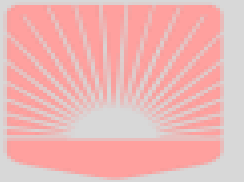




UNIVERSITÉ DE  
VERSAILLES  
ST-QUENTIN-EN-YVELINES  
université PARIS-SACLAY



#6

09/01/2026

jean-michel.batto@cea.fr

cea

[https://gogs.eldarsoft.com/M2\\_IHPS](https://gogs.eldarsoft.com/M2_IHPS)

09 / 01 / 2026



## Installation de clang-format

Dans le dossier .ssh-source

```
sh install_clang_format_go.sh
```

➔ Installation du package et en plus il y a le compilateur go

## Installation de l'extension depuis vscode

EXTENSIONS: MARKETPLACE

@popular

- Debugger for Java** 5.8M ★ 5  
A lightweight Java debugger for Vis...  
vscjava [Install](#)
- Project Manager for Java** 5.7M  
Manage Java projects in Visual Stud...  
vscjava [Install](#)
- Jupyter** 4.9M ★ 5  
Jupyter notebook support, interacti...  
ms-toolsai [Install](#)
- Extension Pack for J...** 4.7M ★ 5  
Popular extensions for Java develop...  
vscjava [Install](#)
- Jupyter Notebook Rend...** 4.5M  
Renderers for Jupyter Notebooks (w...  
ms-toolsai [Install](#)
- GitLens — Git super...** 4.4M ★ 4  
Supercharge Git within VS Code — ...  
GitKraken [Install](#)
- Maven for Java** 4.3M  
Manage Maven projects, execute g...  
vscjava [Install](#)
- GitHub Pull Requests** 4.3M ★ 4  
Pull Request and Issue Provider for ...  
GitHub [Install](#)
- Test Runner for Java** 4.1M ★ 1  
Run and debug JUnit or TestNG test...  
vscjava [Install](#)

**Go** 4M ★ 4

Go

golang | 4,089,634 | ★★★★★ (8)

Rich Go language support for Visual Studio Code

[Install](#) ☒ Auto Update [Settings](#)

DETAILS FEATURES CHANGELOG

## Go for Visual Studio Code

[slack](#) [gophers](#)

The VS Code Go extension provides rich language support for the [Go programming language](#).

### Requirements

- Visual Studio Code 1.90 or newer (or editors compatible with VS Code 1.90+ APIs)
- Go 1.21 or newer.

### Quick Start

Welcome! 🐼

Whether you are new to Go or an experienced Go developer, we hope this extension fits your needs and enhances your development experience.

### Marketplace

Identifier	golang.go
Version	0.52.1
Published	5 years ago
Last Released	2 weeks ago

### Categories

[Programming Languages](#) [Snippets](#) [Linters](#) [Debuggers](#) [Formatters](#) [Testing](#)

### Resources

- [Repository](#)
- [License](#)
- [Marketplace](#)

## Build with Agent

AI responses may be inaccurate.

[Generate Agent Instructions](#) to onboard AI onto your codebase.

### SUGGESTED ACTIONS

[Build Workspace](#) [Show Config](#)

[Add Context...](#)

Describe what to build next



Modification du .profile, (passer root avant)

```
export PATH=/usr/local/go/bin:/usr/lib/code-server/lib/vscode/bin/remote-cli:/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games:/usr/local/go/bin
```



# Pour activer le plugin Go

Settings X {} settings.json

Search settings

User Remote [127.0.0.1:8081] Workspace

Features

- > Application
- > Security
- ✓ Extensions
  - .ipynb Support
  - > CSS Language Features
  - Emmet
  - GHE.com & GitHub En...
  - Git
  - GitHub
  - Go**
  - Grunt
  - Gulp
  - HTML
  - Jake
  - JavaScript Debugger

## Extensions

### Go

tags	json
options	json=omitempty
promptForTags	false
transform	snakecase
template	

#### Alternate Tools


Alternate tools or alternate paths for the same tools used by the Go extension. Provide either absolute path or the name of the binary in GOPATH/bin, GOROOT/bin or PATH. Useful when you want to use wrapper script for the Go tools.

[Edit in settings.json](#)

```
home > coder > .local > share > code-server > User > {} settings.json > {} go.alternateTools > abc go
1  {
2    "workbench.colorTheme": "Quiet Light",
3    "go.alternateTools": {
4      "go": "/usr/local/go/bin/go"
5    }
6  }
```



format clang


**Kylin Clangd**

32K

Fork of clangd, add schema validati...

KylinIdeTeam

Install


**Clang-format disable sele...**

258

Wrap code with clang-format disabl...

izquiratops

Install


**clangd: Extra config**

933

Add more clangd options, what pas...

SR\_team

Install


**Clang-Format**

30K

Use Clang-Format in Visual Studio C...

xaver

Install


**Clang-Format 2025**

1K

Use Clang-Format in Visual Studio C...

ebextensions

Install

**OpenSCAD Formatter**

1K

Simple formatter for OpenSCAD file...

JulianGmp

Install



## Clang-Format 2025

ebextensions | 1,499 | ★★★★★

Use Clang-Format in Visual Studio Code

Install ☒ Auto Update ⚙️

DETAILS

FEATURES

CHANGELOG

## README

license

MIT

Clang-Format is a tool to format many different coding languages. It can be configured with a config file named `.clang-format` in the working folder or a parent folder. Configuration see: <http://clang.llvm.org/docs/ClangFormatStyleOptions.html>

This project is a fork of the older [vscode-clang-format](#) extension, but has been rewritten with modern VSIX and TypeScript best practices. I believe that I have integrated all outstanding feature requests as well.

## Usage

Idée : manipuler un bout de code dans un environnement de test multi-nœuds MPI

Vous allez créer un repo sur gogs

Vous allez cloner le repo <https://gogs.eldarsoft.com/jmbatto/GLCS-CM5-TDXMP> et copier le contenu dans votre repo local

Et nous allons ensemble explorer quelques idées.



<https://xcalablemp.org/handbook>

Auteur : Mitsuhsa Sato

<https://icl.utk.edu/newsletter/presentations/2012/Sato-Updates-on-XcalableMP-PGAS-Language-2012-08-29.pdf> (dans le dossier support biblio)

Modèle de programmation « type SIMD », pour C et Fortran

Extension du langage basée sur des pragmas

Programmation parallèle évolutive et tenant compte des performances

Compilateur source (C+XMP) vers source (C+MPI)

Cartographie des données et des tâches selon patron de programmation





INTERACTIF

~~1/installation code-server et noeuds~~

2/paramétrage de git dans le terminal vscode

```
git config --global user.email "you@example.com"
```

```
git config --global user.name "Your Name"
```

3/ dans gogs, vous allez créer un projet, puis faire un git clone

<https://login:password@votreprojet>

L'idée est de vous permettre de travailler efficacement !

4/ vous récupérer l'exemple GLCS-CM5-TDXMP



INTERACTIF

On peut faire un make run

Si on se positionne comme root avec la commande `ifconfig -a`, on voit 2 réseaux (eth0 et eth1)

```
mpirun --mca orte_base_help_aggregate 0 --mca btl_tcp_if_include  
10.0.1.0/24 -host c1,c2,c3,c4 -n 4 worker_program
```



## Exploration du code init.c

```
#pragma xmp nodes p(2, 2) → on décrit 4 noeuds  
#pragma xmp template t(0 : 3, 0 : 3) → la tâche est une matrice 4 * 4  
#pragma xmp distribute t(block, block) onto p → la tâche t va être distribuée  
sur les 4 nœuds (on va donc avoir t(0,0), t(0,1), t(1,0), t(1,1) → p(0) = le  
head (rank = 0) )  
XMP_Matrix A[4][4]; → une matrice 4x4  
#pragma xmp align A[i][j] with t(j, i) → inversion des indices  
#pragma xmp shadow A[4][4] → on veut un shadow pour la consolidation
```

Plus loin dans le code

```
#pragma xmp reflect(A) → synchronisation de l'espace mémoire
```

<https://xcalablemp.org/handbook/distribute.html>



INTERACTIF

Que fait ce code ?

A quoi sert le `usleep(100)` ?

A quoi servent les barrières MPI

Est-ce que ce code est explicable par un LLM ?

INTERACTIF

Pour ce TD :

Grafana : la télémétrie – pas d'utilité pour ce TD

`philhawthorne/docker-influxdb-grafana:latest`

Vscode : une version http de visual code studio

`codercom/code-server:latest`

4 nœuds MPI nommés : 1 nœud maitre (`slurmctld`) et 4 nœuds esclaves (`c1-4`)

La possibilité d'utiliser SLURM (bonification si utilisation)

## Remarques

Partage de scripts (`certif.sh` et `install_clang_format_go.sh`) via les secrets !

Le filesystem `/usr/local/var/mpishare` est partagé avec vscode.

Les nœuds sont nommés pour avoir un hostname ! (et pas de scale possible)

**Refresh des images au démarrage** – sauf si c'est dans l'espace partagé docker-compose



Vous clonez [http://gogs.eldarsoft.com/M2\\_IHPS/PPCS-CM6-TDXMP.git](http://gogs.eldarsoft.com/M2_IHPS/PPCS-CM6-TDXMP.git)  
dans votre image docker slurmctld

Dans ce TD seuls les conteneurs slurmctld/c1-4 sont utilisées

Il s'agit d'un dépôt « starter » pour travailler sur une implémentation de calcul d'histogramme en utilisant MPI et XMP.

Ce programme est très simple : en entrée un fichier texte avec des valeurs float comprises entre 0.0 et 20.0

En sortie un fichier histogramme, au format texte



INTERACTIF

## Énoncé du problème :

Construire un histogramme à partir d'un fichier de valeurs float (basé sur un générateur),

L'histogramme décrit l'effectif de 20 entiers associés aux valeurs float

Le fichier au format texte est :

100

1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0

2.0,0.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0

3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0

4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0

5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0

6.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,15.0

7.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,15.0,16.0

8.0,9.0,10.0,11.0,12.0,13.0,14.0,15.0,16.0,17.0

9.0,10.0,11.0,12.0,13.0,14.0,15.0,16.0,17.0,18.0

10.0,11.0,12.0,13.0,14.0,15.0,16.0,17.0,18.0,19.0

Effectif des valeurs

Les valeurs

```
//Le résultat est stocké dans une variable globale
//Pour calculer un histogramme il nous faut un tableau de
//valeur (*data) et un effectif (rows)
void calcule_histo(float *data, int rows) {
    for (int i = 0; i < rows; i++) {
        int j = (int)data[i];
        if ((j >= 0) && (j < 20)) {
            data_out[j]++;
        }
    }
}
```





## Quelques instructions pour XMP

```
#pragma xmp nodes p(1)
```

Permet de tester sur 1 noeud

```
#pragma xmp template t[:]
```

Retarde l'allocation des templates

```
#pragma xmp distribute t(block) onto p
```

Distribue les templates sur les noeuds

```
float *data_in;
```

```
#pragma xmp align data_in[i] with t(i)
```

```
#pragma xmp shadow data_in[*]
```

Positionne la variable data\_in  
sur les templates

➔ objet de l'optimisation XMP



#pragma xmp nodes p(2) → notation pour dire qu'il y a 2 noeuds → ils découpent l'espace des 100 templates



t=50;p=0

t=50;p=1

**mpirun -n 2**

#pragma xmp nodes p(\*) → allocation dynamique !

```
xmp_init_mpi(&argc, &argv);
```

→ initialise le contexte MPI

```
xmp_finalize_mpi();
```

→ finalise le contexte MPI

```
//aspect dynamique à fixer dans le code C
```

```
#pragma xmp template_fix t[rows]
```

Il s'agit de définir la taille des templates et  
d'allouer la mémoire

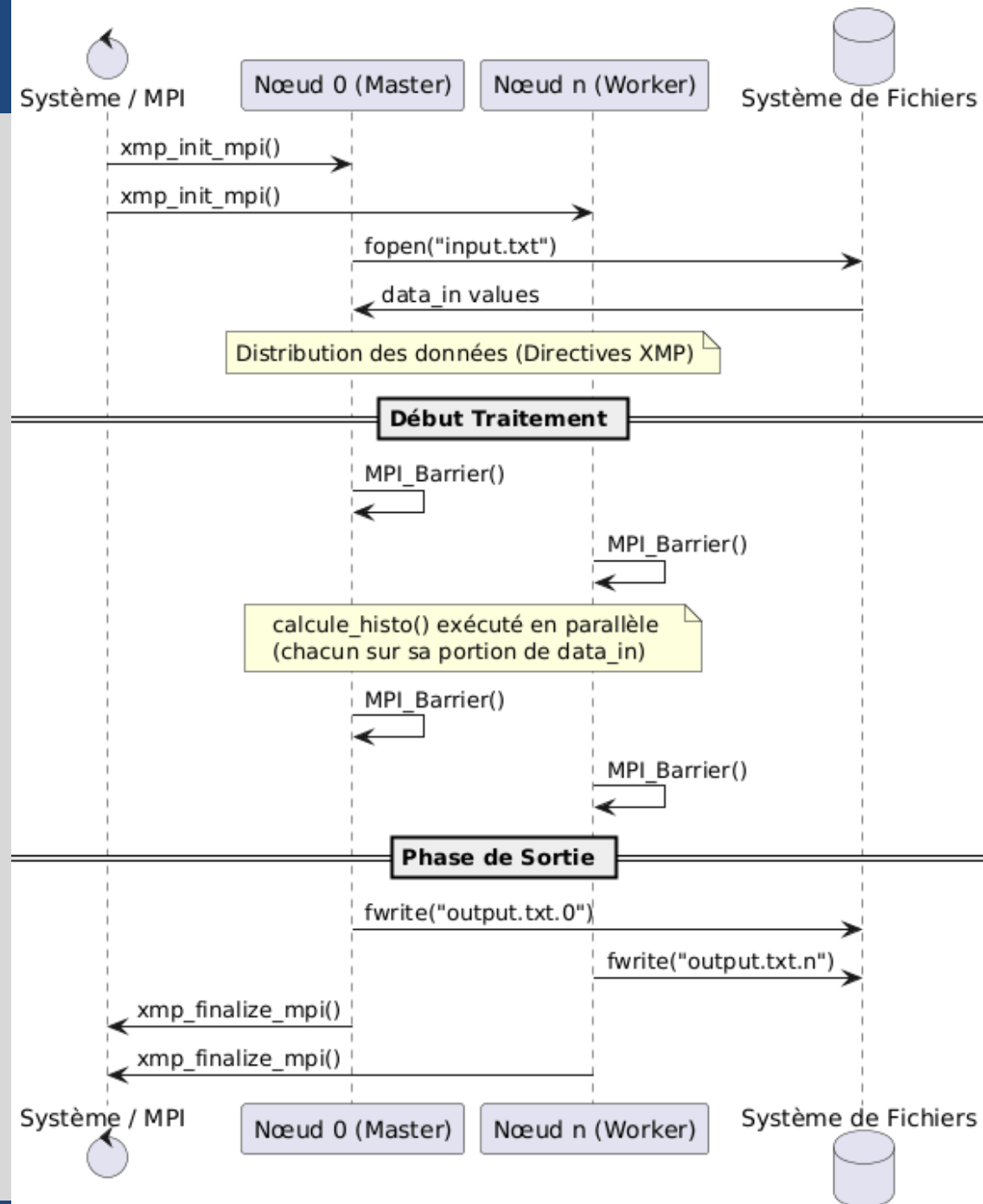
```
data_in = xmp_malloc(xmp_desc_of(data_in), rows);
```



```
void calcule_histo(float *data, int rows) {  
    #pragma xmp loop on t(i)  
    for (int i = 0; i < rows; i++) {  
        int j = (int)data[i];  
        if ((j >= 0) && (j < 20)) {  
            data_out[j]++;  
        }  
    }  
}
```

← Si on enlève ce pragma, plus d'intérêt de XMP

# Cas avec p(2), diagramme de séquence





INTERACTIF

Vous allez tester sur 1 nœud le mécanisme

Vous allez l'étendre à 4 nœuds

→ Quel est le constat sur les résultats ? (par rapport à 1 nœud)

Vous allez ajouter une réduction pour obtenir un résultat conforme au cas sur 1 nœud



Problème il faut faire une réduction

`#pragma xmp reduction(+ : tmp)` → effectue la réduction par rapport aux nœuds

On peut penser que le même code écrit en OpenMP sur 1 seul nœud (donc sans MPI/XMP) ira beaucoup plus vite, car nous sommes sur un supercalculateur simulé via Docker !

# A rendre pour le TD4

Les codes sources de la version « XMP » et de la version « OpenMP »

Ajout d'un timer (plusieurs possibilités : `clock_gettime(CLOCK_MONOTONIC_RAW, &start_time);` ou `double start_time = MPI_Wtime();`)

Puis :

Benchmark avec 10 millions de ligne, 500 Mbytes (le temps de génération du fichier est de 40 secondes, inutile de me transmettre ce fichier) – vous ne travaillerez que sur des « gros » fichiers d'input.

`./scramblegenerator -skewseed votrematricule -outputlinecount 10000000 > ./input.txt`

→ le benchmark consiste à vérifier que 1/2/4 nœuds n'est pas plus performant que la version OpenMP.

**votrematricule** est confidentiel. Inutile de le mettre dans le rapport ou le Makefile



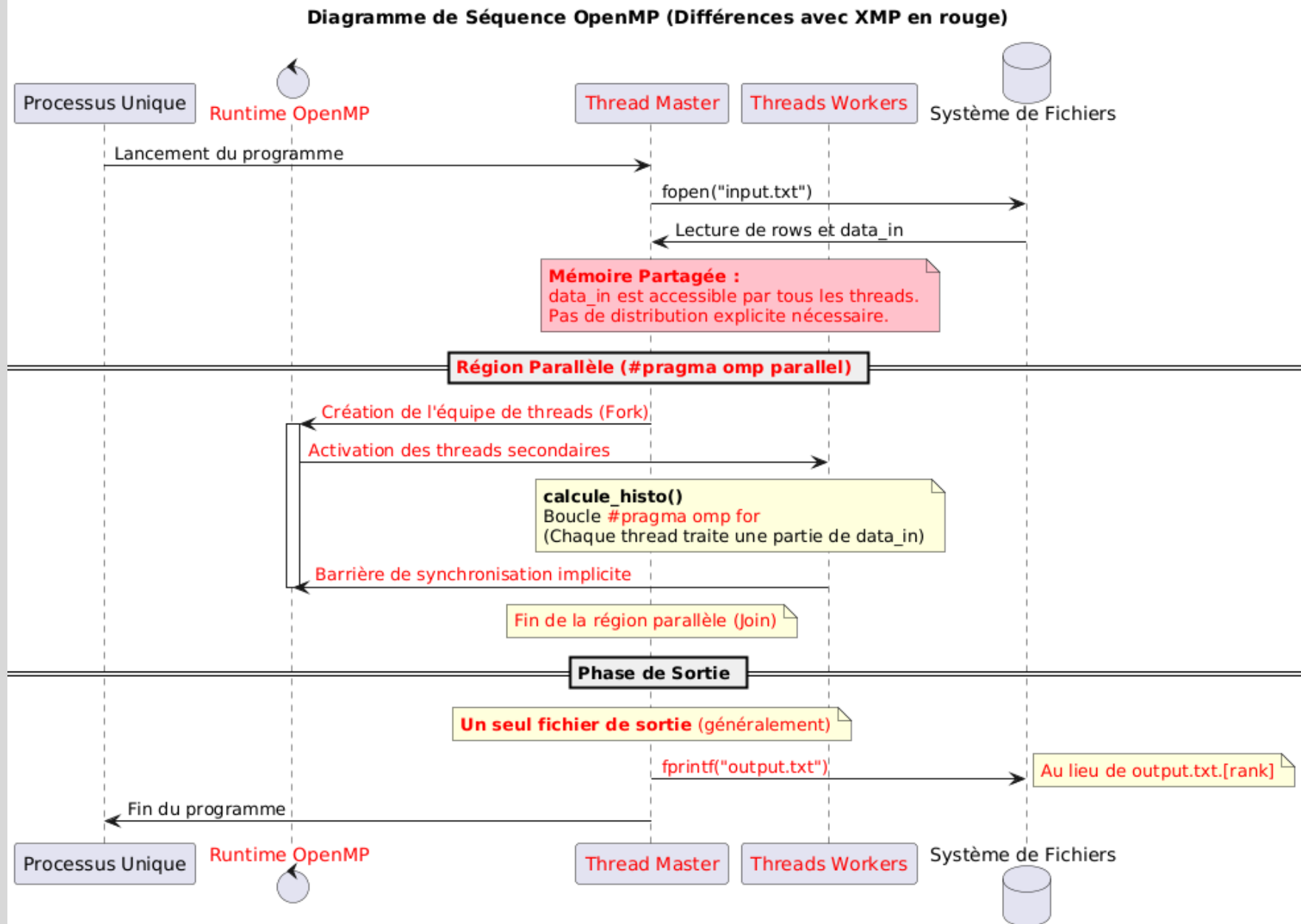


# Version OpenMP souhaitée pour le TD4

Exemple de code élémentaire OMP/XMP

[http://gogs.eldarsoft.com/M2\\_IHPS/PPCS-CM6-OMPXMP.git](http://gogs.eldarsoft.com/M2_IHPS/PPCS-CM6-OMPXMP.git)

→ c'est pour vous aider à démarrer



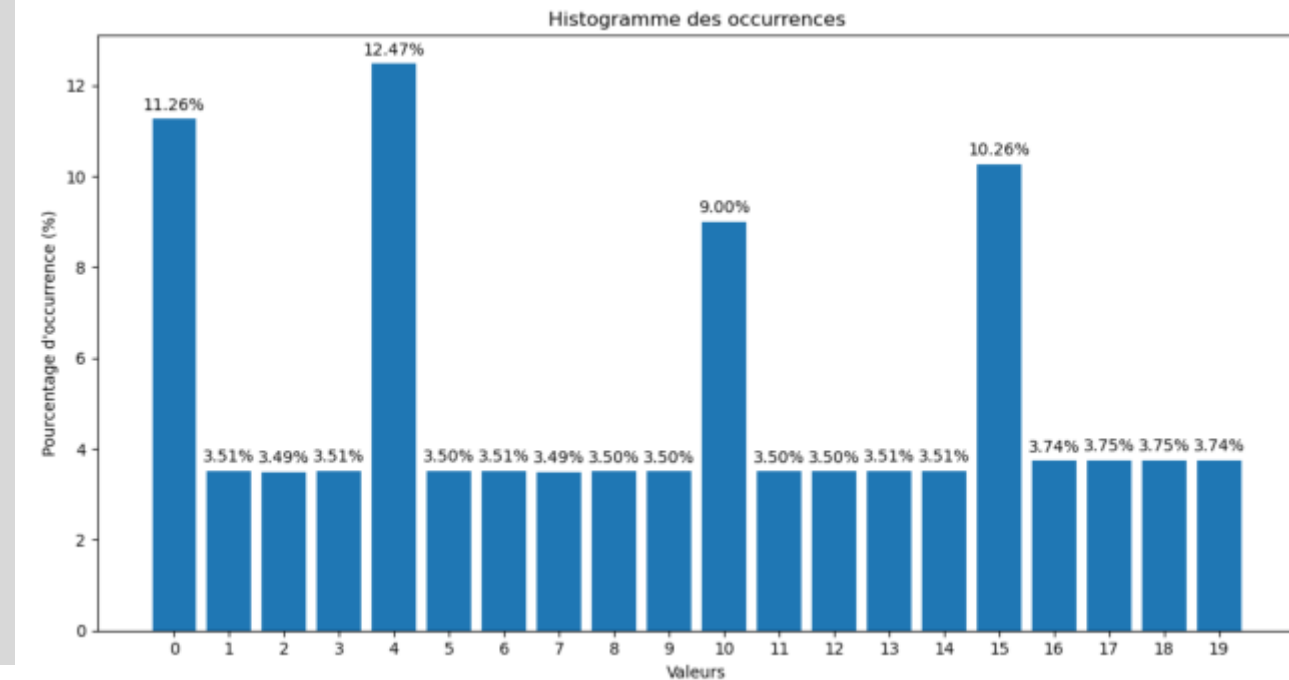
Dans le rendu du TD4 :

Texte PDF avec introduction, présentation du code non modifié, puis vos apports. (attention à la gestion du temps). Les benchmarks exprimeront des speedup

Un ou des diagrammes PlantUML  
(selon le propos)

Un histogramme sera présenté :

Vous donnerez les prompts :



Avec l'utilisation de GrokAI vers. 3 :

À partir de ce fichier, génère un tableau de performance comparant les temps de calcul pour différentes configurations de processeurs et optimisations (-O0 et -O3), incluant les colonnes suivantes :  
Nombre de lignes traitées (100000, 1000000, 10000000), nombre de processeurs utilisés (1, 2, 3, 4), niveau d'optimisation (-O0 ou -O3), et temps de calcul maximum en secondes pour chaque combinaison de ces paramètres, en veillant à bien séparer les temps des 2 niveaux d'optimisation.



TD1 : Docker+MPI, benchmark (moyenne promo : 15.5)

TD2 : Diagramme PlantUML (moyenne promo : 15)

TD3 : Télémétrie applicative (SLURM possible)

TD4 : XMP, OpenMP (SLURM possible sur XMP)

## Elements of Reusable Object-Oriented Software

– Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides  
Chez Addison-Wesley, 1995.

Ces patterns très célèbres ont été conçus par 4 informaticiens surnommés le “Gang of Four” (d’où le terme GoF pour ces patterns).

Ils proposent des solutions élégantes, et toujours différentes pour résoudre différents problèmes récurrents rencontrés par les architectes logiciels.

Patterns et conception : standardiser la conception.

Formalisation de savoir-faire.

Augmenter le niveau d'abstraction → Eviter de construire des objets trop adhérents du réel → simplifie le réemploi.

Arbitrage entre flexibilité et performance → l'adaptation aux évolutions.



En 3 séances (7, 7, 9)

Objectif : apprendre à bien utiliser les patterns de votre choix

+ en Golang : refaire quelques patterns pour bien comprendre le principe des patterns.

Les 23 patterns décrits :

	Creational	Structural	Behavioral
Class	Factory Method	Adapter (class)	Interpreter
			Template Method
Object	Abstract Factory	Adapter (object)	Chain of Responsibility
	Builder	Bridge	Command
	Prototype	Composite	Iterator
	Singleton	Decorator	Mediator
		Facade	Memento
		Flyweight	Observer
		Proxy	State
			Strategy
			Visitor

# Typologie

Typologie	Nom du Design Pattern	Ce qui doit être ajuster	verbe	composition sous jacente	mélange de classes ?
Creational	Abstract Factory	famille d'objets dépendants		structure	non
	Builder	Comment créer un objet composite dont la structure du composite est indépendante		liste	non
	Factory Method	Sous classe d'un objet qui est instanciée sans connaître la classe ancêtre (connaissance retardée)			filtrage vtab
	Prototype	Classe d'objet qui est instanciée grâce à un constructeur de copie	copie=verbe	liste	non
	Singleton	La seule instance d'une classe	copie=o=verbe		non
Structural	Adapter	accède à un objet en modifiant l'interface			non
	Bridge	Fait l'implémentation d'un objet par découplage	découplage		non
	Composite	structure et composition d'un objet vue de manière uniforme		arbre	non
	Decorator	Responsabilité d'un objet sans héritage - ajout dynamique			filtrage vtab
	Facade	Exposer une interface à un sous-système			filtrage vtab
	Flyweight	cout de stockage d'un objet, partage de l'état	état=verbe	liste	non
	Proxy	Comment un objet est accédé, son emplacement (mémoire, disque) - effet miroir		queue	non
Behavioral	Chain of Responsibility	Un objet qui peut répondre à une demande avec découplage	découplage	queue	filtrage vtab
	Command	quand et comment une commande peut être faite - la commande devient un objet		structure	non
	Interpreter	grammaire et interprétation d'un objet			non
	Iterator	se déplacer dans une structure d'objet sans en connaître le détail		liste	filtrage vtab
	Mediator	Comment et avec quels objets sont décrites les interactions			non
	Memento	Quelles sont les informations privées qui sont stockée à part et quand?	état=verbe	état (queue=infinie)	non
	Observer	l'effectif des objets observés et quand s'effectue la mise à jour		queue	non
	State	les états d'un objet sont des variables, avec un handler()	état=verbe	structure	filtrage vtab
	Strategy	un algorithme	extension=verbe	structure	filtrage vtab
	Template Method	les étapes/squelette d'un algorithme		structure	non
	Visitor	les opérations élémentaires sont appliquées à un objet sans modifier sa classe		liste	non





# Réemploi des objets

Spécifier la classe d'un objet explicitement	Abstract Factory
	Factory Method
	Prototype
Coder explicitement les comportements	Command
	Chain of Responsibility
Prise en compte des dépendances	Abstract Factory
	Bridge
Dépendance sur les représentations et les objets	Abstract Factory
	Memento
	Bridge
	Proxy
Dépendance algorithmique	Strategy
	Builder
	Iterator
	Template Method
	Visitor
Couplage léger	Facade
	Mediator
	Observer
	Command
	Abstract Factory
	Bridge
Sous-classe pour étendre le comportement	Bridge
	Composite
	Decorator
	Chain of Responsibility
	Strategy
Incapacité à altérer la classe ancêtre	Visitor
	Decorator
	Adapter

# Creational / Factory Method

Permettre à une classe de créer des objets dont elle ne connaît pas la classe

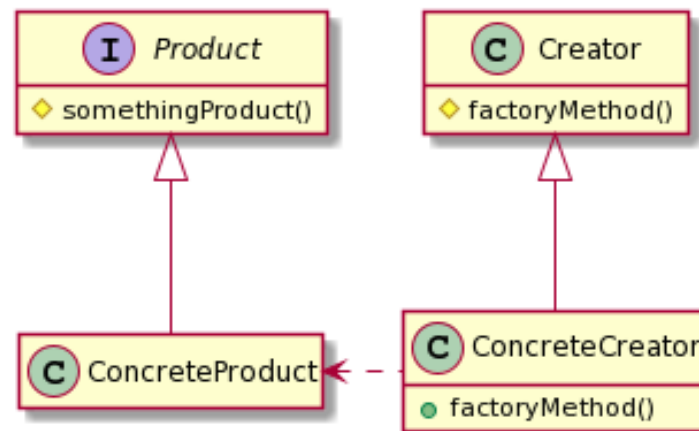
<https://godbolt.org/z/7WPYfzvsa>

```
int main(int argc, char* argv[]) {
    Creator *creator = new ConcreateCreator(ConcreateCreator::A);

    Product *productA = creator->factoryMethod();
    productA->somethingProduct();

    Product *productB = creator->factoryMethod();
    productB->somethingProduct();

}
```



## Factory Method

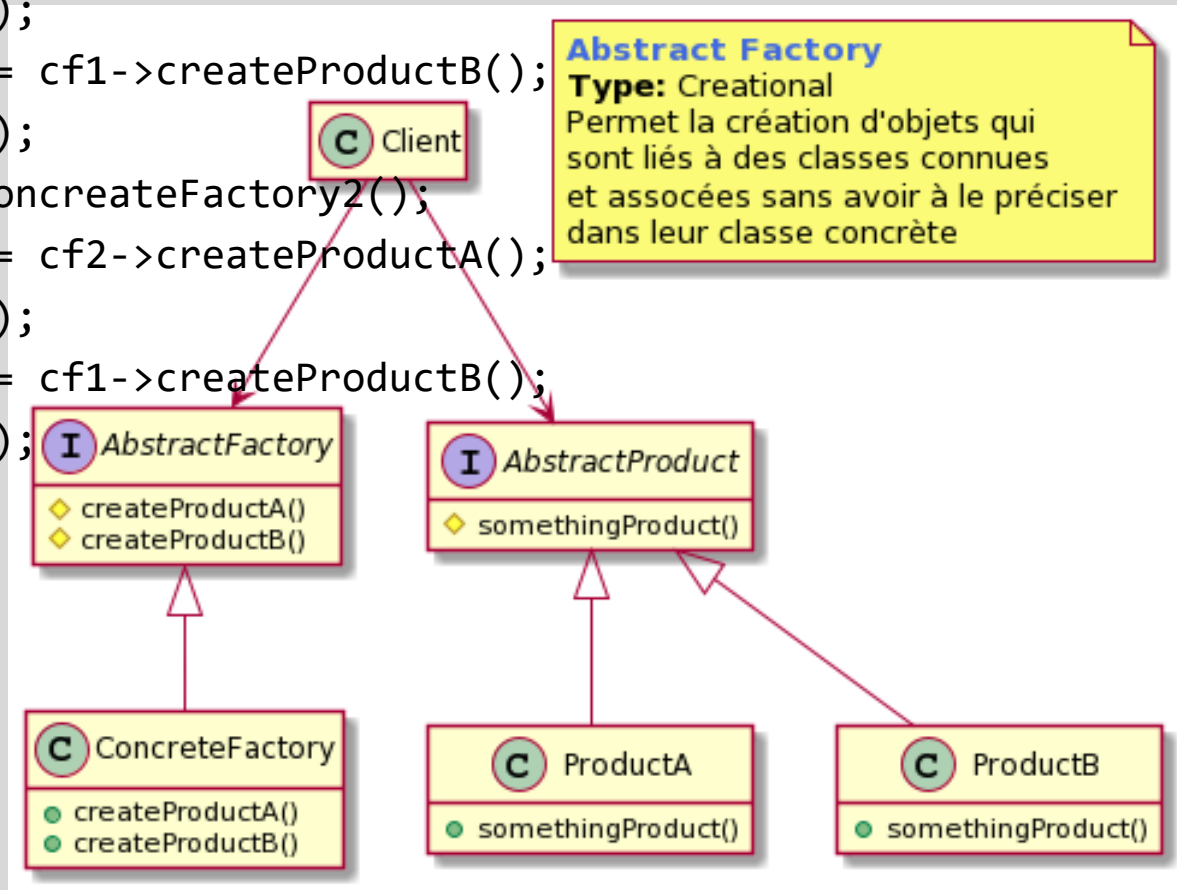
**Type:** Creational

Décrit une interface pour créer un objet, mais laisse la sous-classe décider. L'instanciation est choisie par la sous-classe.

# Creational / Abstract Factory

<https://godbolt.org/z/hh5se4o7j>

```
int main(int argc, char* argv[]) {
    AbstractFactory* cf1 = new ConcreateFactory1();
    AbstractProductA* productA1 = cf1->createProductA();
    productA1->somethingProduct();
    AbstractProductB* productB1 = cf1->createProductB();
    productB1->somethingProduct();
    AbstractFactory* cf2 = new ConcreateFactory2();
    AbstractProductA* productA2 = cf2->createProductA();
    productA2->somethingProduct();
    AbstractProductB* productB2 = cf1->createProductB();
    productB2->somethingProduct();
}
```



// Factory Method Pattern

```
type Animal interface { Speak() string }
```

```
type Dog struct{}
```

```
type Cat struct{}
```

```
func (d *Dog) Speak() string { return "Woof!" }
```

```
func (c *Cat) Speak() string { return "Meow!" }
```

```
func CreateAnimal(animalType string) Animal {  
    switch animalType { case "dog": return &Dog{} case "cat": return &Cat{}  
    default: return nil }
```

```
}
```

// Abstract Factory Pattern

```
type Button interface { Paint() }
```

```
type WinButton struct{}
```

```
type MacButton struct{}
```

```
func (w *WinButton) Paint() { fmt.Println("Windows button") }
```

```
func (m *MacButton) Paint() { fmt.Println("Mac button") }
```

```
type GUIFactory interface { CreateButton() Button }
```

```
type WinFactory struct{}
```

```
type MacFactory struct{}
```

```
func (w *WinFactory) CreateButton() Button { return &WinButton{} }
```

```
func (m *MacFactory) CreateButton() Button { return &MacButton{} }
```

[En Golang](#)

[https://go.dev/play/p/dJ7xS\\_I92S9](https://go.dev/play/p/dJ7xS_I92S9)

<https://go.dev/play/p/BNninTWvAj6>

Les 23 patterns décrits :

	Creational	Structural	Behavioral
Class	Factory Method	Adapter (class)	Interpreter
			Template Method
Object	Abstract Factory	Adapter (object)	13 Chain of Responsibility
	14 Builder	9 Bridge	Command
	Prototype	Composite	Iterator
	8 Singleton	10 Decorator	Mediator
		11 Facade	12 Memento
		Flyweight	Observer
		Proxy	State
			Strategy
			15 Visitor



# Les 23 Patterns

Typologie	Nom du Design Pattern	Ce qui doit être ajuster	verbe	composition sous jacente	mélange de classes ?
Creational	Abstract Factory	famille d'objets dépendants		structure	non
	Builder	Comment créer un objet composite dont la structure du composite est indépendante		liste	non
	Factory Method	Sous classe d'un objet qui est instanciée sans connaître la classe ancêtre (connaissance retardée)			filtrage vtab
	Prototype	Classe d'objet qui est instanciée grâce à un constructeur de copie	copie=verbe	liste	non
	Singleton	La seule instance d'une classe	copie=o=verbe		non
Structural	Adapter	accède à un objet en modifiant l'interface			non
	Bridge	Fait l'implémentation d'un objet par découplage	découplage		non
	Composite	structure et composition d'un objet vue de manière uniforme		arbre	non
	Decorator	Responsabilité d'un objet sans héritage - ajout dynamique			filtrage vtab
	Facade	Exposer une interface à un sous-système			filtrage vtab
	Flyweight	cout de stockage d'un objet, partage de l'état	état=verbe	liste	non
	Proxy	Comment un objet est accédé, son emplacement (mémoire, disque) - effet miroir		queue	non
Behavioral	Chain of Responsibility	Un objet qui peut répondre à une demande avec découplage	découplage	queue	filtrage vtab
	Command	quand et comment une commande peut être faite - la commande devient un objet		structure	non
	Interpreter	grammaire et interprétation d'un objet			non
	Iterator	se déplacer dans une structure d'objet sans en connaître le détail		liste	filtrage vtab
	Mediator	Comment et avec quels objets sont décrites les interactions			non
	Memento	Quelles sont les informations privées qui sont stockée à part et quand?	état=verbe	état (queue=infinie)	non
	Observer	l'effectif des objets observés et quand s'effectue la mise à jour		queue	non
	State	les états d'un objet sont des variables, avec un handler()	état=verbe	structure	filtrage vtab
	Strategy	un algorithme	extension=verbe	structure	filtrage vtab
	Template Method	les étapes/squelette d'un algorithme		structure	non
	Visitor	les opérations élémentaires sont appliquées à un objet sans modifier sa classe		liste	non

Classe qui ne donne qu'une seule instance

<https://godbolt.org/z/5916Mc6Ez>

```
int main(int argc, char* argv[]) {
    Singleton *singleton = Singleton::Instance();
    singleton->checkSingleton();
    //we create a new singleton2 but...
    Singleton *singleton2 = Singleton::Instance();
    singleton2->checkSingleton();
}
```



Singleton

□ static unique\_instance

● static Instance()

## Singleton

**Type:** Creational

Classe qui ne peut avoir qu'une seule instance et qui donne un point d'accès global.

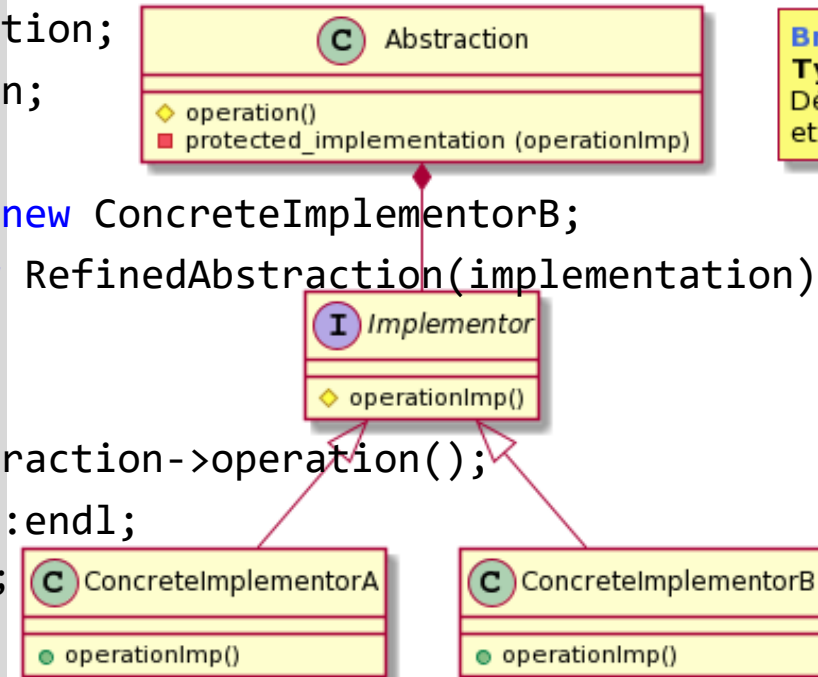
## Découpler le contrat et l'implémentation

<https://godbolt.org/z/EfafGzMWG>

```
int main() {
    Implementor* implementation = new ConcreteImplementorA;
    Abstraction* abstraction = new Abstraction(implementation);
    std::cout << abstraction->operation();
    std::cout << std::endl;
    delete implementation;
    delete abstraction;

    implementation = new ConcreteImplementorB;
    abstraction = new RefinedAbstraction(implementation);

    std::cout << abstraction->operation();
    std::cout << std::endl;
    delete implementation;
    delete abstraction;
}
```



### Bridge

**Type:** Structural

Découple l'abstraction de son implémentation et les 2 peuvent évoluer indépendamment.

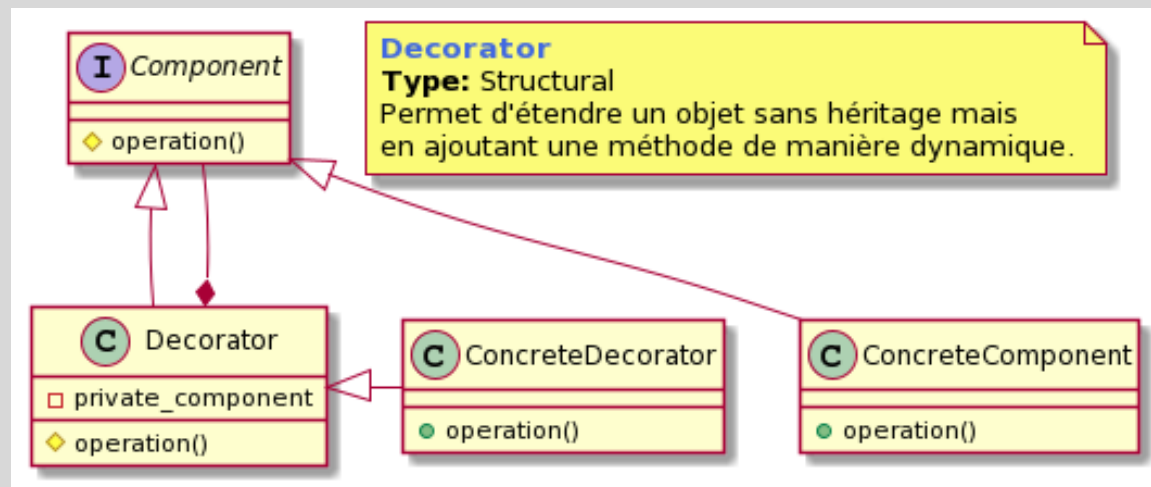




Ajouter des responsabilités dynamiquement

<https://godbolt.org/z/9f8j3xM6r>

```
int main() {  
    Component *cc = new ConcreteComponent();  
    ConcreteDecoratorA *cda = new ConcreteDecoratorA(cc);  
    ConcreteDecoratorB *cdb = new ConcreteDecoratorB(cc);  
  
    cda->operation();  
    cdb->operation();  
}
```



Interface de sous-système simplifiée

<https://godbolt.org/z/oWTWTaPc3>

**private:**

SubSystem1 \*subsystem1;

SubSystem2 \*subsystem2;

SubSystem3 \*subsystem3;

SubSystem4 \*subsystem4;

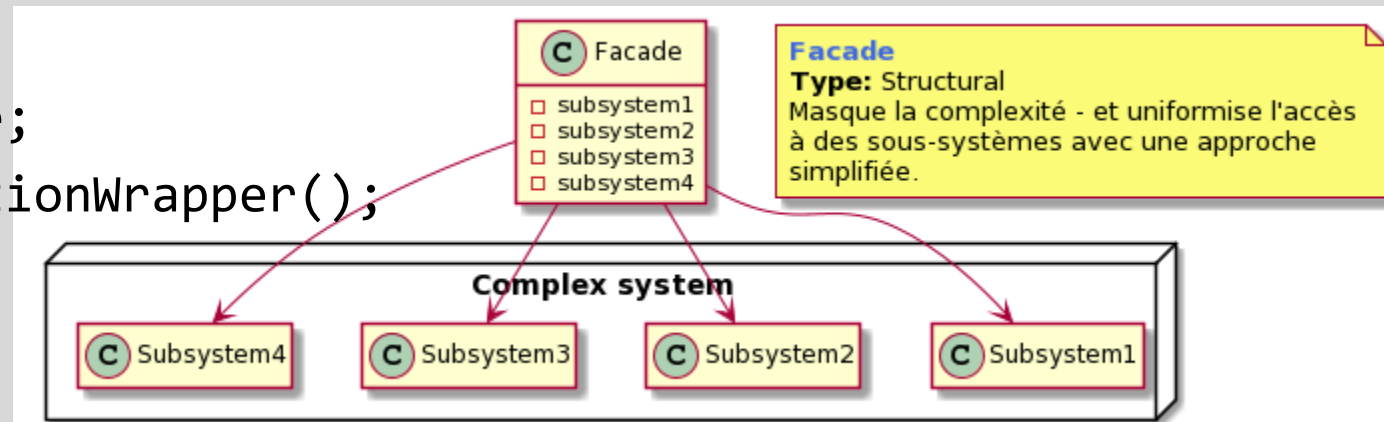
};

**int** main() {

Facade facade;

facade.operationWrapper();

}

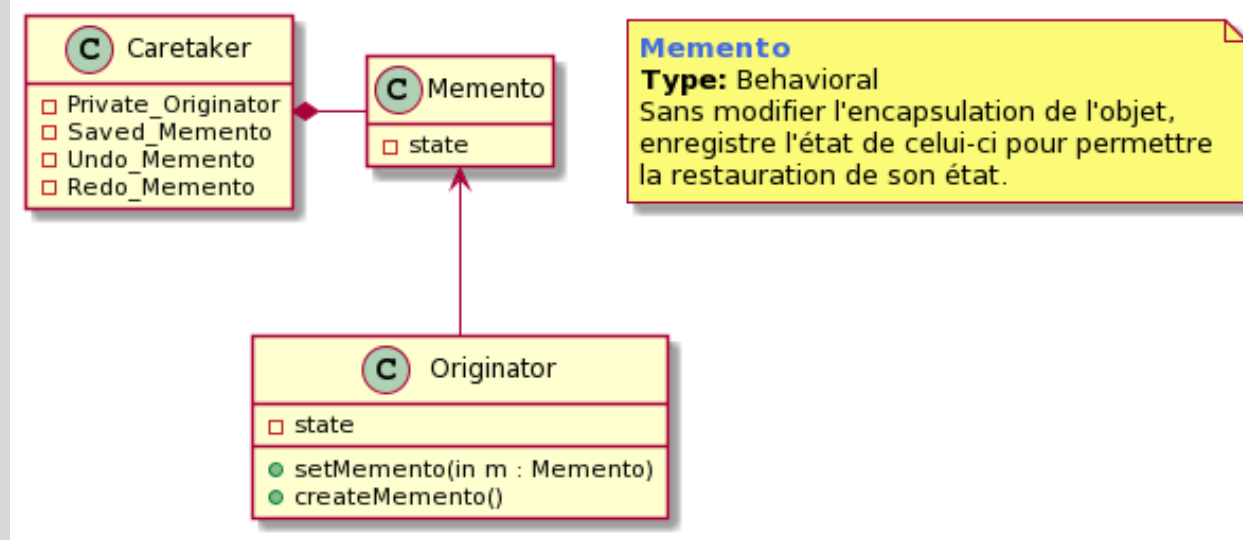


## Enregistrer l'état d'un objet sans briser l'encapsulation

<https://godbolt.org/z/v7qW3TWqb>

```
int main() {
    Originator *originator = new Originator();
    CareTaker *careTaker = new CareTaker( originator);
```

```
    originator->setState(0);
    careTaker->save();
    originator->setState(1);
    careTaker->save();
    originator->setState(2);
    careTaker->save();
    careTaker->printSavedStates();
    careTaker->undo();
    careTaker->printSavedStates();
    careTaker->redo();
    careTaker->printSavedStates();
    careTaker->undo();
    careTaker->undo();
    careTaker->undo();
    careTaker->printSavedStates();
```

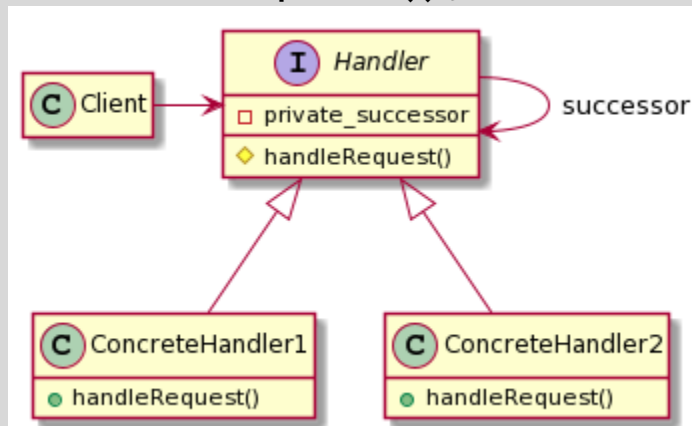


# Behavioral / Chain of Responsibility

Découpler le client et le fournisseur

<https://godbolt.org/z/KMzn87ThM>

```
int main() {
    Client* client = new Client();
    client->h = new ConcreteHandler1();
    Handler* h2 = new ConcreteHandler2();
    client->h->handleRequest();
    client->h->setSuccessor(h2);
    client->h->handleRequest();
    //h2->setSuccessor(client->h);
    client->h->handleRequest();
}
```



## Chain of Responsibility

**Type:** Behavioral

Permet de construire une chaîne de responsabilité avec découplage, le traitement de la requête peut être déroulé par plusieurs objets qui héritent du Handler.

L'algorithme de création est indépendant de la structure

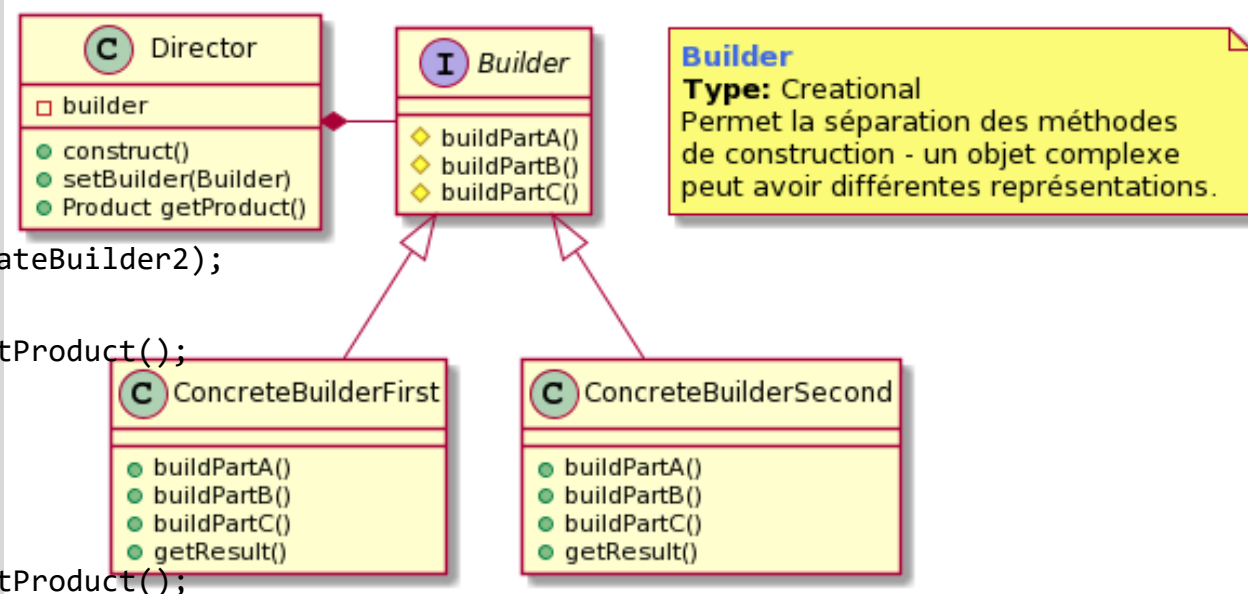
<https://godbolt.org/z/zh8vjWj44>

```
int main(int argc, char* argv[]) {
    Director director;
    director.setBuilder(new ConcreteBuilder1);
    director.construct();
    Product product1 = director.getProduct();
    product1.checkProduct();
```

```
    director.setBuilder(new ConcreteBuilder2);
    director.construct();
    Product product2 = director.getProduct();
    product2.checkProduct();
```

```
    director.setBuilder(0);
    director.construct();
    Product product3 = director.getProduct();
    product3.checkProduct();
```

```
}
```



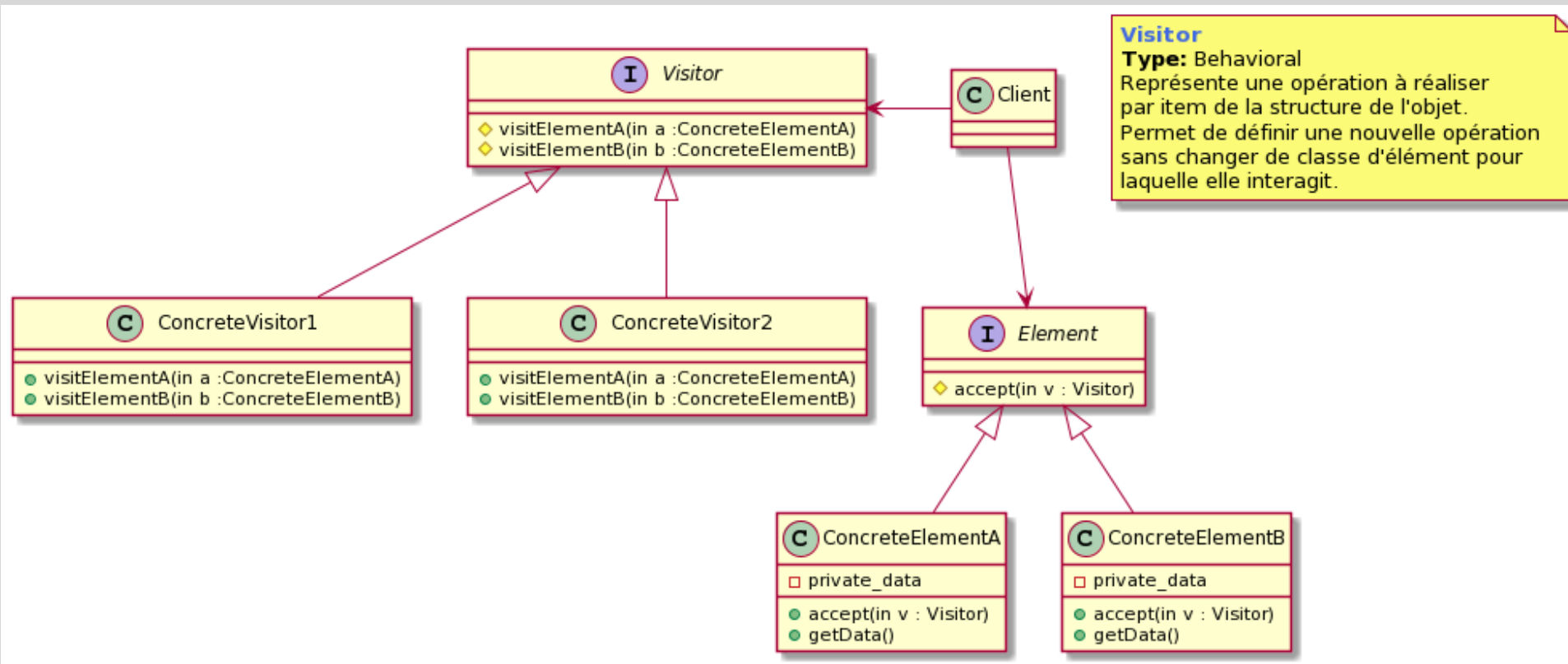
## Builder

**Type:** Creational

Permet la séparation des méthodes de construction - un objet complexe peut avoir différentes représentations.

les opérations élémentaires sont appliquées à un objet sans modifier sa classe

<https://godbolt.org/z/6zsxsaPog>





```
int main() {  
    ConcreteVisitor1 visitor1;  
    ConcreteVisitor2 visitor2;  
    ConcreteElementA elementA("String ElementA ConcreteElementA");  
    elementA.accept(visitor1);  
    elementA.accept(visitor2);  
    ConcreteElementB elementB("String ElementB ConcreteElementB");  
    elementB.accept(visitor1);  
    elementB.accept(visitor2);  
}
```