



Design Patterns

Airam Rafael Luque León
Helena García Díaz

Team

Airam Rafael Luque León
alu0101335148



Helena García Díaz
alu0100829150



TABLE OF CONTENTS

01 **Introduction**

- a) What is a design pattern
- b) Design pattern beginnings
- c) What is its use?
- d) Classification

02 **Creational Patterns**

- a) Singleton
- b) Factory method

TABLE OF CONTENTS

03

Structural Patterns

a) Adapter

04

Behavioral Patterns

a) Command
b) Strategy

05

Conclusions

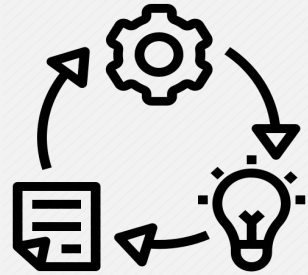
01

Introduction

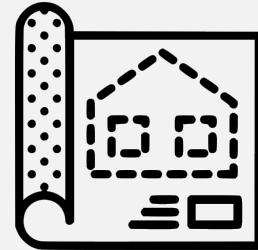


what is a design pattern?

They are solutions to problems that occur in software design.

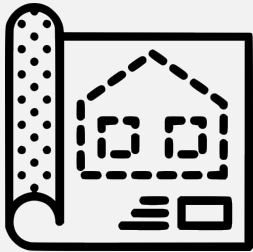


They are like pre-made blueprints that can be customized to solve a design problem.

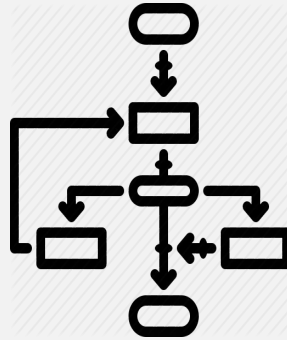


what is a design pattern?

Design
patterns

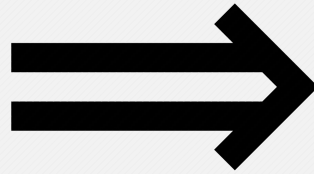
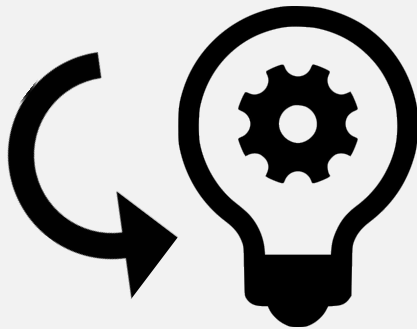


Algorithms



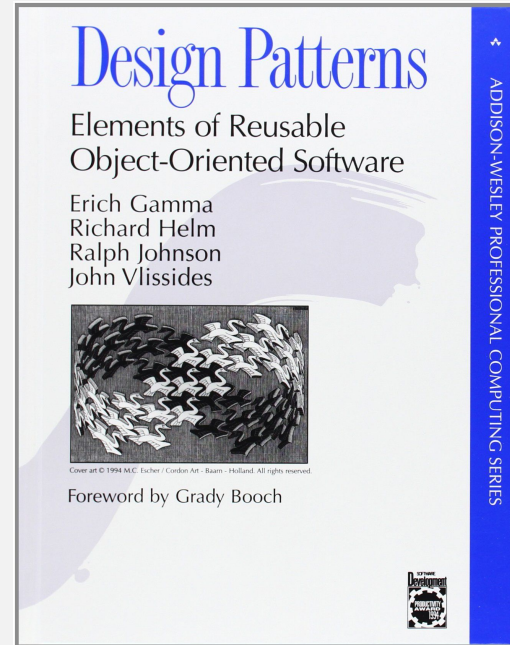
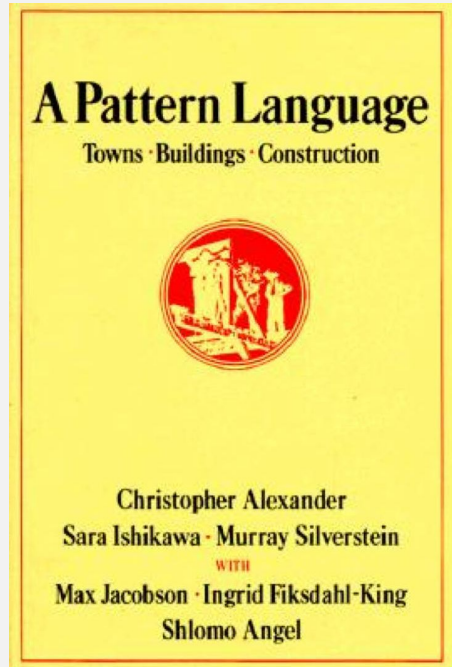
what is a design pattern?

They are common solutions to common problems in OOD.



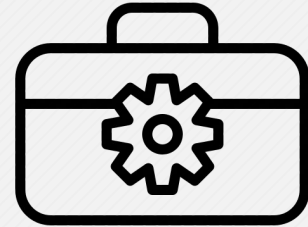
**NEW
PATTERN**

Design pattern beginnings



What is its use?

They are a toolkit of solutions to common problems in software design.



They define a common language that you can use to facilitate communication.



Classification

Idioms: More basic and lower level patterns

Architecture patterns: More universal and higher level patterns.



Design patterns differ in:

- Complexity
- Level of detail
- Scalability.

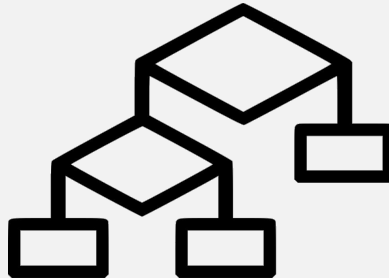
Classification

All patterns can be classified by their purpose.

**Creational
patterns**

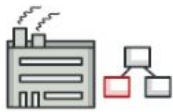


**Structural
patterns**

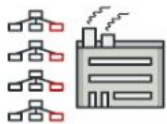


**Behavioral
patterns**

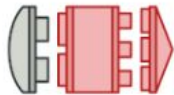




Factory Method



Abstract Factory



Adapter



Bridge



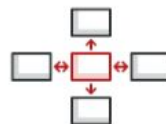
Chain of Responsibility



Command



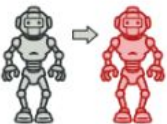
Iterator



Mediator



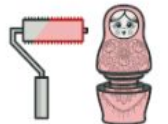
Builder



Prototype



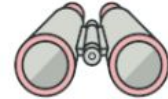
Composite



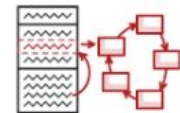
Decorator



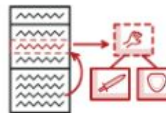
Memento



Observer



State



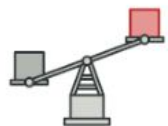
Strategy



Singleton



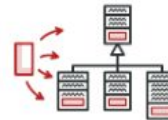
Facade



Flyweight



Template Method



Visitor

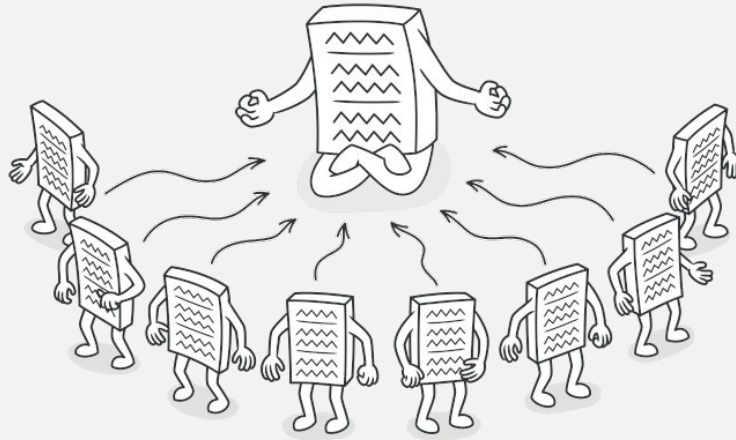
02

Creational Patterns



Singleton

Each class has a single instance, but we have to provide a global access point to that instance.



Problem

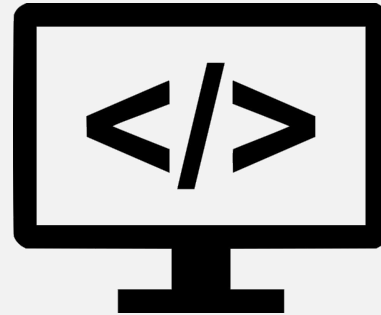
It violates the Single Responsibility Principle, resolving two problems:

- Ensures that a class has only one instance.
- Provides a global access point to that instance.

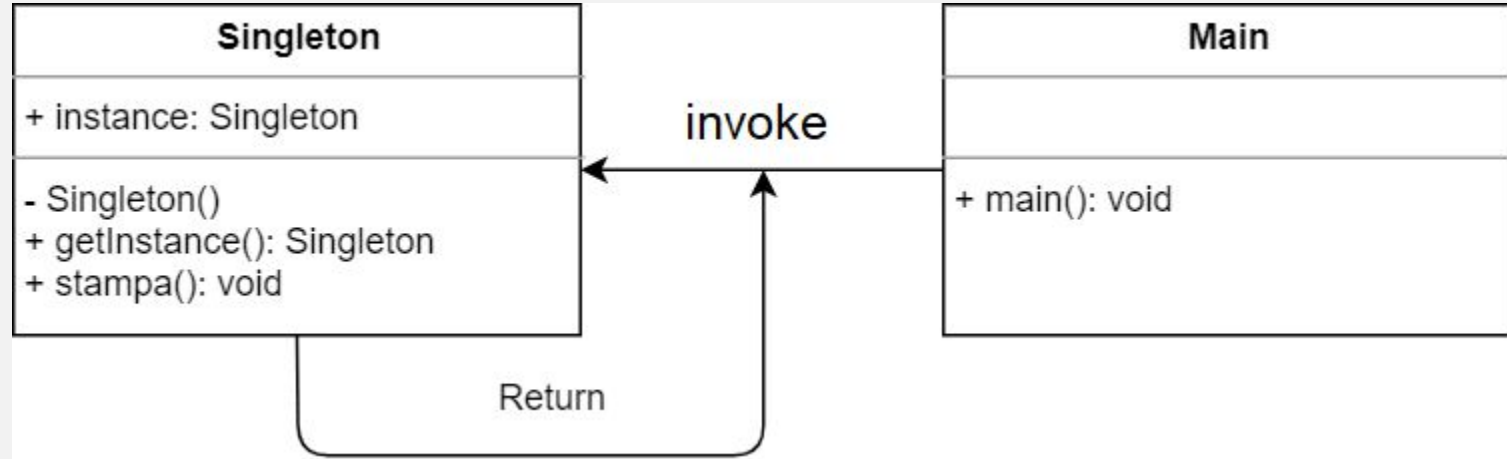


Solution

Let's go to the code...

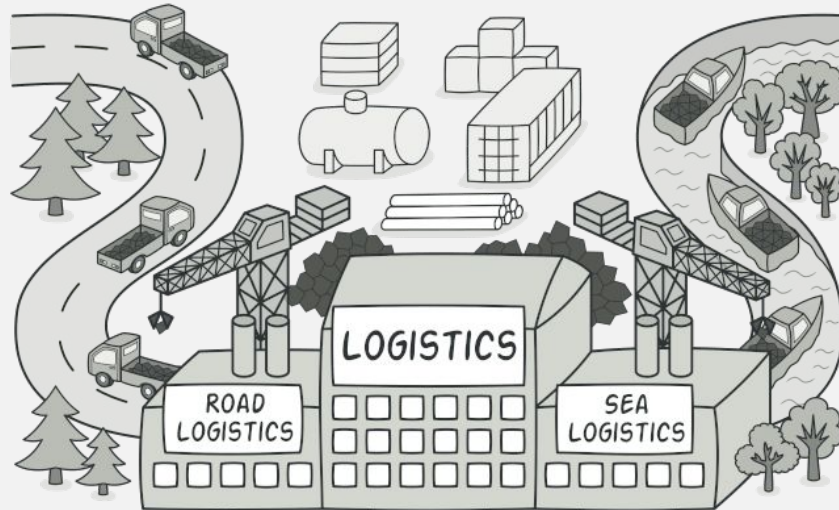


UML Diagram



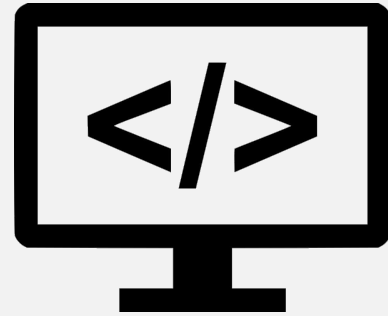
Factory Method

Provides an interface for creating objects in a superclass, while allowing subclasses to alter the type of objects that will be created.

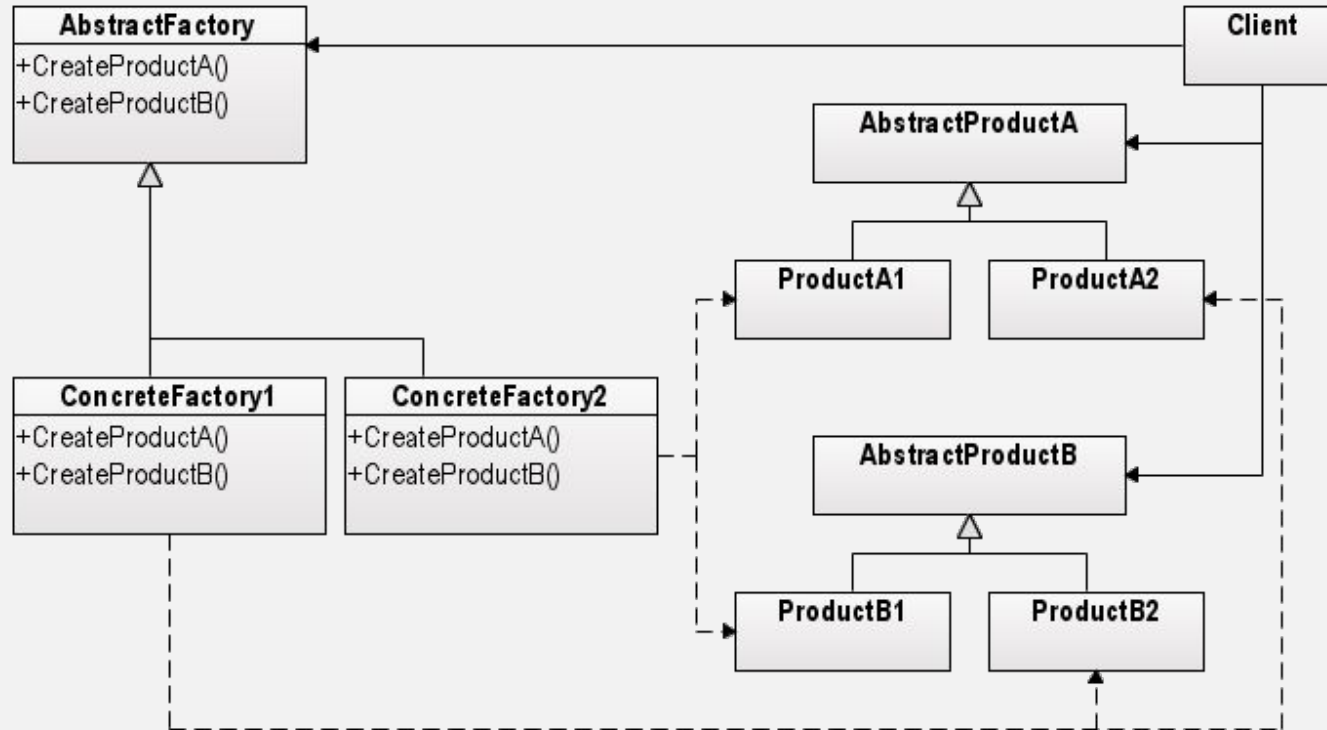


Everything is better with an example

Let's go to the code...



UML Diagram

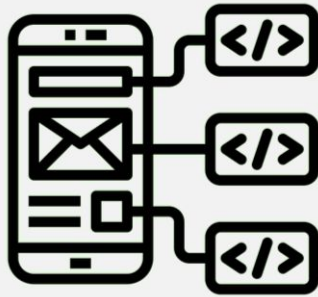


03

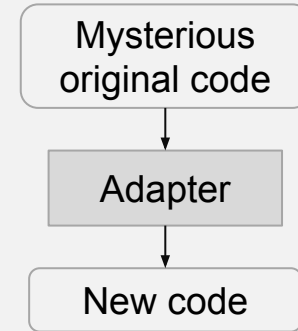
Structural Patterns



Structural patterns - Adapter



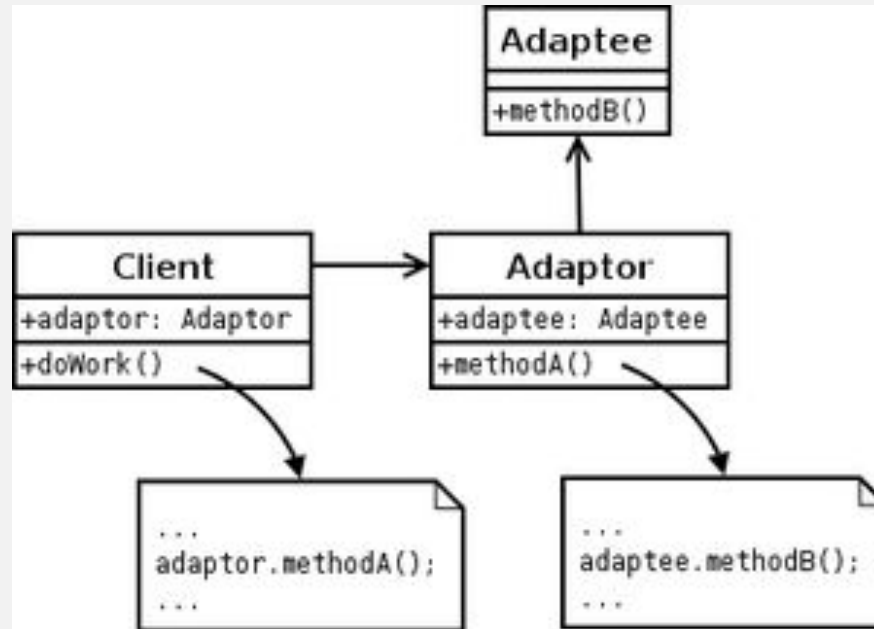
“Interface”



src/Structural_patterns

Structural patterns - Adapter

UML Diagram



04

Behavioral patterns



Behavioral patterns - Command

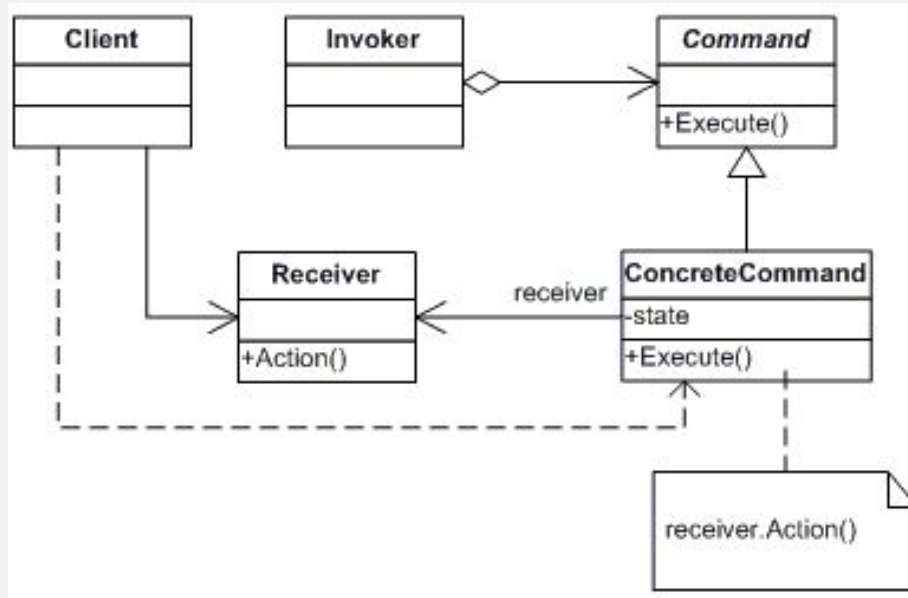


“Encapsulate”

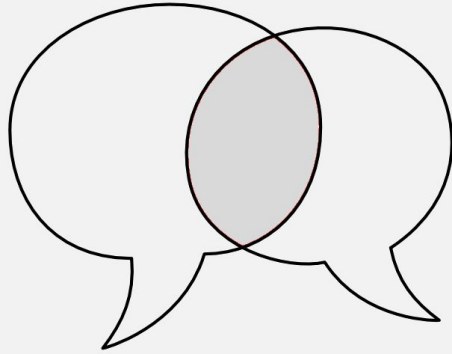
src/Behavioral_patterns/Command

Behavioral patterns - Command

UML Diagram



Behavioral patterns - Strategy

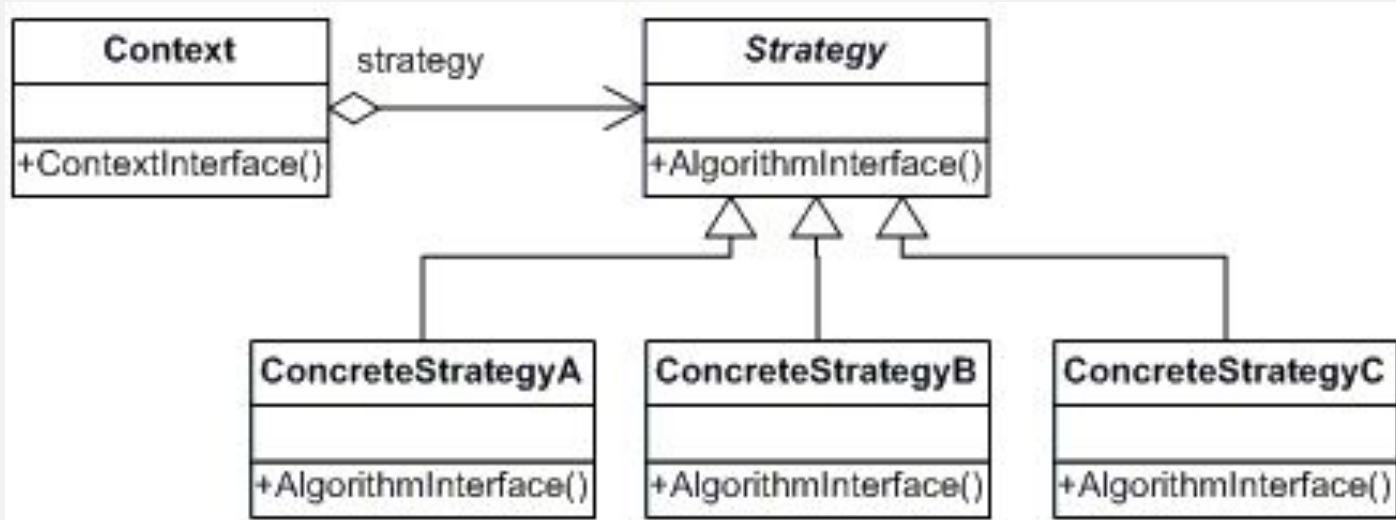


“Context”

src/Behavioral_patterns/Strategy

Behavioral patterns - Strategy

UML Diagram



05

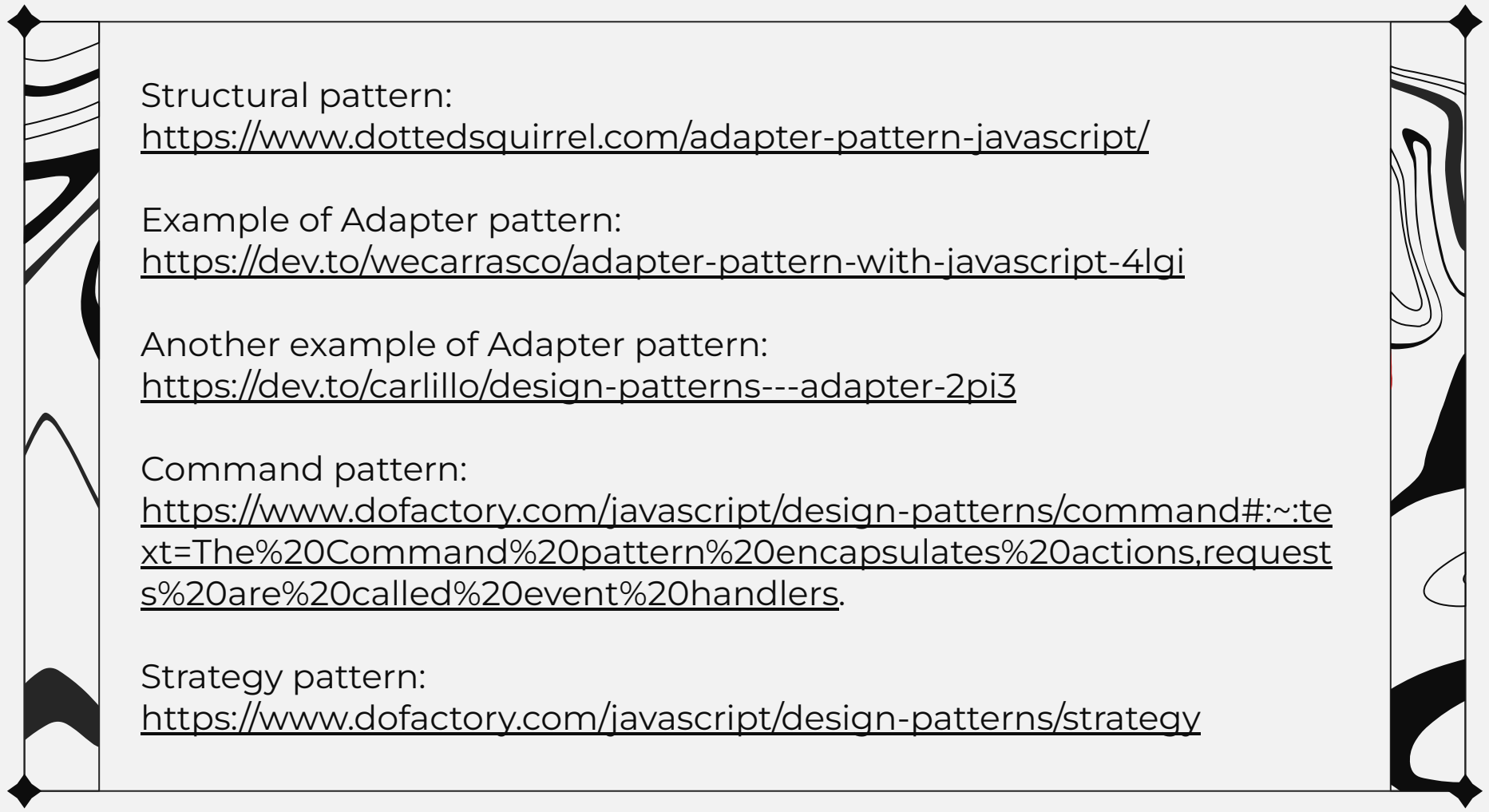
Conclusions



06

References





Structural pattern:

<https://www.dottedsqirrel.com/adapter-pattern-javascript/>

Example of Adapter pattern:

<https://dev.to/wecarrasco/adapter-pattern-with-javascript-4lqi>

Another example of Adapter pattern:

<https://dev.to/carlillo/design-patterns---adapter-2pi3>

Command pattern:

<https://www.dofactory.com/javascript/design-patterns/command#:~:text=The%20Command%20pattern%20encapsulates%20actions,request s%20are%20called%20event%20handlers.>

Strategy pattern:

<https://www.dofactory.com/javascript/design-patterns/strategy>



Some design patterns in Java:

<https://java-design-patterns.com/patterns/>

TypeScript 4 Design Patterns and Best Practices

<https://learning.oreilly.com/library/view/typescript-4-design/9781800563421/>

Code examples:

<https://github.com/PacktPublishing/TypeScript-4-Design-Patterns-and-Best-Practices>

Example of singleton:

<https://www.dofactory.com/javascript/design-patterns/singleton>

Example of Factory method:

<https://www.dofactory.com/javascript/design-patterns/factory-method>

THANKS!

Do you have any questions?

Airam Rafael Luque León

<airam.luque.10@ull.edu.es>

Helena García Díaz

<helena.garcia.diaz.20@ull.edu.es>

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**

