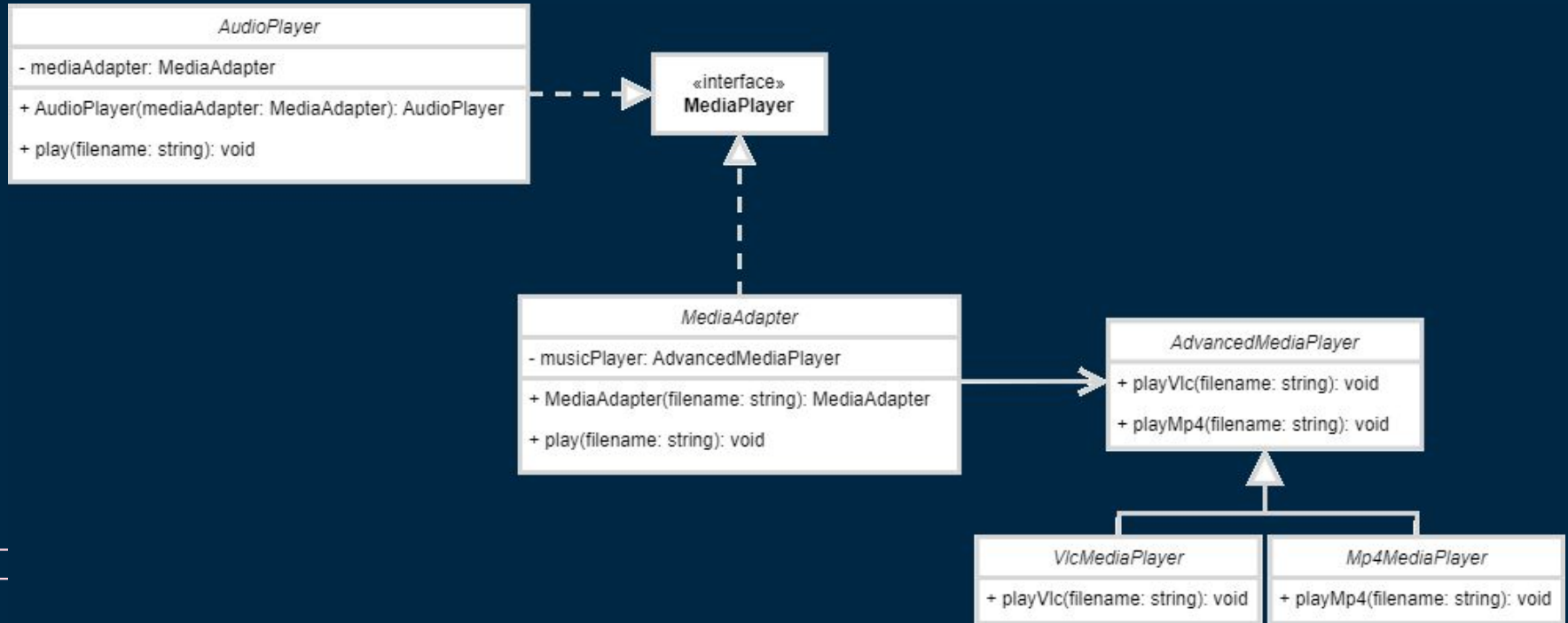
Structurals - Adapter - UML



Adapter – Advantages and Disadvantages

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

Structurals – Adapter – UML – MediaPlayer



Behavioral – Command

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



Frequency
of use

1

2

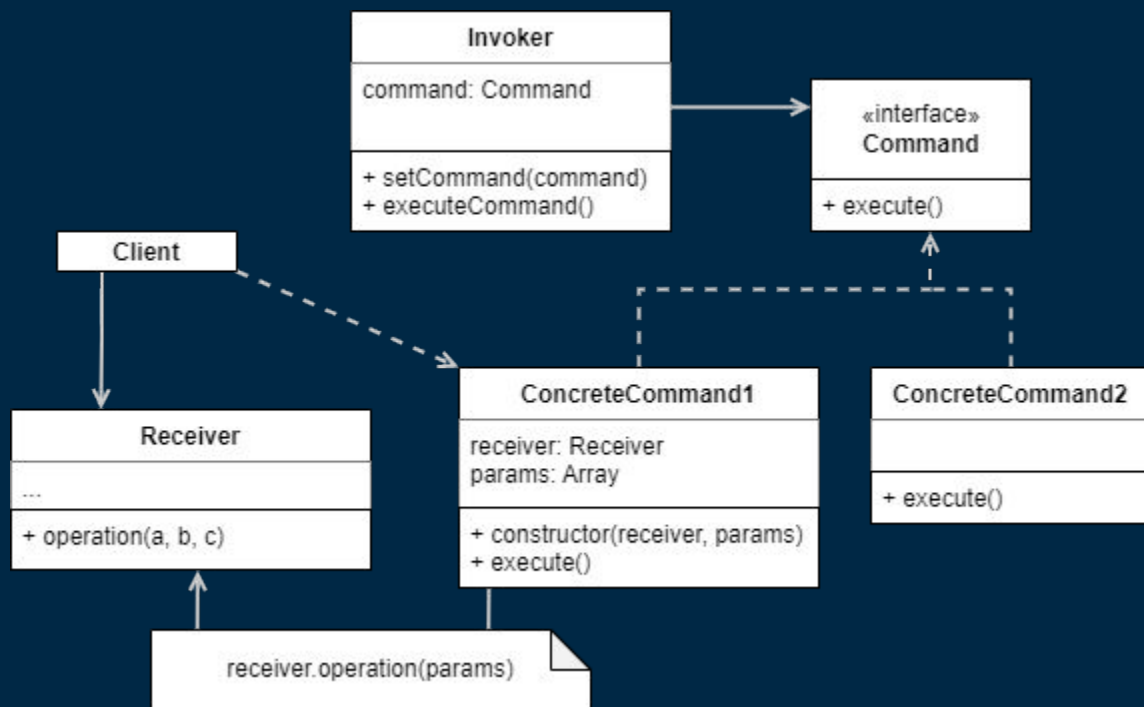
3

4

5



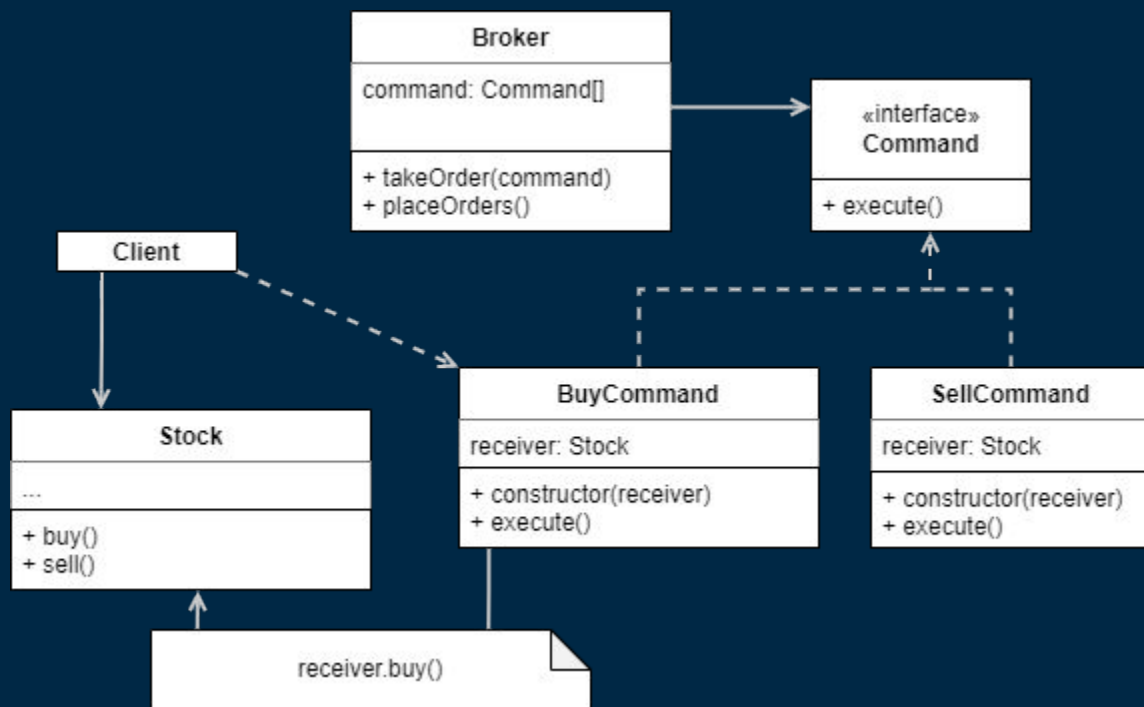
Behavioral - Command - UML



Command – Advantages and Disadvantages

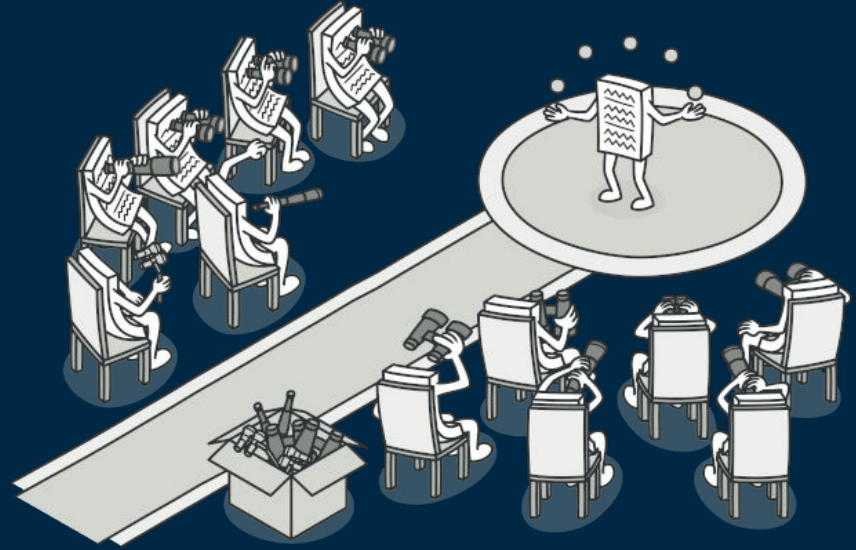
Advantages	Disadvantages
Single Responsibility Principle.	New layer between senders/receivers.
Open/Closed Principle.	
Undo/redo	
Simple commands into a complex one.	

Behavioral - Command - UML



Behavioral – Observer

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



Frequency
of use

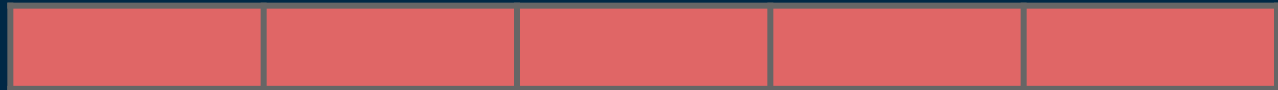
1

2

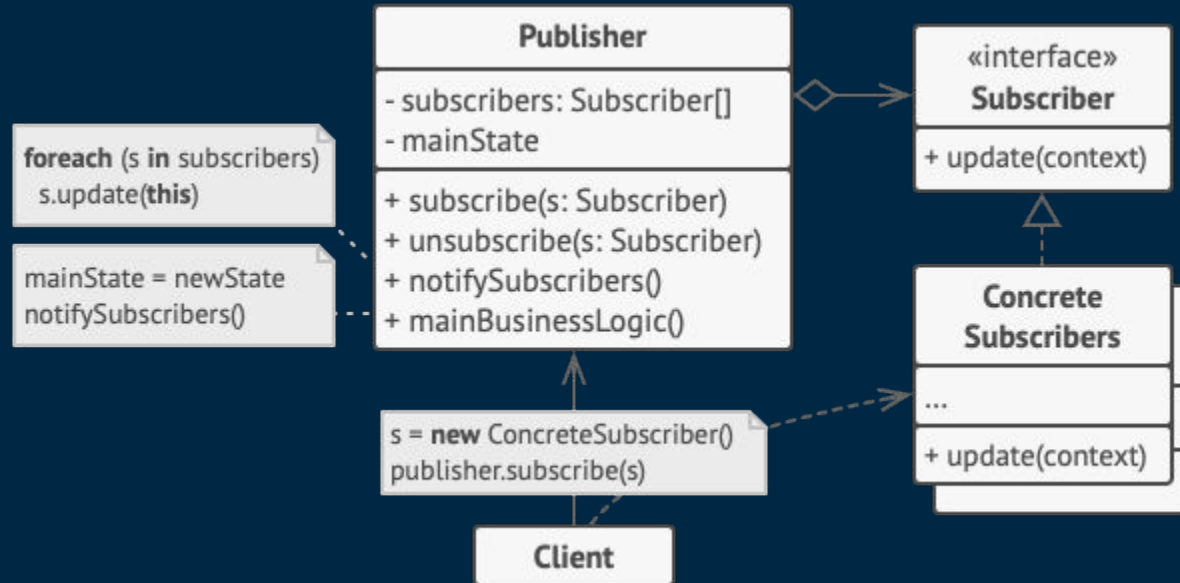
3

4

5



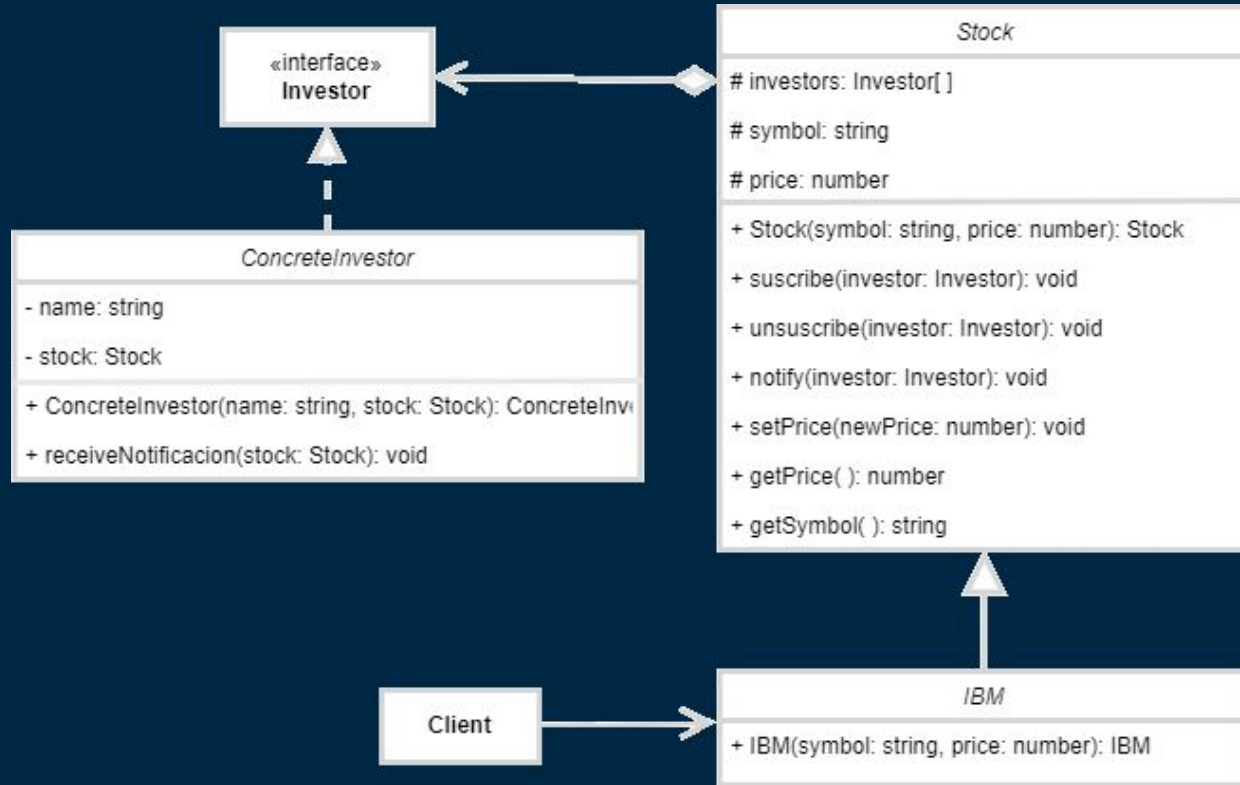
Behavioral – Observer – UML



Observer – Advantages and Disadvantages

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

Behavioral – Observer – UML – IBM Stock



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?

