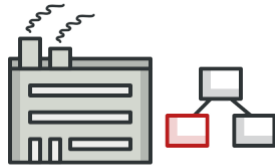




# The Catalog of Design Patterns

## Creational patterns

These patterns provide various object creation mechanisms, which increase flexibility and reuse of existing code.



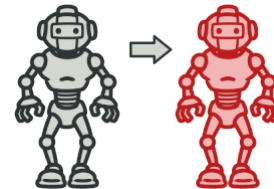
**Factory Method**



**Abstract Factory**



**Builder**



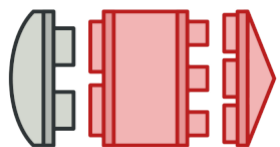
**Prototype**



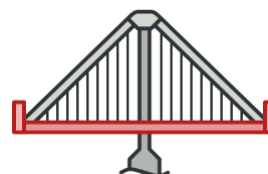
**Singleton**

## Structural patterns

These patterns explain how to assemble objects and classes into larger structures while keeping these structures flexible and efficient.



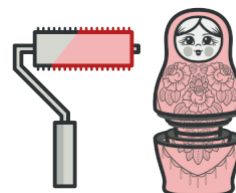
**Adapter**



**Bridge**



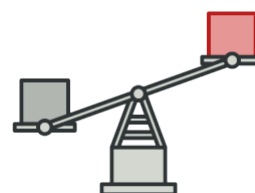
**Composite**



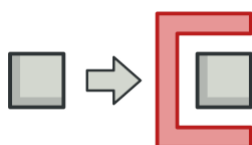
**Decorator**



**Facade**



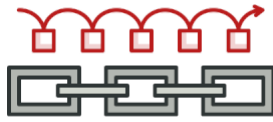
**Flyweight**



**Proxy**

## Behavioral patterns

These patterns are concerned with algorithms and the assignment of responsibilities between objects.



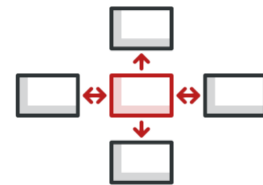
**Chain of Responsibility**



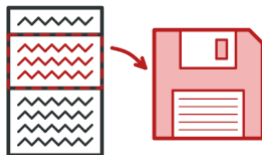
**Command**



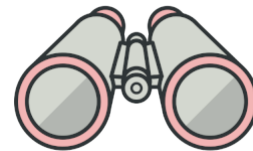
**Iterator**



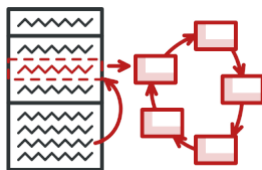
**Mediator**



**Memento**



**Observer**




**State**



**Strategy**

**Template Method****Visitor**[Home](#)[Refactoring](#)[Design Patterns](#)[Premium Content](#)[Forum](#)[Contact us](#)

© 2014-2019 Refactoring.Guru. All rights reserved.

 Illustrations by Dmitry Zhart

[Terms & Conditions](#)[Privacy Policy](#)