



# **Design Patterns**

Airam Rafael Luque León  
Helena García Díaz

# Team

**Airam Rafael Luque León**  
**airam.luque.10@ull.edu.es**



**Helena García Díaz**  
**helen.garcia.diaz.20@ull.edu.es**



# TABLE OF CONTENTS

## **01**   **Introduction**

- What is a design pattern
- Design pattern beginnings
- What is its use?
- Classification

## **02**   **Creational Patterns**

- Singleton
- Factory method

# TABLE OF CONTENTS

**03**

**Adapter**

- Adapter

**04**

**Behavioral Patterns**

- Command  
- Strategy

**05**

**Conclusions**

**06**

**References**

**01**

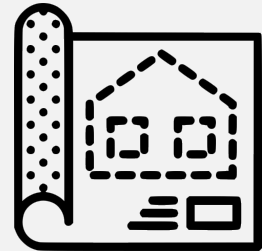
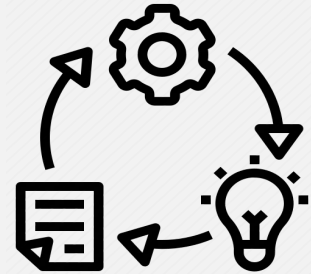
# **Introduction**



# what is a design pattern?

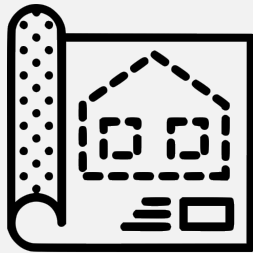
They are solutions to problems that occur in software design.

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



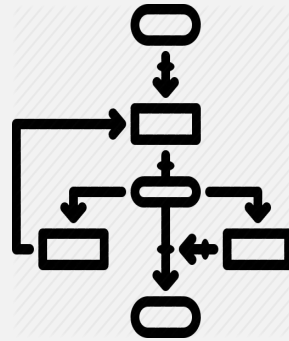
# what is a design pattern?

Design  
patterns



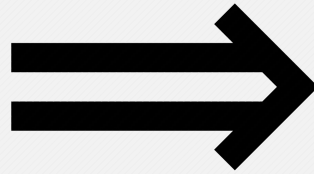
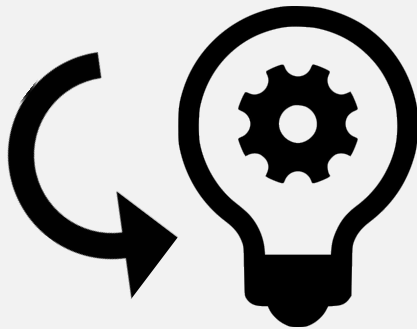
≠

Algorithms



# what is a design pattern?

They are common solutions to common problems in OOD.



**NEW  
PATTERN**



# **Design pattern beginnings**

A Pattern Language (1977):  
Christopher Alexander

Design Patterns (1994):  
Erich Gamma – Richard Helm  
Ralph Johnson – John Vlissides



# 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.

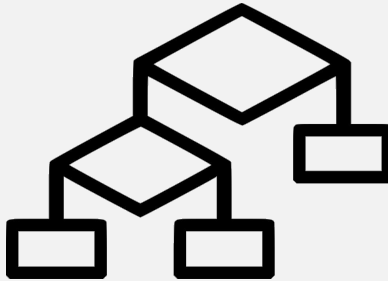
# Classification

All patterns can be classified by their purpose.

**Creational  
patterns**



**Structural  
patterns**



**Behavioral  
patterns**



# Some design patterns

[Link to the page](#)



**02**

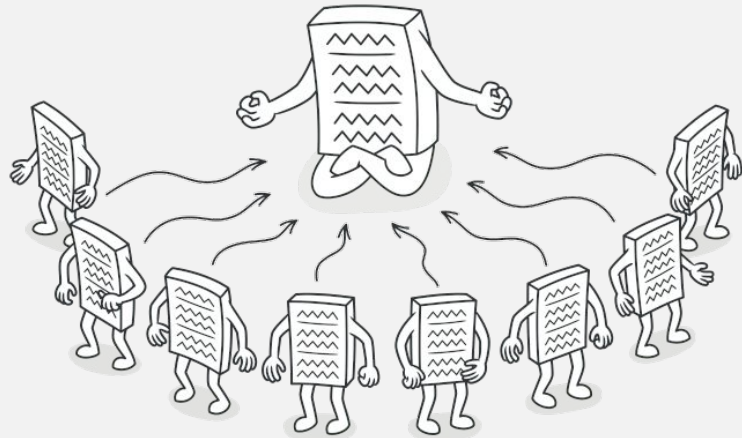
# **Creational Patterns**



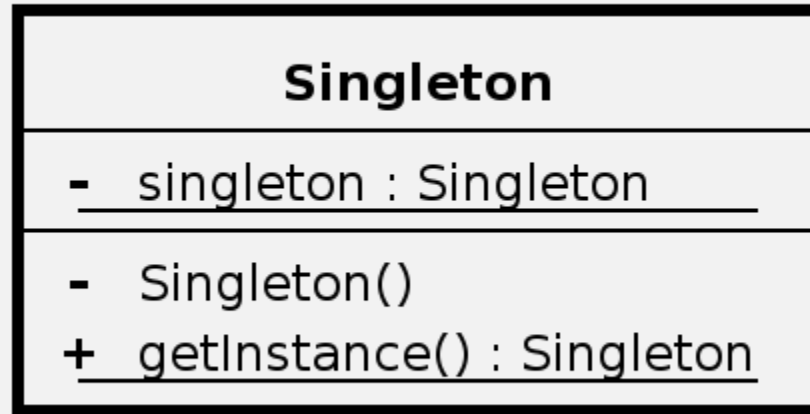
# Singleton

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

Popularity:



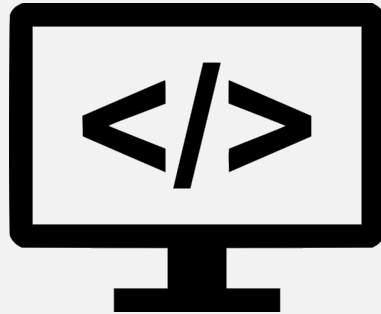
# UML Diagram





# Solution

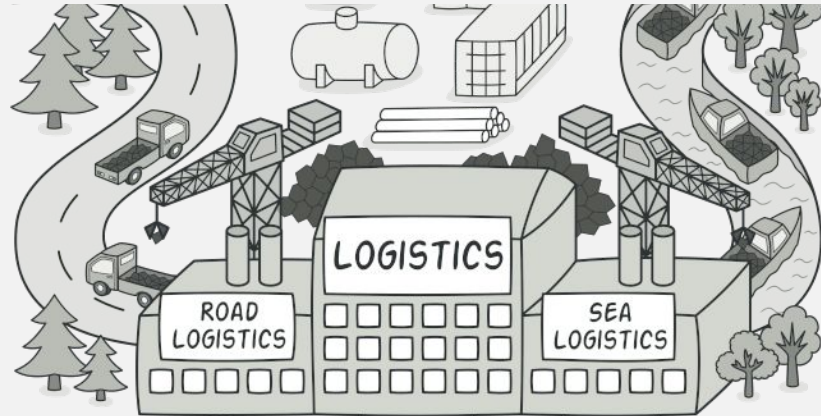
Let's go to the code...



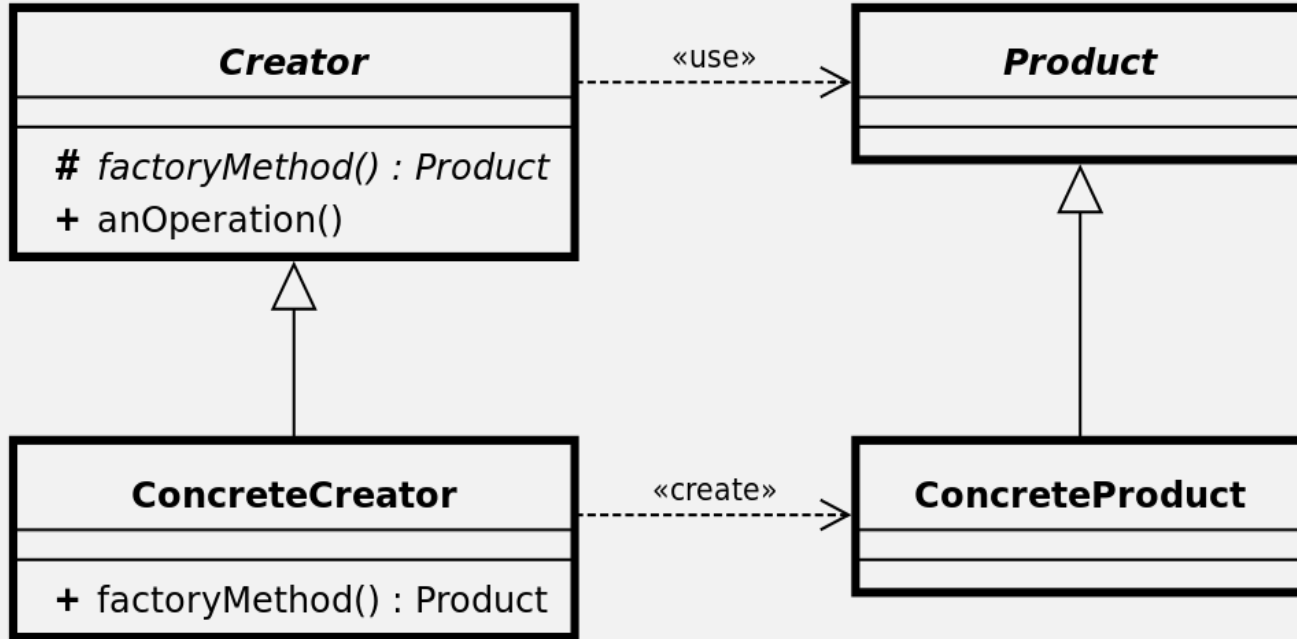
# Factory Method

Define an interface for creating an object but let the subclasses decide which class to instantiate.

Popularity:

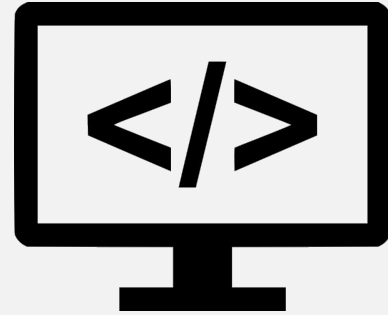


# UML Diagram



# Everything is better with an example

Let's go to the code...

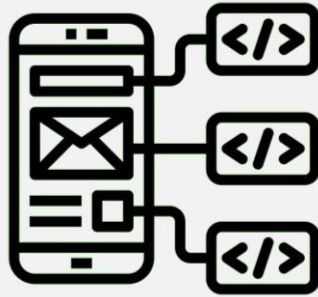


**03**

# **Structural Patterns**



# Structural patterns - Adapter



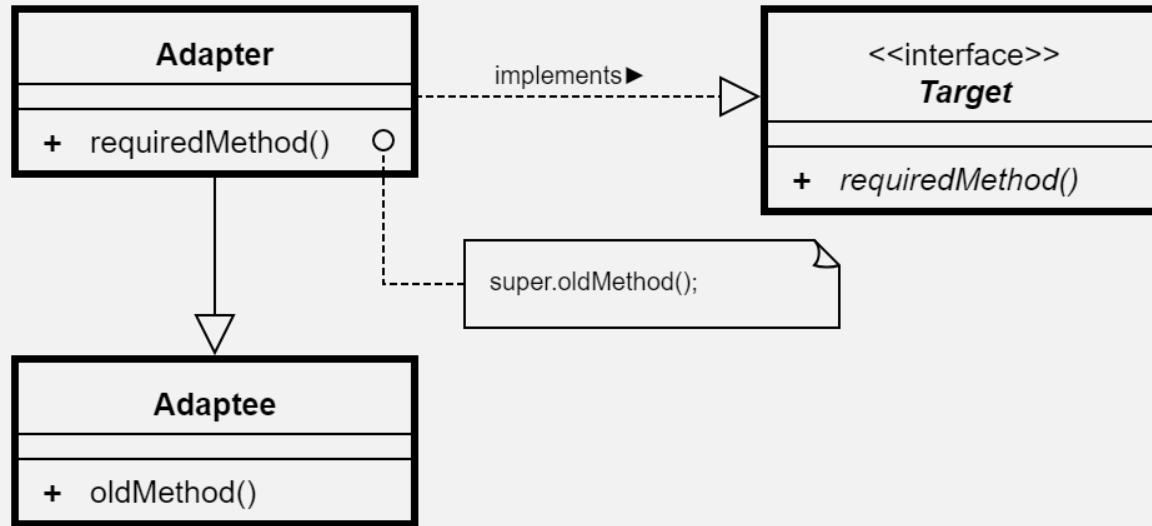
**“Interface”**

Popularity:

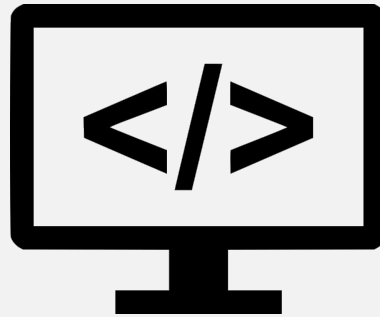


# Structural patterns - Adapter

## UML Diagram



# Structural patterns - Adapter



src/Structural\_patterns



**04**

# **Behavioral patterns**



# Behavioral patterns - Command



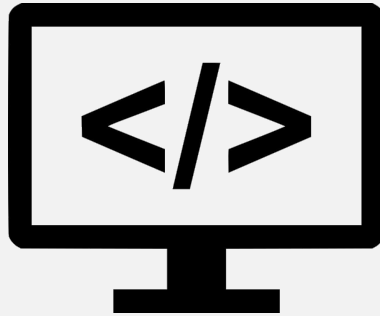
**“Encapsulate”**

Popularity:



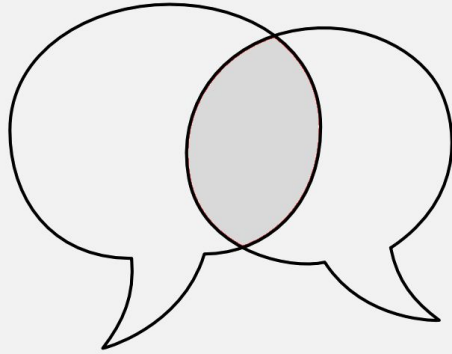
[illegible]

# Behavioral patterns - Command



**src/Behavioral\_patterns/Command**

# Behavioral patterns - Strategy

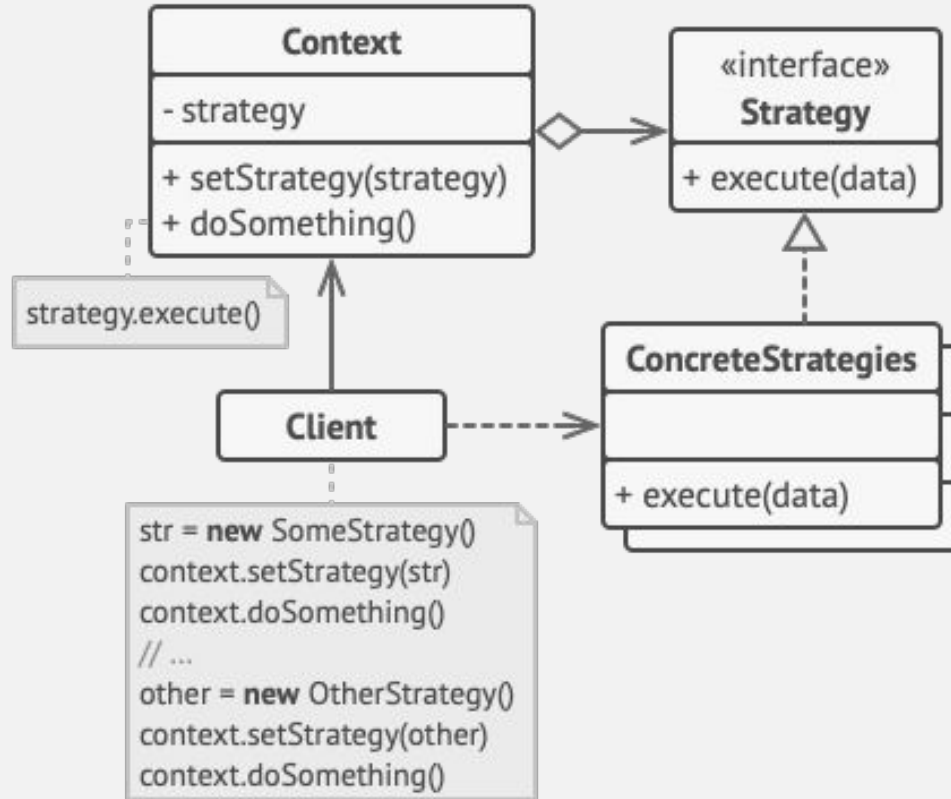


**“Context”**

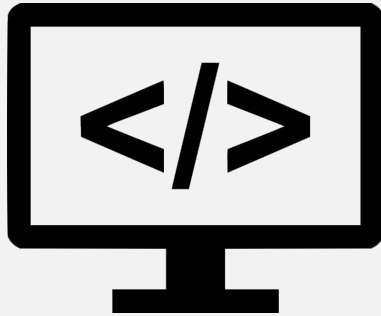
Popularity:



# UML Diagram



# Behavioral patterns - Strategy



**src/Behavioral\_patterns/Strategy**

**05**

# **Conclusions**





**06**

# **References**



Structural pattern:

<https://www.dottedsquirrel.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>

Factory Method UML Diagram

[https://commons.wikimedia.org/wiki/File:Factory\\_Method\\_UML\\_class\\_diagram.svg](https://commons.wikimedia.org/wiki/File:Factory_Method_UML_class_diagram.svg)

Singleton UML Diagram

<https://www.simogrima.com/php/php-design-pattern-singleton/>

# 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**

