

A word cloud of various design patterns is contained within a black circle on the left side of the slide. The patterns listed include Command, Bridge, Iterator, Abstract, Chain, Decorator, Facade, Prototype, Mediator, Adapter, Interpreter, Template, Design\_patterns, Visitor, Flyweight, Strategy, Singleton, Memento, Builder, Composite, State, Factory, Observer, and Proxy. The text 'Design\_patterns' is the largest and most prominent in the cloud.

Command Bridge  
Iterator Abstract  
Chain Decorator  
Facade Prototype Mediator  
Adapter Interpreter Template  
Design\_patterns  
Visitor Flyweight Strategy  
Singleton Memento  
Builder Composite State  
Factory Observer  
factory Proxy

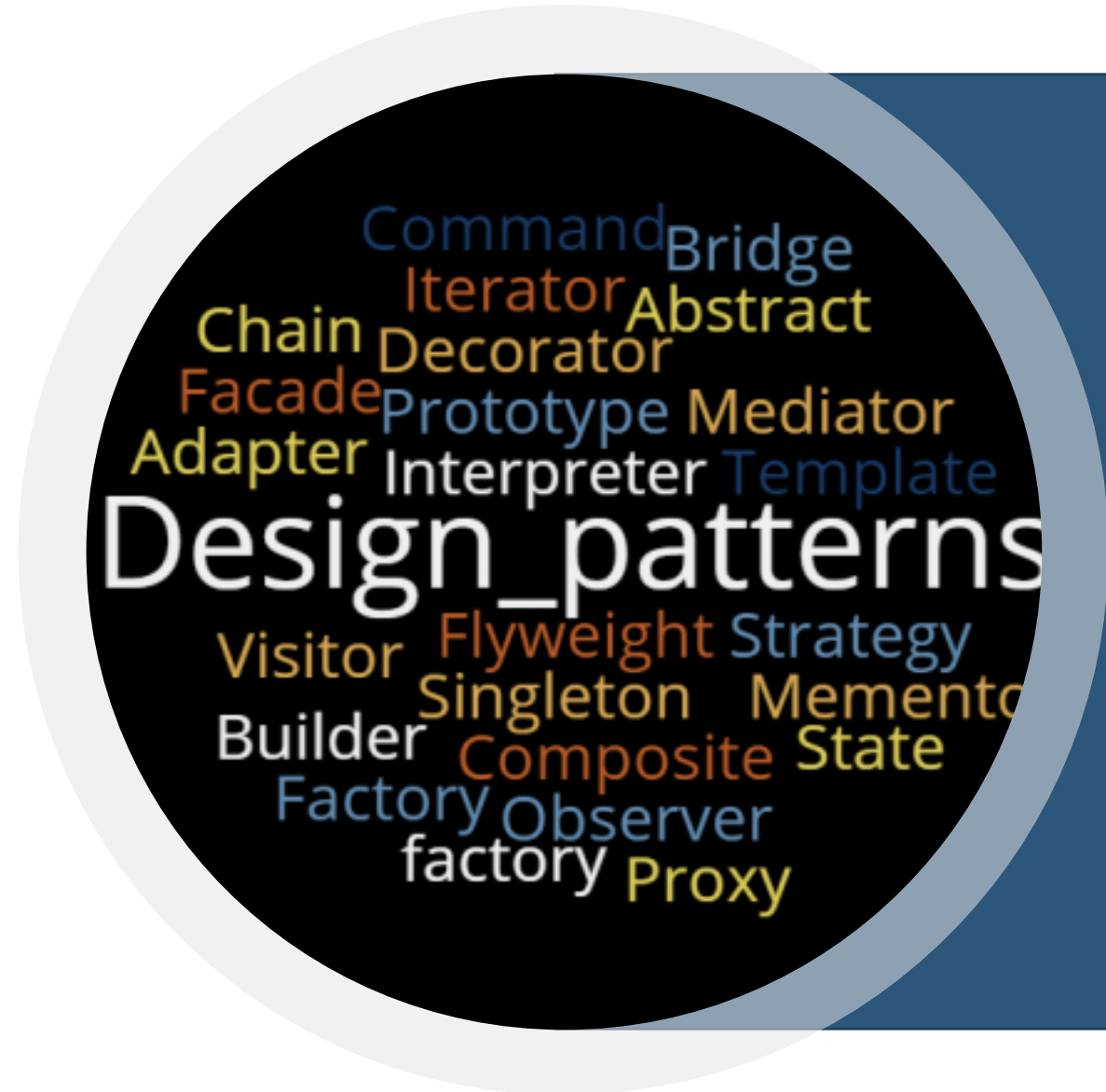
# DESIGN PATTERN OBSERVER

---

VERDEYME NATHAN  
BOURDIER LOANN

# SOMMAIRE

- I. Introduction
- II. Introduction générale sur les patterns
- III. Pattern Observer
- IV. Live Coding
- V. Principes SOLID
- VI. Limite du pattern
- VII. Lien avec les autres patterns
- VIII.QCM



# I. INTRODUCTION

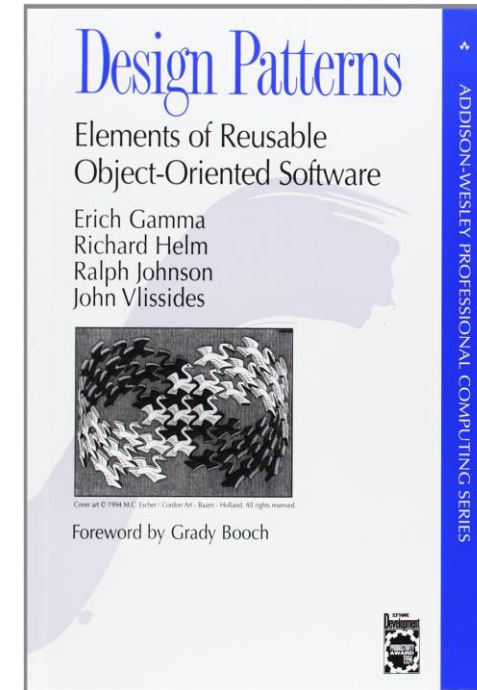
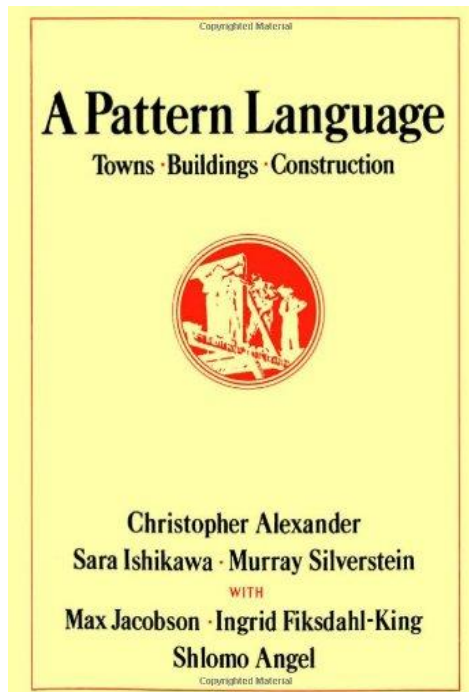
A word cloud of design patterns is centered on a black circle. The words are in various colors (blue, orange, yellow, white) and sizes. The largest word is "Design\_patterns" in white. Other words include "Command", "Bridge", "Iterator", "Abstract", "Chain", "Decorator", "Facade", "Prototype", "Mediator", "Adapter", "Interpreter", "Template", "Visitor", "Flyweight", "Strategy", "Singleton", "Memento", "Builder", "Composite", "State", "Factory", "Observer", "factory", and "Proxy".

Command Bridge  
Iterator Abstract  
Chain Decorator  
Facade Prototype Mediator  
Adapter Interpreter Template  
Design\_patterns  
Visitor Flyweight Strategy  
Singleton Memento  
Builder Composite State  
Factory Observer  
factory Proxy

# ORIGINE DES PATTERNS

A PATTERN LANGUAGE : TOWNS,  
BUILDINGS, CONSTRUCTION

1977



1994

DESIGNS PATTERNS : ELEMENTS OF  
REUSABLE OBJECT-ORIENTED SOFTWARE

# ECRIVAINS DU LIVRE « A PATTERN LANGUAGE : TOWNS, BUILDINGS, CONSTRUCTION »



CHRISTOPHER ALEXANDER



SARA ISHIKAWA



MURRAY SILVERSTEIN

# ECRIVAINS DU LIVRE : « DESIGNS PATTERNS : ELEMENTS OF REUSABLE OBJECT-ORIENTED SOFTWARE



ERICH GAMMA



JOHN VLISSIDES



RALPH JOHNSON



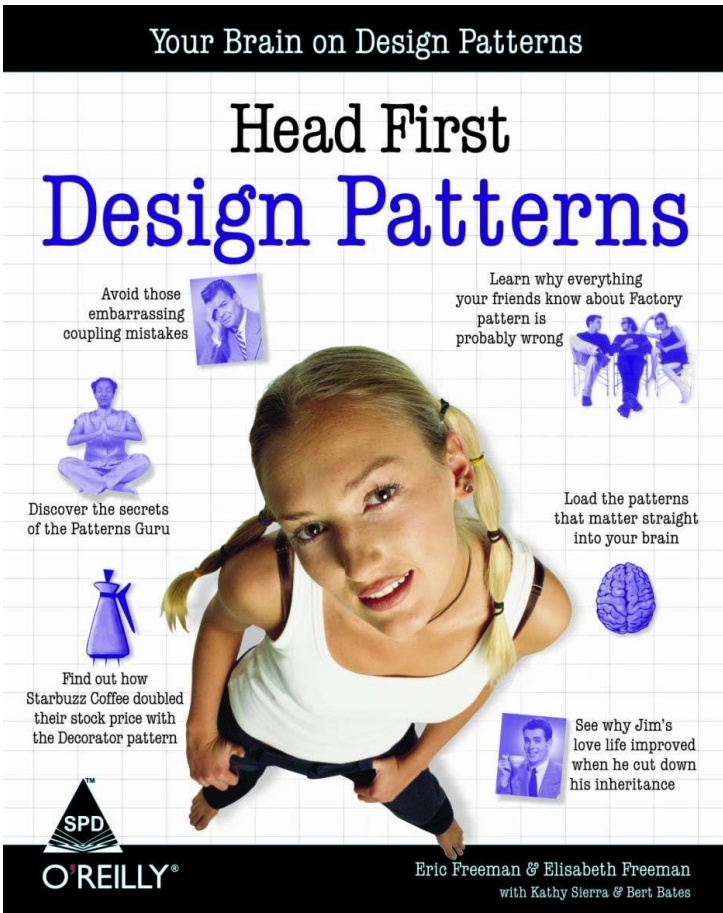
RICHARD HELM

Apport du livre du  
GoF

Solutions simples aux  
problèmes de conception les  
plus courants

23 modèles permettant de  
créer des design patterns  
flexibles, élégants et  
réutilisables





- Complément à la bible
- Parution : 2004
- Apport : Bénéficier des meilleurs pratiques ainsi que de l'expérience des autres et donne informations sur les différents design pattern.



## II. INTRODUCTION GÉNÉRALE SUR LES PATTERNS



A word cloud of design patterns is centered on a black circle. The patterns are listed in various colors (blue, orange, yellow, white) and sizes. The central and largest text is "Design\_patterns". Other patterns include Command, Bridge, Iterator, Abstract, Chain, Decorator, Facade, Prototype, Mediator, Adapter, Interpreter, Template, Visitor, Flyweight, Strategy, Singleton, Memento, Builder, Composite, State, Factory, Observer, factory, and Proxy.

Command Bridge  
Iterator Abstract  
Chain Decorator  
Facade Prototype Mediator  
Adapter Interpreter Template  
**Design\_patterns**  
Visitor Flyweight Strategy  
Singleton Memento  
Builder Composite State  
Factory Observer  
factory Proxy

# QU'EST CE QU'UN DESIGN PATTERN ?



---

Méthode de  
développement  
logiciel

---

Optimiser, clarifier le  
code en répondant aux  
problématiques

---

Trois familles :

**Créateurs**

instanciation et configuration des classes et objets.

**Structuraux**

Organiser les classes dans une structures plus large.

**Comportementaux**

organiser les objets pour leur collaboration et explique  
le fonctionnement des algos.

# III. PATTERN OBSERVER



A word cloud of design patterns is centered on a black circle. The patterns listed include: Command, Bridge, Iterator, Abstract, Chain, Decorator, Facade, Prototype, Mediator, Adapter, Interpreter, Template, Design\_patterns, Visitor, Flyweight, Strategy, Singleton, Memento, Builder, Composite, State, Factory, Observer, factory, and Proxy. The text is in various colors (blue, orange, yellow, white) and sizes, with 'Design\_patterns' being the largest and most prominent.

# A QUOI IL SERT ?

Issu de la famille des patterns comportementaux.

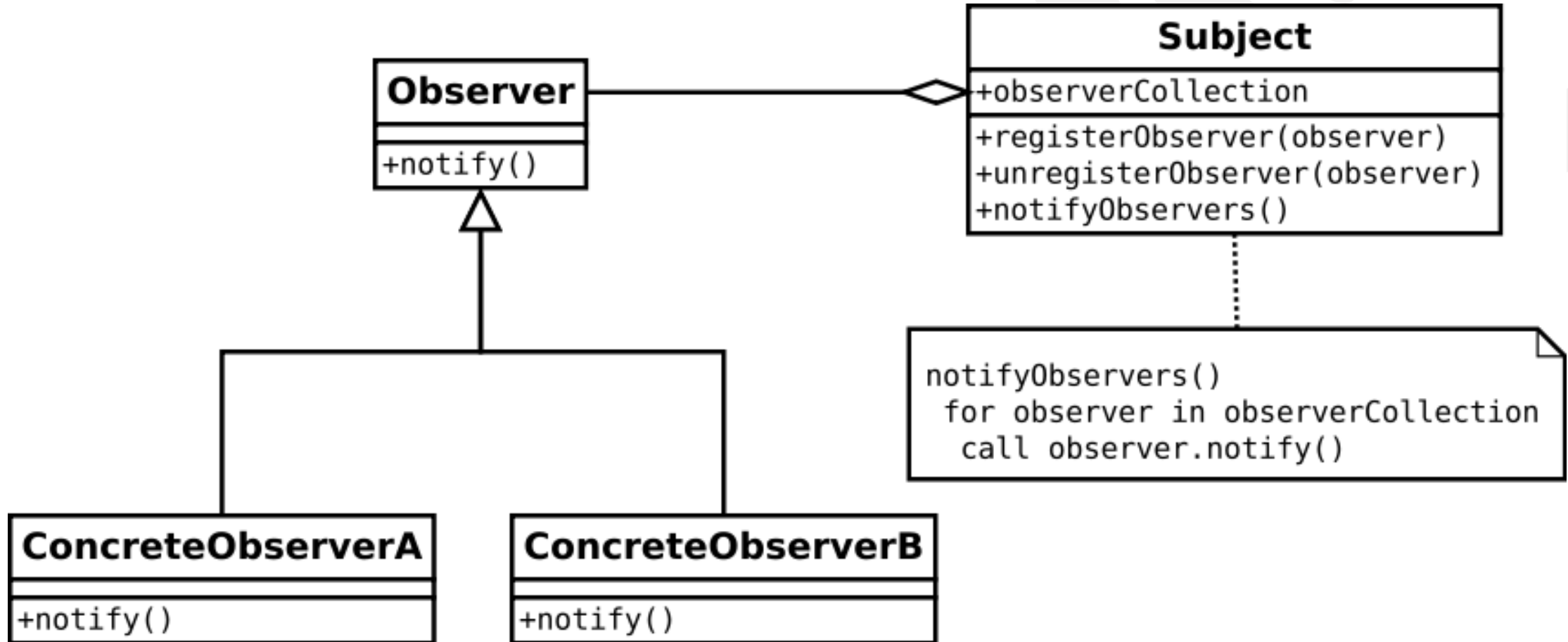
Définit un rapport de dépendance entre  $\geq 2$  objets

Communiquer les modifications de manière optimisé.

2 acteurs :

- **Sujet**: objet qui est observé
- **Observateurs** : informé des modifications du sujet

# DIAGRAMME GÉNÉRAL



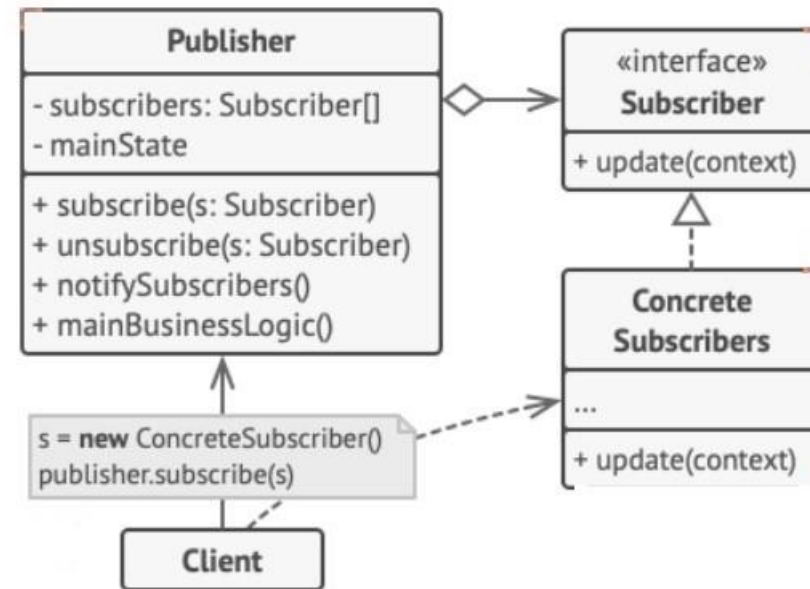
# UTILISATION

Chargé de la  
communication entre  
toutes les classes

Relation One To Many

Informations du  
changement aux autres  
classes de manière  
simultané pour tout ceux  
en dépendance avec lui.

# EXEMPLE DE PROBLÉMATIQUE





# SOLUTION

## Interface Observable

- Notifie de SON changement d'état à une collection d'observateur liée.

## Interface Observateur

- Récupère et met à jour le changement notifié par l'observable.

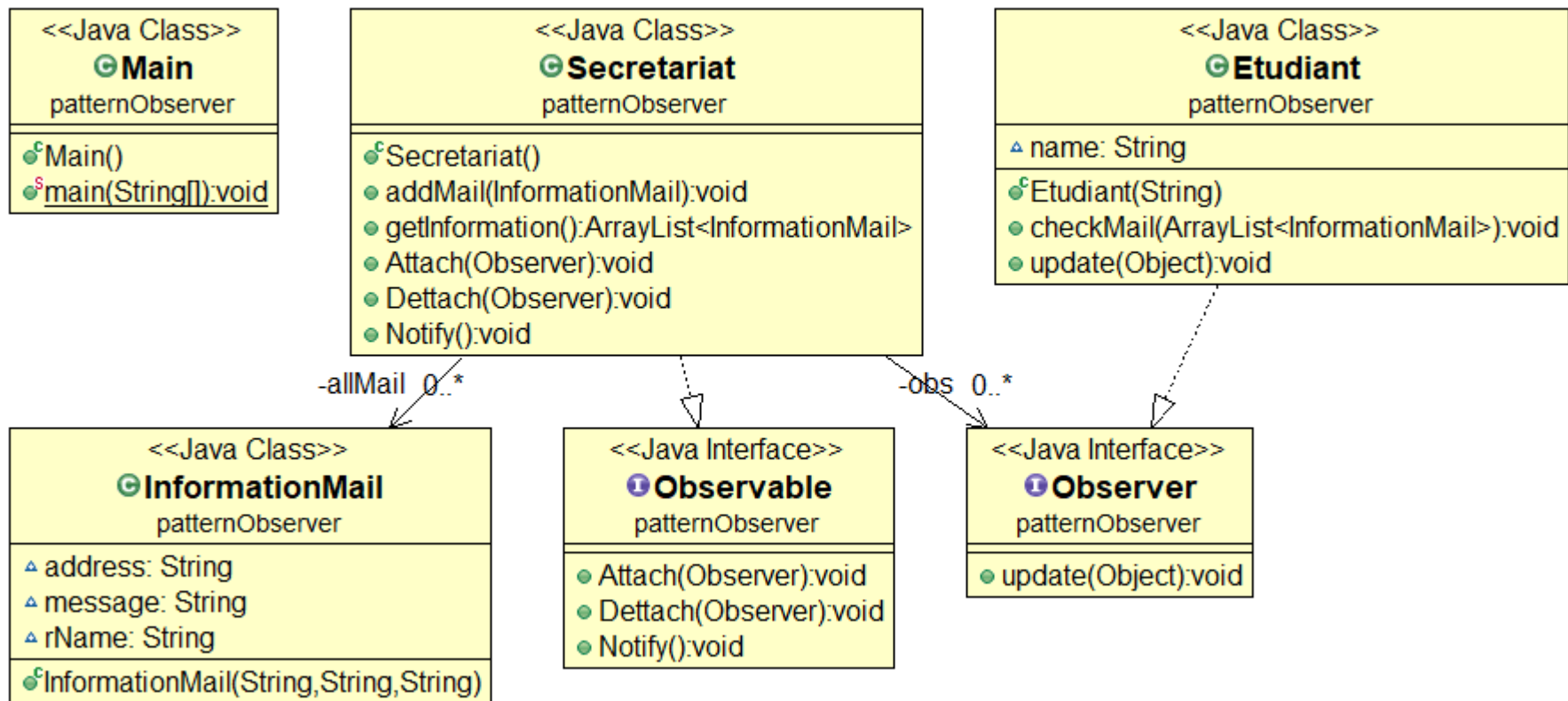
# IV. LIVE CODING



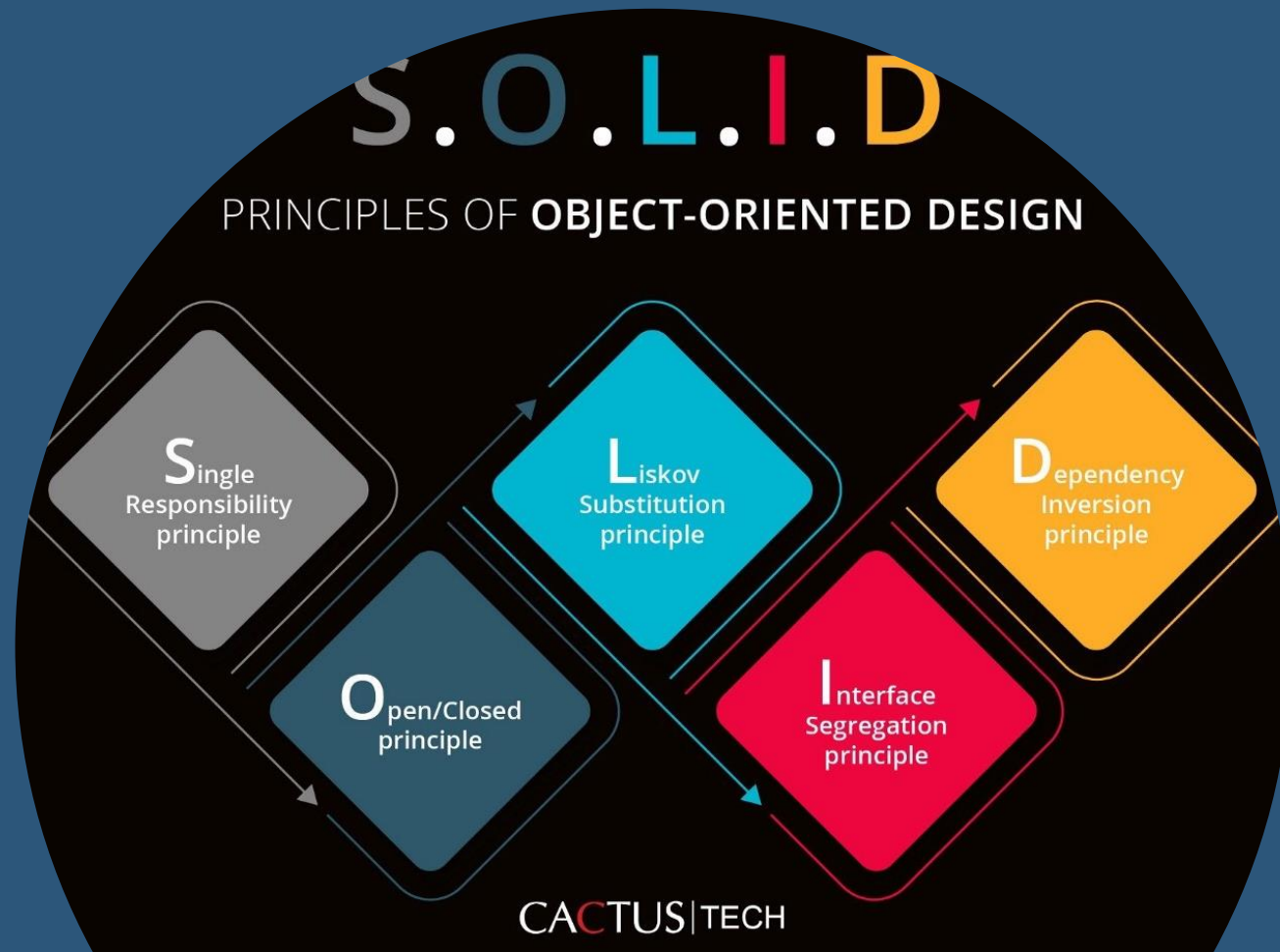
A word cloud of design patterns is centered on a black circle. The words are in various colors (blue, orange, yellow, white) and sizes. The largest word is "Design\_patterns" in white. Other words include "Command", "Bridge", "Iterator", "Abstract", "Chain", "Decorator", "Facade", "Prototype", "Mediator", "Adapter", "Interpreter", "Template", "Visitor", "Flyweight", "Strategy", "Singleton", "Memento", "Builder", "Composite", "State", "Factory", "Observer", "factory", and "Proxy".

Design\_patterns



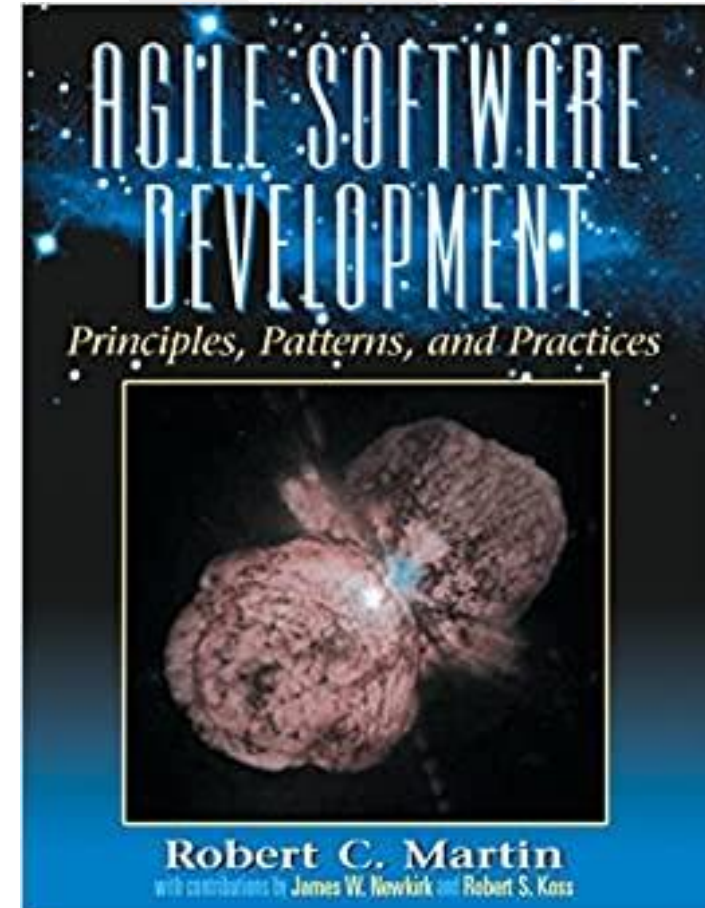


# V. PRINCIPE SOLID



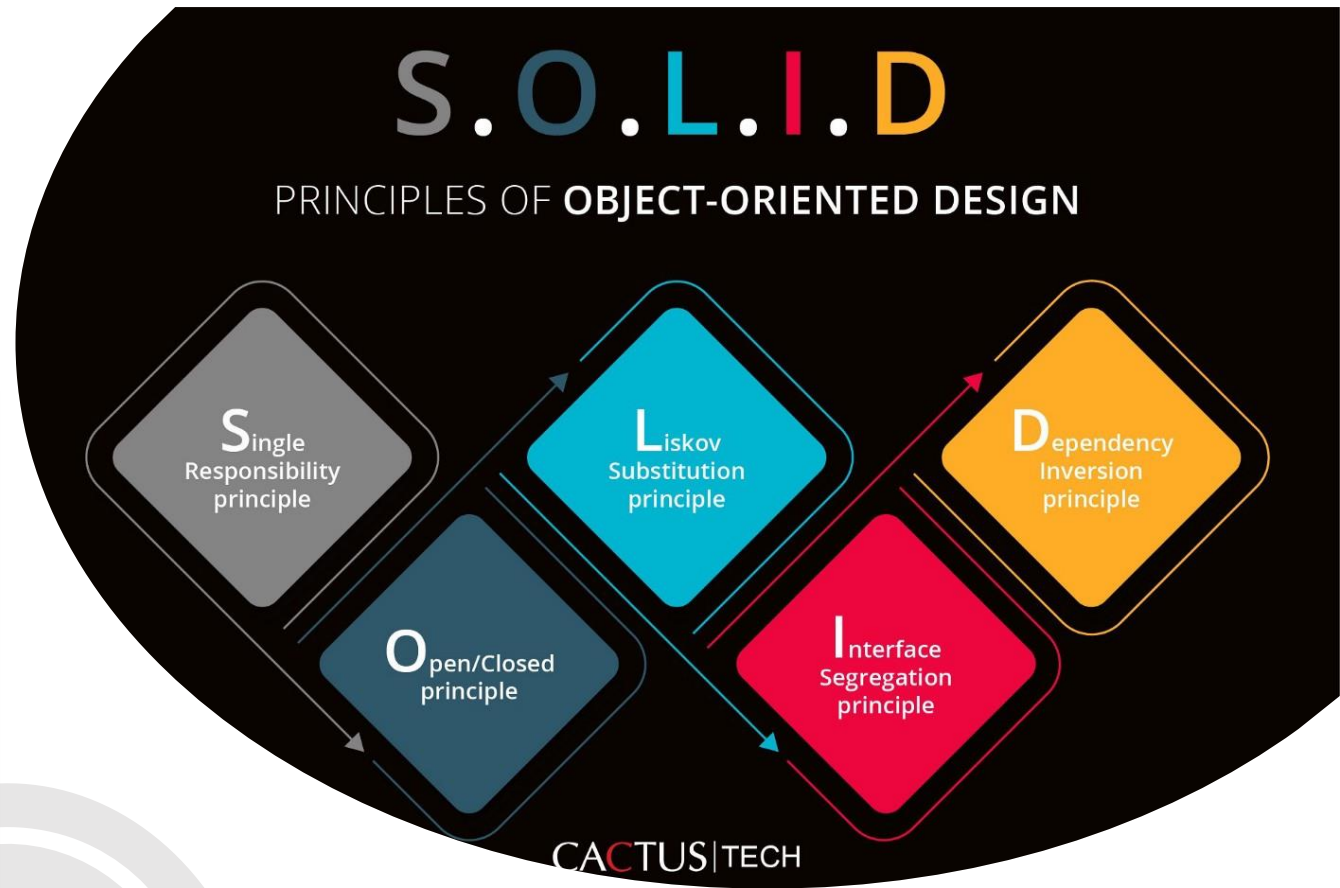
# QU'EST CE QUE C'EST

- Dans le livre « Agile Software Development, Principles, Patterns and Practices », Robert C. Martin a condensé, en 2002, cinq principes fondamentaux de conception, répondant aux problèmes liés à l'évolution du code, sous l'acronyme SOLID.



# SINGLE RESPONSABILITY PRINCIPLE

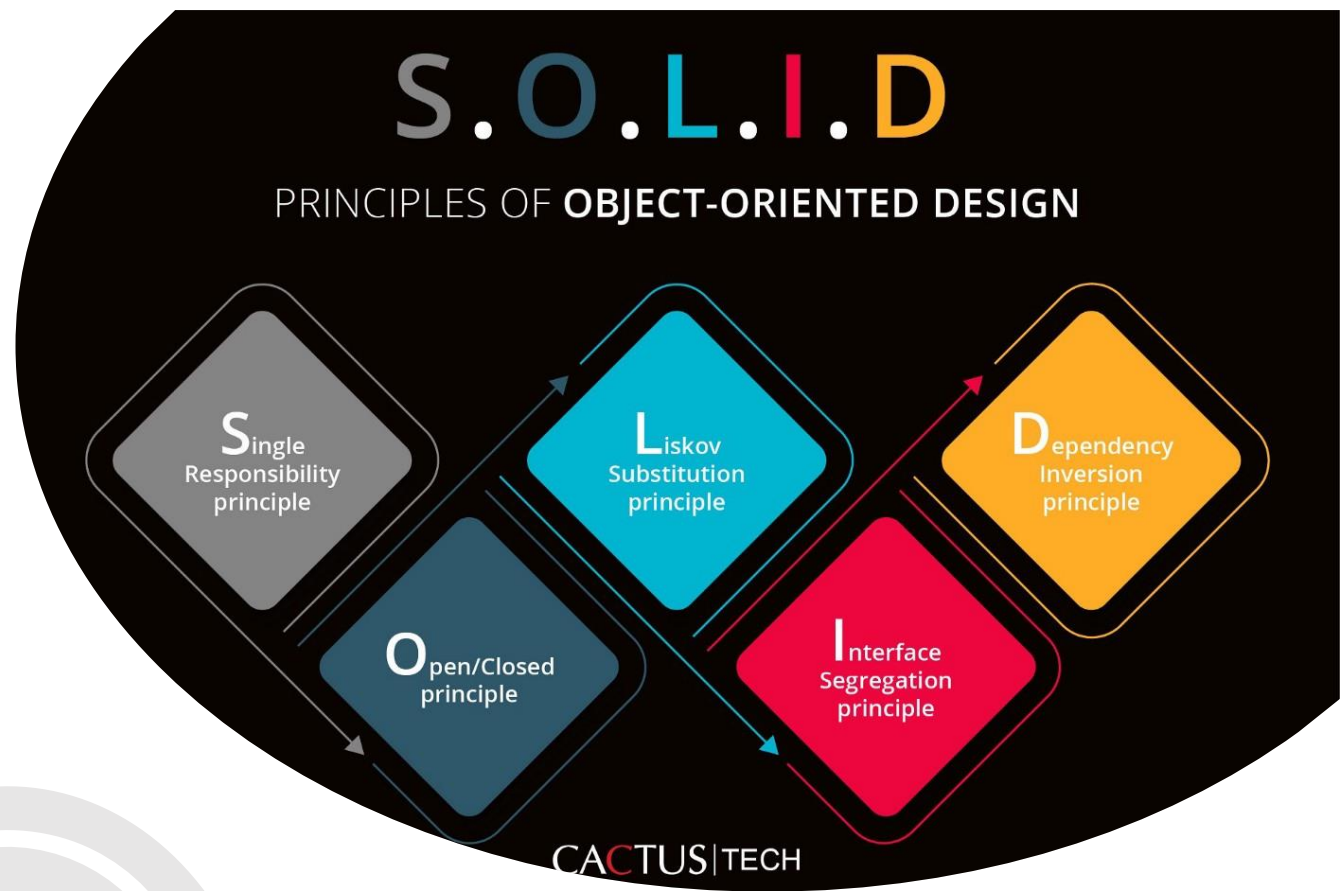
Une seule responsabilité





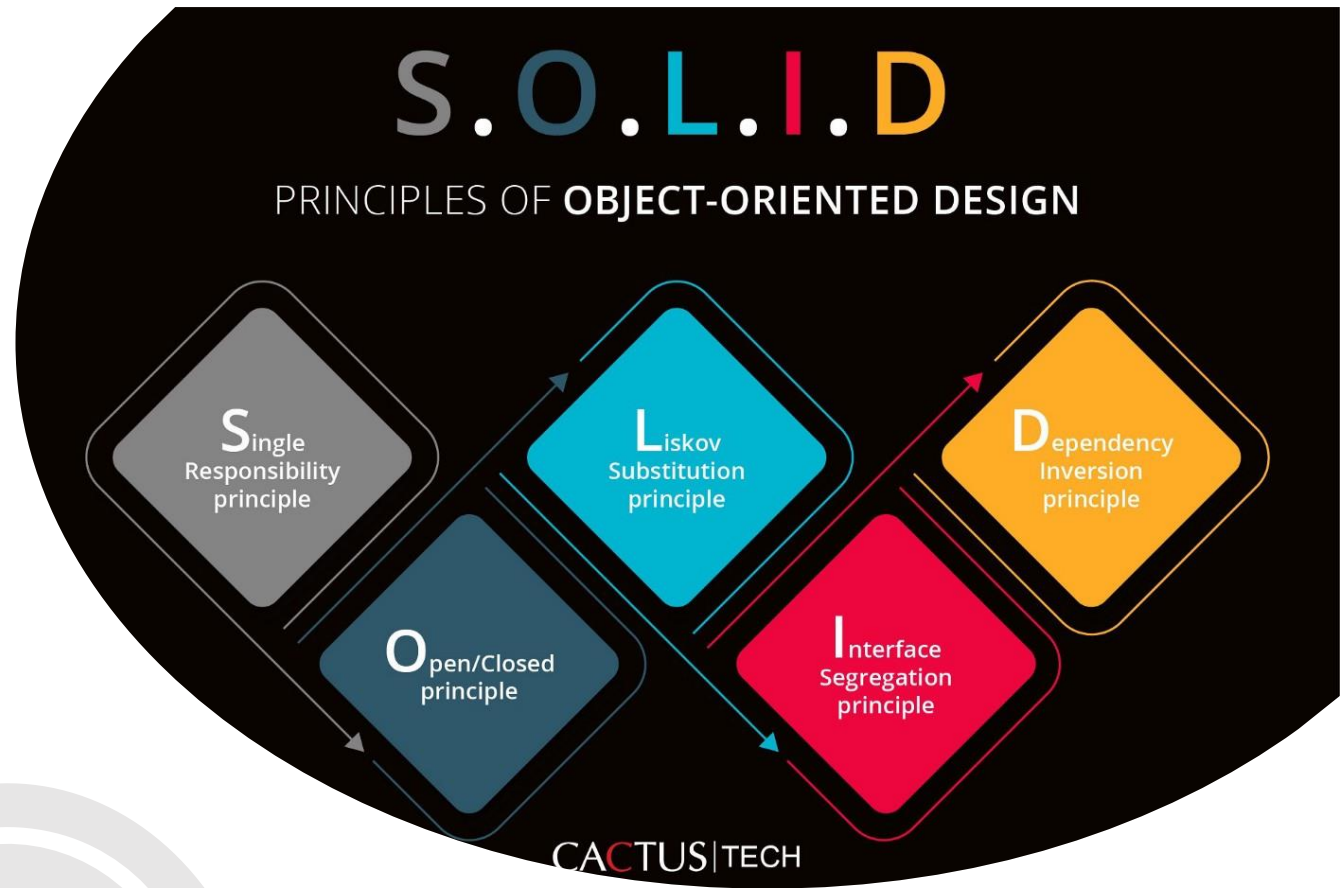
# OPEN / CLOSE PRINCIPLE

Classe / méthode ou module doit pouvoir être étendu / supporter différentes implémentations SANS devoir être par la suite modifié



# LSKOV SUBSTITUTION PRINCIPLE

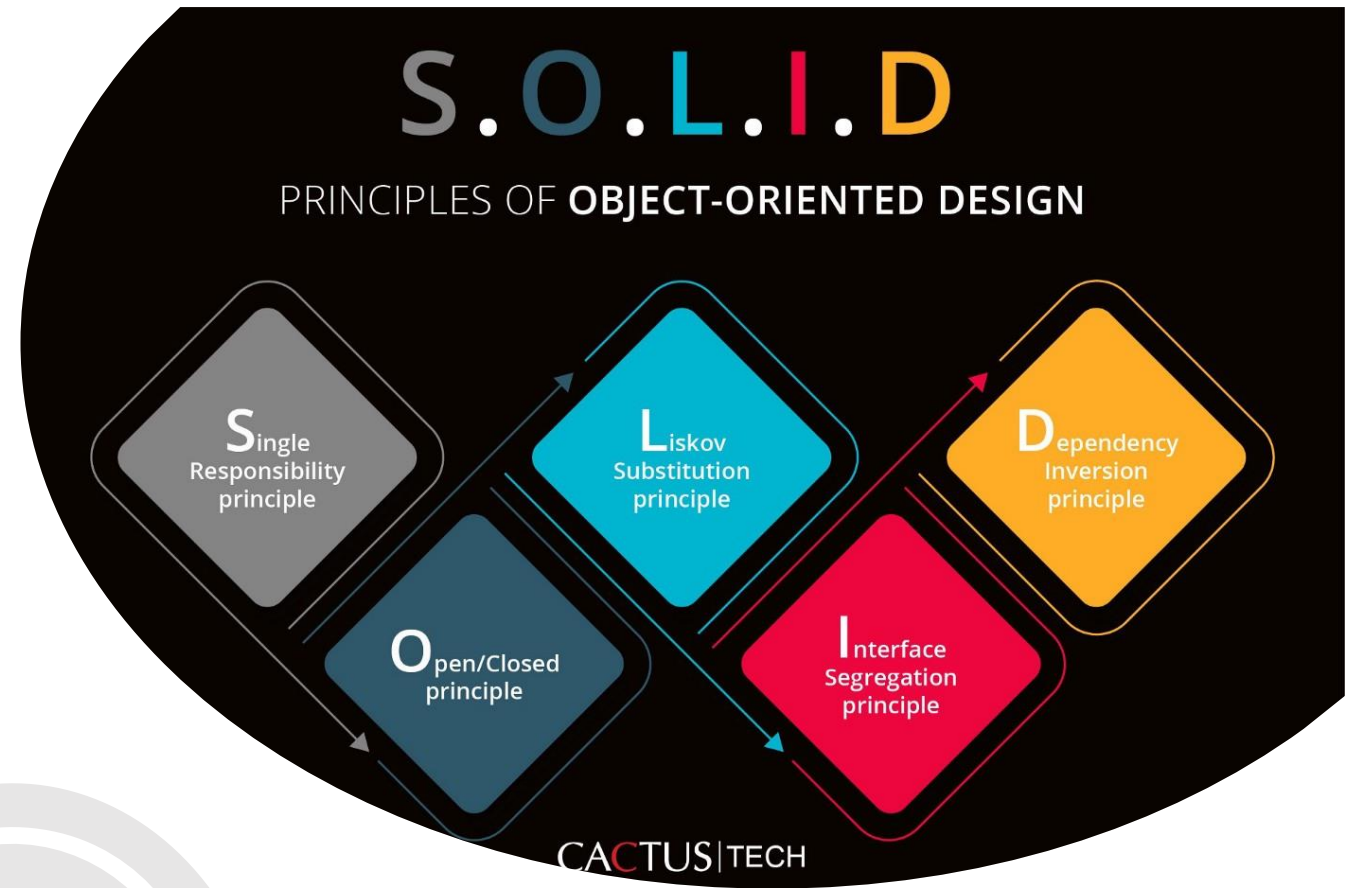
Sous classes doivent pouvoir être substituées à leur classe de base sans altérer le comportement de ses utilisateurs.



# INTERFACE SEGREGATION PRINCIPLE

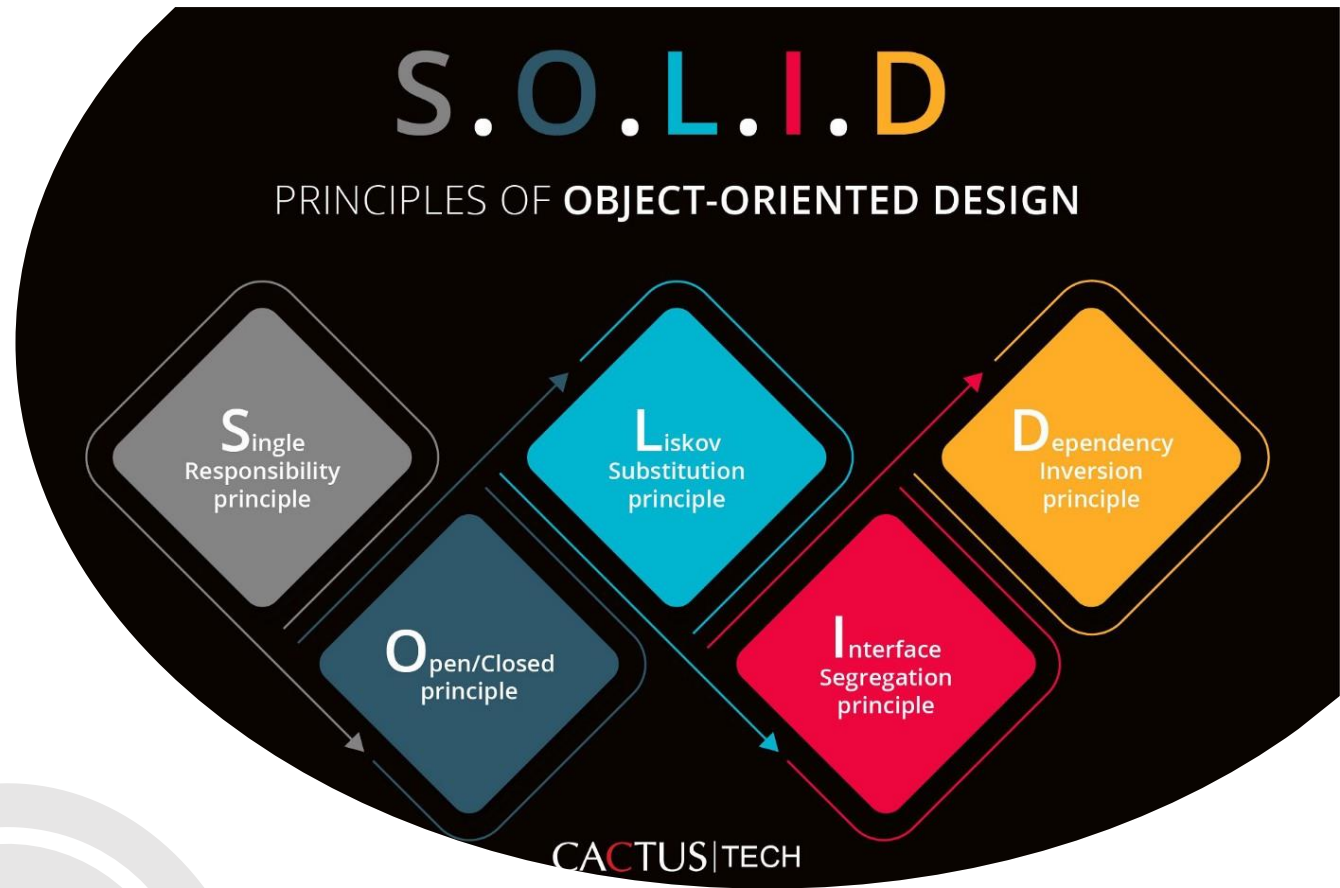
Eviter d'avoir des interfaces aux multiples responsabilités

Les redécouper en multiples interfaces avec une seule responsabilité



# DEPENDENCY INVERSION PRINCIPLE

Les modules de hauts niveaux ne doivent en aucun cas dépendre des modules de bas niveau.



# VI. LIMITES DU PATTERN



A word cloud of design patterns is centered on a black circle. The patterns are listed in various colors (blue, orange, yellow, white) and sizes. The central and largest text is "Design\_patterns". Other patterns include Command, Bridge, Iterator, Abstract, Chain, Decorator, Facade, Prototype, Mediator, Adapter, Interpreter, Template, Visitor, Flyweight, Strategy, Singleton, Memento, Builder, Composite, State, Factory, Observer, factory, and Proxy.

Command Bridge  
Iterator Abstract  
Chain Decorator  
Facade Prototype Mediator  
Adapter Interpreter Template  
**Design\_patterns**  
Visitor Flyweight Strategy  
Singleton Memento  
Builder Composite State  
Factory Observer  
factory Proxy

## MISE A JOUR ET COUTS

Toutes les informations  
sont transmises même  
si elles ne sont pas  
pertinentes.

Couts élevés dans  
certain cas.



# VII. LIENS AVEC AUTRES PATTERN

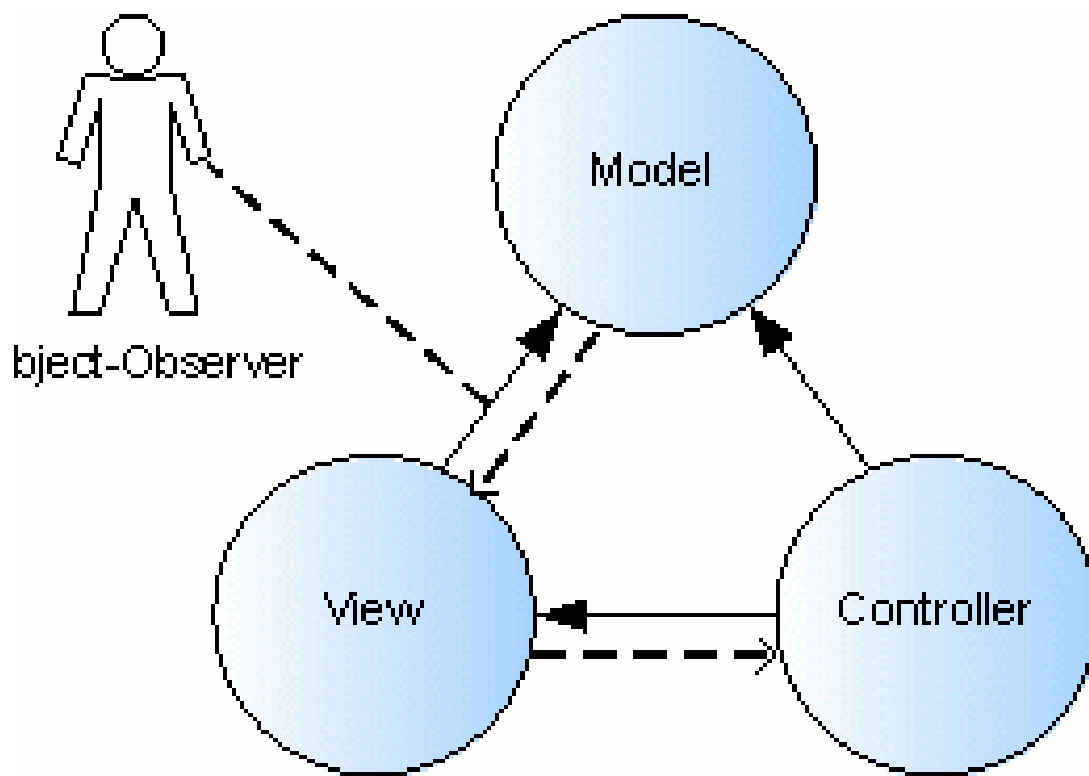


A word cloud of design patterns is centered on a black circle. The patterns are listed in various colors (blue, orange, yellow, white) and sizes. The central and largest text is "Design\_patterns". Other patterns include Command, Bridge, Iterator, Abstract, Chain, Decorator, Facade, Prototype, Mediator, Adapter, Interpreter, Template, Visitor, Flyweight, Strategy, Singleton, Memento, Builder, Composite, State, Factory, Observer, factory, and Proxy.

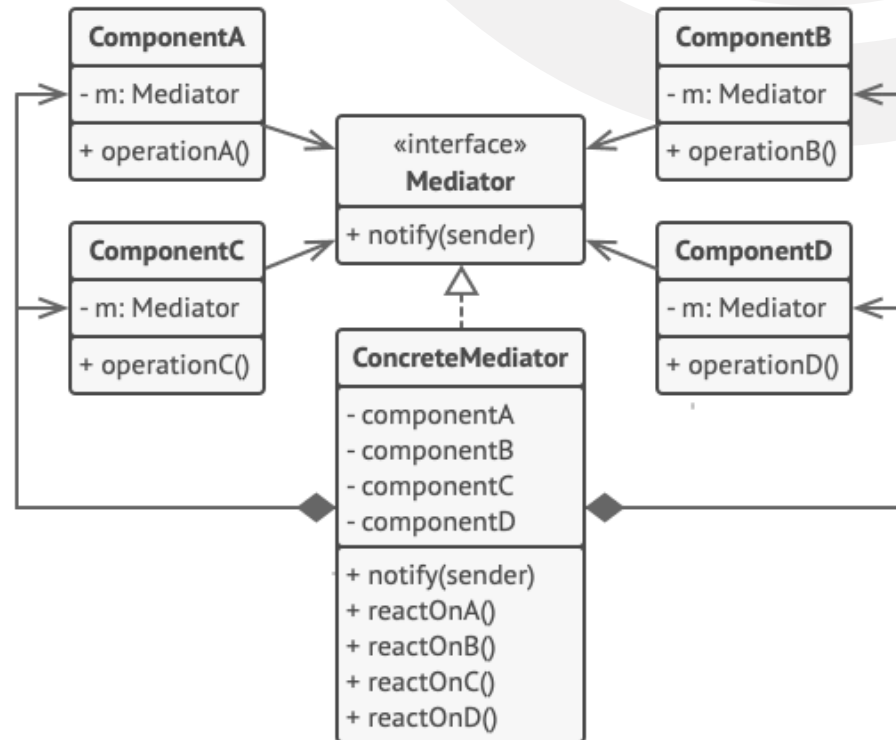
Command Bridge  
Iterator Abstract  
Chain Decorator  
Facade Prototype Mediator  
Adapter Interpreter Template  
**Design\_patterns**  
Visitor Flyweight Strategy  
Singleton Memento  
Builder Composite State  
Factory Observer  
factory Proxy



## MVC



## Mediateur



# VIII. QCM



A word cloud of design patterns is centered on a black circle. The words are arranged in a circular pattern around the central text "Design\_patterns". The words are in various colors (blue, orange, yellow, green) and sizes, with "Design\_patterns" being the largest and most prominent. The words include: Command, Bridge, Iterator, Abstract, Chain, Decorator, Facade, Prototype, Mediator, Adapter, Interpreter, Template, Visitor, Flyweight, Strategy, Singleton, Memento, Builder, Composite, State, Factory, Observer, factory, and Proxy.

Design\_patterns

# DE QUELLE FAMILLE FAIT PARTIE LE PATTERN OBSERVER ?

- ☐ Créateur
- ☐ Comportementaux
- ☐ Structuraux
- ☐ La famille Lopez

# QUEL EST L'UN DES PRINCIPE SOLID QU'IL RESPECTE ?

☐ SRP

☐ OCP

☐ LSP

☐ ISP

# AVEC QUEL(S) AUTRE(S) PATTERN(S) PEUT-IL ÊTRE EN LIEN ?

☐ MVC

☐ Composite

☐ Decorator

☐ J'ai pas écouté la (superbe) présentation

# QUEL SONT LES LIMITES DU PATTERN ?

- ☐ Baisse de résultat
- ☐ Construire une dépendance entre un sujet et des observateurs
- ☐ Mise à jour inattendues
- ☐ Observer la vie humaine