Membres : Schaffo Raphaël, Ryter Nils, Vulliemin Kevin

Equipe : Numéro 6

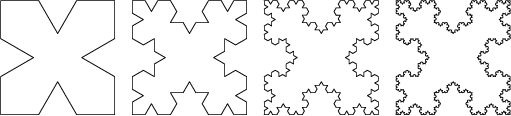
Destinataire : Monsieur Gobron Stéphane

# Introduction

## Problématique

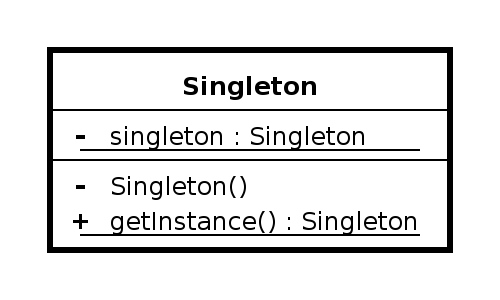
Pour ce laboratoire, nous devons développer un programme qui permet de dessiner des fractales. Ce programme devra contenir le pattern de singleton et le pattern composite. Chaque groupe doit choisir une fractale différente à représenter graphiquement. Nous avons choisi la fractale de cesaro. Pour pouvoir représenter graphiquement cette fractale nous avons décidé d’utiliser Qt Creator.

Qu’est-ce qu’une fractale ? Une fractale est une courbe ou une surface ayant une forme irrégulière ou morcelé. Elle se créer en suivant des règles déterminées. Sur l’image ci-dessous vous pouvez voir la fractale de cesaro. Elle se créer en ajoutant sur chaque ligne un triangle. Sur la première image de gauche, on voit un carré formé de ces 4 lignes sur lesquelles on a ajouté un triangle. On applique cette règle sur la nouvelle forme obtenue et ainsi de suite. Après avoir appliqué 3 fois la règle sur la première forme, on obtient la forme toute à droite.



# Conception

## Singleton

Le singleton est un pattern ayant pour but de limiter, restreindre le nombre d’instanciation d’un objet. Pour implémenter celui-ci il suffit de créer une classe qui va contenir une méthode qui elle, créer une instance d’un objet uniquement si il n’y en a pas encore. S’il y a déjà un objet instancié, la classe renvoie une référence vers cet objet. Le singleton est à manier avec précaution si on l’utilise avec des Threads. Dans le cas où aucune instanciation d’objets n’a encore été faite et que deux threads exécutent la méthode de création il faut bien veiller à ce que seulement un instancie un nouvel objet.

*Diagramme de classes*

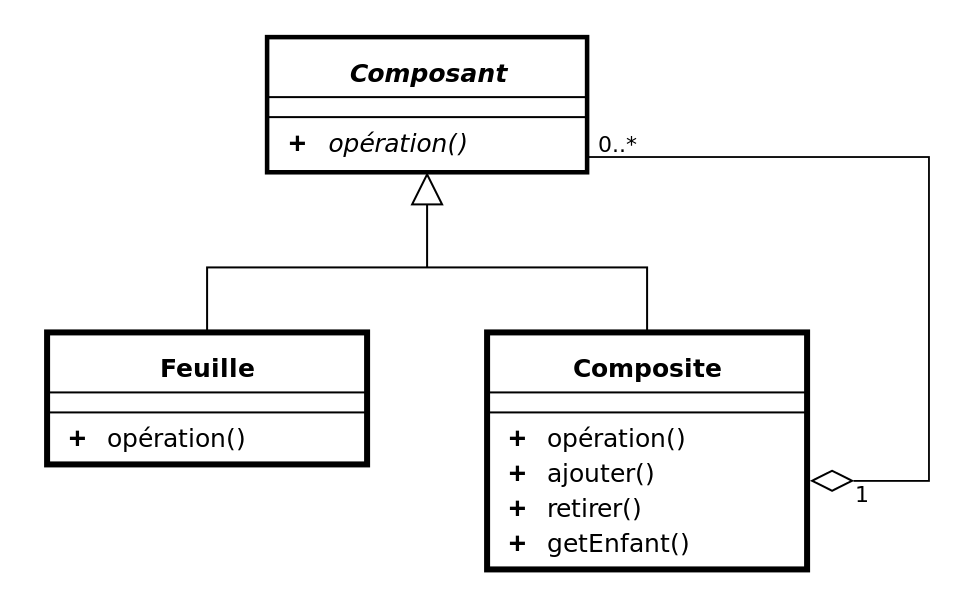
## Composite

Un composite, un objet composite, est un pattern structurel. Il permet de concevoir une structure d’arbre en limitant à deux le nombre de sous éléments. L’idée de ce pattern est enfaite de pouvoir manipuler un groupe d’objet comme si il s’agissait d’un seul objet. Pour cela il faut donc que les objets regroupés aient des opérations communes. Le pattern de composite peut être décomposé en 3 parties (voir le schéma ci-dessous).

***Le Composant :*** Il représente tous les composants, y compris ceux qui sont composés. Il déclare l’interface pour le comportement par défaut.

***La Feuille****:* Elle représente un composant qui ne possède pas de sous-éléments. Les feuilles implémentent le comportement par défaut.

***Le Composite :*** Il représente un composant qui peut avoir des sous-éléments. Il stocke les composants enfant et permet l’accès à ceux-ci. Il implémente le comportement en utilisant son ou ses enfants.



*Diagramme de classes*

Pour manipuler des objets, l’utilisateur, aussi dit client dans ce cas va utiliser la classe composant.

# Implémentation

## Singleton

## Composite

# Conclusion