READYDEV

[GitHub](https://github.com/gkesse) | [LinkedIn](https://www.linkedin.com/in/tia-gerard-kesse/) | [SiteWeb](http://readydev.freeboxos.fr/)

Par Gérard KESSE

Plateforme de Développement Continu

Comprendre la Théorie pour mieux Pratiquer

Sciences de l'Ingénieur

Cours - Tutoriels

READYDEV

[GitHub](https://github.com/gkesse) | [LinkedIn](https://www.linkedin.com/in/tia-gerard-kesse/) | [SiteWeb](http://readydev.freeboxos.fr/)

Par Gérard KESSE

J'aime, Je partage

Montez en Compétences

Patrons de conception

Apprendre les patrons de conception en C++

# [Auteur](#_Sommaire)

Je suis **Gérard KESSE**,

Ingénieur en Développement Informatique C/C++/Qt,

Avec à la fois des compétences en Systèmes Embarqués et en Robotique.

Formé à Polytech'Montpellier, Je suis un professionnel de conception de projets logiciel applicatif ou embarqué dans les secteurs de l'Aéronautique, de la Robotique, des Drones et de la Vision par Ordinateur. Aussi, Je reste ouvert à d'autres types de secteurs tels que l'Énergie et les Finances.

Les Sciences de l’Ingénieur sont au cœur du métier d’ingénieur. Sur le site

**ReadyDev**, la Plateforme de Développement Continu, dont j'en suis le concepteur, vous trouverez des cours et des tutoriels adaptés aux sciences de l’ingénieur.

J'aime, Je partage.

Gérard KESSE

[GitHub](https://github.com/gkesse) | [LinkedIn](https://www.linkedin.com/in/tia-gerard-kesse/) | [SiteWeb](http://readydev.freeboxos.fr/)

[](http://readydev.freeboxos.fr/)

# Sommaire

[Auteur 2](#_Toc511137206)

[Sommaire 3](#_Toc511137207)

[Introduction 4](#_Toc511137208)

[Installation sous Windows avec MinGW 4](#_Toc511137209)

[Téléchargements 4](#_Toc511137210)

[Installation de MinGW 5](#_Toc511137211)

[Installation de Notepad++ 5](#_Toc511137212)

[Patrons de création 5](#_Toc511137213)

[Patron Singleton (Singleton Pattern) 5](#_Toc511137214)

[Patron Fabrique (Factory Pattern) 10](#_Toc511137215)

[Patron Fabrique Abstraite (Abstract Factory Pattern) 11](#_Toc511137216)

[Patron Prototype (Prototype Pattern) 12](#_Toc511137217)

[Patron Monteur (Builder Pattern) 13](#_Toc511137218)

[Patrons de structure 14](#_Toc511137219)

[Patron Pont (Bridge Pattern) 14](#_Toc511137220)

[Patron Façade (Facade Pattern) 15](#_Toc511137221)

[Patron Adaptateur (Adapter Pattern) 16](#_Toc511137222)

[Patron Objet Composite (Composite Pattern) 17](#_Toc511137223)

[Patron Proxy (Proxy Pattern) 18](#_Toc511137224)

[Patron Poids-mouche (Flyweight Pattern) 19](#_Toc511137225)

[Patron Décorateur (Decorator Pattern) 20](#_Toc511137226)

[Patrons de comportement 21](#_Toc511137227)

[Patron Chaine de responsabilité (Chaine-of-responsability Pattern) 21](#_Toc511137228)

[Patron Commande (Command Pattern) 22](#_Toc511137229)

[Patron Interpréteur (Interpreter Pattern) 23](#_Toc511137230)

[Patron Itérateur (Iterator Pattern) 24](#_Toc511137231)

[Patron Médiateur (Mediator Pattern) 25](#_Toc511137232)

[Patron Mémento (Memento Pattern) 26](#_Toc511137233)

[Patron Observateur (Observer Pattern) 27](#_Toc511137234)

[Patron Etat (State Pattern) 28](#_Toc511137235)

[Patron Stratégie (Strategy Pattern) 29](#_Toc511137236)

[Patron Patron-de-méthode (Template method Pattern) 30](#_Toc511137237)

[Patron Visiteur (Visitor Pattern) 31](#_Toc511137238)

# [Introduction](#_Sommaire)

Le C++ est un langage de programmation orienté objet. Les patrons de conception sont des modèles de conception basés sur la programmation orientée objet et nécessaires au développement d’une application évolutive, modulaire et réutilisable dans le temps. Le but de ce tutoriel est de vous apprendre les patrons de conception en C++.

**Prérequis :**

Apprendre la programmation orientée objet en C++.

# [Installation sous Windows avec MinGW](#_Sommaire)

## [Téléchargements](#_Sommaire)

**Notepad++ :**

<https://notepad-plus-plus.org/fr/>

**MinGW :**

<http://www.mingw.org/>

## [Installation de MinGW](#_Sommaire)

**Packages MinGW :**

mingw32-base

mingw32-gcc-g++

## [Installation de Notepad++](#_Sommaire)

**Plugins Notepad++ :**

TextFX

NppExport

# [Patrons de création](#_Sommaire)

## [Patron Singleton (Singleton Pattern)](#_Sommaire)

**Objectif :**

Créer un patron Singleton en C++.

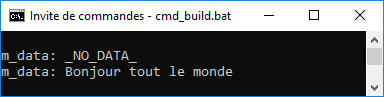
**Utilité :**

Le patron Singleton (Singleton Pattern) est utilisé pour avoir une instance unique d’une classe dans un programme.

**Implémentation :**

Créer un Singleton (GSingleton) permettant d’enregistrer (setData()) et d’afficher (showData()) une donnée (m\_data). Créer une fonction (saveData()) utilisant la méthode (setData()) du Singleton (GSingleton) pour enregistrer la donnée (Bonjour tout le monde). Créer une fonction (printData()) utilisant la méthode (showData()) du Singleton (GSingleton) pour afficher la donnée (Bonjour tout le monde). La donnée (m\_data) du Singleton (GSingleton) sera initialisée à (\_NO\_DATA\_).

**Résultat :**



**Dossier projet :**

src/main.cpp

src/GSingleton.h

src/GSingleton.cpp

win/Makefile

win/cmd\_build.bat

win/cmd\_clean.bat

win/cmd\_compile.bat

win/cmd\_run.bat

win/bin/

win/build/

**src/main.cpp**

//===============================================

#include "GSingleton.h"

//===============================================

void saveData**(**string data**)** **{**

GSingleton**::**Instance**()->**setData**(**data**);**

**}**

//===============================================

void printData**()** **{**

string m\_data **=** GSingleton**::**Instance**()->**getData**();**

cout **<<** "m\_data: " **<<** m\_data **<<** "\n"**;**

**}**

//===============================================

int main**(**int argc**,** char**\*\*** argv**)** **{**

printData**();**

saveData**(**"Bonjour tout le monde"**);**

printData**();**

**return** 0**;**

**}**

//===============================================

**src/GSingleton.cpp**

//===============================================

#ifndef \_GSingleton\_

#define \_GSingleton\_

//===============================================

#include <iostream>

#include <string>

//===============================================

**using** **namespace** std**;**

//===============================================

class GSingleton **{**

public**:**

GSingleton**();**

**~**GSingleton**();**

public**:**

static GSingleton**\*** Instance**();**

void setData**(**const string**&** data**);**

string getData**()** const**;**

private**:**

static GSingleton**\*** m\_instance**;**

string m\_data**;**

**};**

//===============================================

#endif

//===============================================

**src/GSingleton.cpp**

//===============================================

#include "GSingleton.h"

//===============================================

GSingleton**\*** GSingleton**::**m\_instance **=** 0**;**

//===============================================

GSingleton**::**GSingleton**()** **{**

m\_data **=** "\_NO\_DATA\_"**;**

**}**

//===============================================

GSingleton**::~**GSingleton**()** **{**

**}**

//===============================================

GSingleton**\*** GSingleton**::**Instance**()** **{**

**if(**m\_instance **==** 0**)** **{**

m\_instance **=** **new** GSingleton**;**

**}**

**return** m\_instance**;**

**}**

//===============================================

void GSingleton**::**setData**(**const string**&** data**)** **{**

m\_data **=** data**;**

**}**

//===============================================

string GSingleton**::**getData**()** const **{**

**return** m\_data**;**

**}**

//===============================================

**win/Makefile**

GSRC **=** ../src

GBIN **=** bin

GBUILD **=** build

GTARGET **=** $(GBIN)/GProject.exe

GOBJS **=** \

$(GBUILD)/main.o \

$(GBUILD)/GSingleton.o \

all**:** $(GOBJS)

g++ -o $(GTARGET) $(GOBJS)

$(GBUILD)/main.o**:** $(GSRC)/main.cpp

g++ -c $(GSRC)/main.cpp -o $(GBUILD)/main.o

$(GBUILD)/GSingleton.o**:** $(GSRC)/GSingleton.cpp

g++ -c $(GSRC)/GSingleton.cpp -o $(GBUILD)/GSingleton.o

clean**:**

del /q $(GBUILD)\\* $(GBIN)\\*

**Construction du projet :**

cmd\_build.bat

**Construction manuel :**

del /q build\\* bin\\*

g++ -c ../src/main.cpp -o build/main.o

g++ -c ../src/GSingleton.cpp -o build/GSingleton.o

g++ -o bin/GProject.exe build/main.o build/GSingleton.o

## [Patron Prototype (Prototype Pattern)](#_Sommaire)

**Objectif :**

Créer un patron Singleton en C++.

**Utilité :**

Le patron Singleton (Singleton Pattern) est utilisé pour

**Implémentation :**

**Résultat :**

**Dossier projet :**

src/main.cpp

src/GSingleton.h

src/GSingleton.cpp

win/Makefile

win/cmd\_build.bat

win/cmd\_clean.bat

win/cmd\_compile.bat

win/cmd\_run.bat

win/bin/

win/build/

**src/main.cpp**

## [Patron Fabrique (Factory Pattern)](#_Sommaire)

**Objectif :**

Créer un patron Singleton en C++.

**Utilité :**

Le patron Singleton (Singleton Pattern) est utilisé pour

**Implémentation :**

**Résultat :**

**Dossier projet :**

src/main.cpp

src/GSingleton.h

src/GSingleton.cpp

win/Makefile

win/cmd\_build.bat

win/cmd\_clean.bat

win/cmd\_compile.bat

win/cmd\_run.bat

win/bin/

win/build/

**src/main.cpp**

## [Patron Fabrique Abstraite (Abstract Factory Pattern)](#_Sommaire)

**Objectif :**

Créer un patron Singleton en C++.

**Utilité :**

Le patron Singleton (Singleton Pattern) est utilisé pour

**Implémentation :**

**Résultat :**

**Dossier projet :**

src/main.cpp

src/GSingleton.h

src/GSingleton.cpp

win/Makefile

win/cmd\_build.bat

win/cmd\_clean.bat

win/cmd\_compile.bat

win/cmd\_run.bat

win/bin/

win/build/

**src/main.cpp**

## [Patron Monteur (Builder Pattern)](#_Sommaire)

**Objectif :**

Créer un patron Singleton en C++.

**Utilité :**

Le patron Singleton (Singleton Pattern) est utilisé pour

**Implémentation :**

**Résultat :**

**Dossier projet :**

src/main.cpp

src/GSingleton.h

src/GSingleton.cpp

win/Makefile

win/cmd\_build.bat

win/cmd\_clean.bat

win/cmd\_compile.bat

win/cmd\_run.bat

win/bin/

win/build/

**src/main.cpp**

# [Patrons de structure](#_Sommaire)

## [Patron Pont (Bridge Pattern)](#_Sommaire)

**Objectif :**

Créer un patron Singleton en C++.

**Utilité :**

Le patron Singleton (Singleton Pattern) est utilisé pour

**Implémentation :**

**Résultat :**

**Dossier projet :**

src/main.cpp

src/GSingleton.h

src/GSingleton.cpp

win/Makefile

win/cmd\_build.bat

win/cmd\_clean.bat

win/cmd\_compile.bat

win/cmd\_run.bat

win/bin/

win/build/

**src/main.cpp**

## [Patron Façade (Facade Pattern)](#_Sommaire)

**Objectif :**

Créer un patron Singleton en C++.

**Utilité :**

Le patron Singleton (Singleton Pattern) est utilisé pour

**Implémentation :**

**Résultat :**

**Dossier projet :**

src/main.cpp

src/GSingleton.h

src/GSingleton.cpp

win/Makefile

win/cmd\_build.bat

win/cmd\_clean.bat

win/cmd\_compile.bat

win/cmd\_run.bat

win/bin/

win/build/

**src/main.cpp**

## [Patron Adaptateur (Adapter Pattern)](#_Sommaire)

**Objectif :**

Créer un patron Singleton en C++.

**Utilité :**

Le patron Singleton (Singleton Pattern) est utilisé pour

**Implémentation :**

**Résultat :**

**Dossier projet :**

src/main.cpp

src/GSingleton.h

src/GSingleton.cpp

win/Makefile

win/cmd\_build.bat

win/cmd\_clean.bat

win/cmd\_compile.bat

win/cmd\_run.bat

win/bin/

win/build/

**src/main.cpp**

## [Patron Objet composite (Composite Pattern)](#_Sommaire)

**Objectif :**

Créer un patron Singleton en C++.

**Utilité :**

Le patron Singleton (Singleton Pattern) est utilisé pour

**Implémentation :**

**Résultat :**

**Dossier projet :**

src/main.cpp

src/GSingleton.h

src/GSingleton.cpp

win/Makefile

win/cmd\_build.bat

win/cmd\_clean.bat

win/cmd\_compile.bat

win/cmd\_run.bat

win/bin/

win/build/

**src/main.cpp**

## [Patron Proxy (Proxy Pattern)](#_Sommaire)

**Objectif :**

Créer un patron Singleton en C++.

**Utilité :**

Le patron Singleton (Singleton Pattern) est utilisé pour

**Implémentation :**

**Résultat :**

**Dossier projet :**

src/main.cpp

src/GSingleton.h

src/GSingleton.cpp

win/Makefile

win/cmd\_build.bat

win/cmd\_clean.bat

win/cmd\_compile.bat

win/cmd\_run.bat

win/bin/

win/build/

**src/main.cpp**

## [Patron Poids-mouche (Flyweight Pattern)](#_Sommaire)

**Objectif :**

Créer un patron Singleton en C++.

**Utilité :**

Le patron Singleton (Singleton Pattern) est utilisé pour

**Implémentation :**

**Résultat :**

**Dossier projet :**

src/main.cpp

src/GSingleton.h

src/GSingleton.cpp

win/Makefile

win/cmd\_build.bat

win/cmd\_clean.bat

win/cmd\_compile.bat

win/cmd\_run.bat

win/bin/

win/build/

**src/main.cpp**

## [Patron Décorateur (Decorator Pattern)](#_Sommaire)

**Objectif :**

Créer un patron Singleton en C++.

**Utilité :**

Le patron Singleton (Singleton Pattern) est utilisé pour

**Implémentation :**

**Résultat :**

**Dossier projet :**

src/main.cpp

src/GSingleton.h

src/GSingleton.cpp

win/Makefile

win/cmd\_build.bat

win/cmd\_clean.bat

win/cmd\_compile.bat

win/cmd\_run.bat

win/bin/

win/build/

**src/main.cpp**

# [Patrons de comportement](#_Sommaire)

## [Patron Chaine de responsabilité (Chaine-of-responsability Pattern)](#_Sommaire)

**Objectif :**

Créer un patron Singleton en C++.

**Utilité :**

Le patron Singleton (Singleton Pattern) est utilisé pour

**Implémentation :**

**Résultat :**

**Dossier projet :**

src/main.cpp

src/GSingleton.h

src/GSingleton.cpp

win/Makefile

win/cmd\_build.bat

win/cmd\_clean.bat

win/cmd\_compile.bat

win/cmd\_run.bat

win/bin/

win/build/

**src/main.cpp**

## [Patron Commande (Command Pattern)](#_Sommaire)

**Objectif :**

Créer un patron Singleton en C++.

**Utilité :**

Le patron Singleton (Singleton Pattern) est utilisé pour

**Implémentation :**

**Résultat :**

**Dossier projet :**

src/main.cpp

src/GSingleton.h

src/GSingleton.cpp

win/Makefile

win/cmd\_build.bat

win/cmd\_clean.bat

win/cmd\_compile.bat

win/cmd\_run.bat

win/bin/

win/build/

**src/main.cpp**

## [Patron Interpréteur (Interpreter Pattern)](#_Sommaire)

**Objectif :**

Créer un patron Singleton en C++.

**Utilité :**

Le patron Singleton (Singleton Pattern) est utilisé pour

**Implémentation :**

**Résultat :**

**Dossier projet :**

src/main.cpp

src/GSingleton.h

src/GSingleton.cpp

win/Makefile

win/cmd\_build.bat

win/cmd\_clean.bat

win/cmd\_compile.bat

win/cmd\_run.bat

win/bin/

win/build/

**src/main.cpp**

## [Patron Itérateur (Iterator Pattern)](#_Sommaire)

**Objectif :**

Créer un patron Singleton en C++.

**Utilité :**

Le patron Singleton (Singleton Pattern) est utilisé pour

**Implémentation :**

**Résultat :**

**Dossier projet :**

src/main.cpp

src/GSingleton.h

src/GSingleton.cpp

win/Makefile

win/cmd\_build.bat

win/cmd\_clean.bat

win/cmd\_compile.bat

win/cmd\_run.bat

win/bin/

win/build/

**src/main.cpp**

## [Patron Médiateur (Mediator Pattern)](#_Sommaire)

**Objectif :**

Créer un patron Singleton en C++.

**Utilité :**

Le patron Singleton (Singleton Pattern) est utilisé pour

**Implémentation :**

**Résultat :**

**Dossier projet :**

src/main.cpp

src/GSingleton.h

src/GSingleton.cpp

win/Makefile

win/cmd\_build.bat

win/cmd\_clean.bat

win/cmd\_compile.bat

win/cmd\_run.bat

win/bin/

win/build/

**src/main.cpp**

## [Patron Mémento (Memento Pattern)](#_Sommaire)

**Objectif :**

Créer un patron Singleton en C++.

**Utilité :**

Le patron Singleton (Singleton Pattern) est utilisé pour

**Implémentation :**

**Résultat :**

**Dossier projet :**

src/main.cpp

src/GSingleton.h

src/GSingleton.cpp

win/Makefile

win/cmd\_build.bat

win/cmd\_clean.bat

win/cmd\_compile.bat

win/cmd\_run.bat

win/bin/

win/build/

**src/main.cpp**

## [Patron Observateur (Observer Pattern)](#_Sommaire)

**Objectif :**

Créer un patron Singleton en C++.

**Utilité :**

Le patron Singleton (Singleton Pattern) est utilisé pour

**Implémentation :**

**Résultat :**

**Dossier projet :**

src/main.cpp

src/GSingleton.h

src/GSingleton.cpp

win/Makefile

win/cmd\_build.bat

win/cmd\_clean.bat

win/cmd\_compile.bat

win/cmd\_run.bat

win/bin/

win/build/

**src/main.cpp**

## [Patron Etat (State Pattern)](#_Sommaire)

**Objectif :**

Créer un patron Singleton en C++.

**Utilité :**

Le patron Singleton (Singleton Pattern) est utilisé pour

**Implémentation :**

**Résultat :**

**Dossier projet :**

src/main.cpp

src/GSingleton.h

src/GSingleton.cpp

win/Makefile

win/cmd\_build.bat

win/cmd\_clean.bat

win/cmd\_compile.bat

win/cmd\_run.bat

win/bin/

win/build/

**src/main.cpp**

## [Patron Stratégie (Strategy Pattern)](#_Sommaire)

**Objectif :**

Créer un patron Singleton en C++.

**Utilité :**

Le patron Singleton (Singleton Pattern) est utilisé pour

**Implémentation :**

**Résultat :**

**Dossier projet :**

src/main.cpp

src/GSingleton.h

src/GSingleton.cpp

win/Makefile

win/cmd\_build.bat

win/cmd\_clean.bat

win/cmd\_compile.bat

win/cmd\_run.bat

win/bin/

win/build/

**src/main.cpp**

## [Patron Patron-de-méthode (Template method Pattern)](#_Sommaire)

**Objectif :**

Créer un patron Singleton en C++.

**Utilité :**

Le patron Singleton (Singleton Pattern) est utilisé pour

**Implémentation :**

**Résultat :**

**Dossier projet :**

src/main.cpp

src/GSingleton.h

src/GSingleton.cpp

win/Makefile

win/cmd\_build.bat

win/cmd\_clean.bat

win/cmd\_compile.bat

win/cmd\_run.bat

win/bin/

win/build/

**src/main.cpp**

## [Patron Visiteur (Visitor Pattern)](#_Sommaire)

**Objectif :**

Créer un patron Singleton en C++.

**Utilité :**

Le patron Singleton (Singleton Pattern) est utilisé pour

**Implémentation :**

**Résultat :**

**Dossier projet :**

src/main.cpp

src/GSingleton.h

src/GSingleton.cpp

win/Makefile

win/cmd\_build.bat

win/cmd\_clean.bat

win/cmd\_compile.bat

win/cmd\_run.bat

win/bin/

win/build/

**src/main.cpp**