# Patron Composite

## Question 1

### L’intention du patron Composite

Selon les notes de cours, l’intention d’un patron composite est de « traiter les objets individuels et les objets multiples, composés récursivement, de façon uniforme ». En d’autres mots, un patron composite permet de gérer des objets ayant des fonctions similaires de la même manière en ignorant les différences entre les classes. Ce patron augmente l’uniformité et la flexibilité du code.

### Structure du patron Composite dans PolyIcone3D

Dans PolyIcone3D, les classes Objet3DAbs, Objet3DComposite ainsi que PrimitiveAbs et ses dérivés forment le patron Composite. Le role du *component* est associé à la classe Objet3Dabs étant donné que c’est la classe abstraite dont toutes les autres classes du patron héritent. C’est l’interface uniforme. Le role de *leaf* est associé à la classe abstraite PrimitiveAbs et à ses dérivées. Quant au role du *composite*, il est associé à la classe Objet3DComposite, car c’est celle-ci qui peut être composé d’un ou de plusieurs objets de classe Objet3DAbs.

## Question 2

### Identification des abstractions

Dans cette structure, il y a deux abstractions, soit la classe Objet3DAbs et la classe PrimitiveAbs. La responsabilité principale d’Objet3DAbs est de regrouper et d’uniformiser les fonctionnalités des classes Objet3DComposite des classes héritantes de PrimitiveAbs. Cette abstraction regroupe des fonctions telles que addChild() et cbegin() qui permettent la gestion des enfants dans la classe Objet3DComposite, des fonctions telles que getParameters() et getNbParameters() qui permettent la gestion des paramètres d’une primitive (classe qui dérive de PrimitiveAbs) et des fonctions telles que clone(), getCenter() et setCenter() qui sont des fonctionnalités communes aux classes Objet3DComposite, PrimitiveAbs et aux dérivées de PrimitiveAbs.

# Patron Decorator

## Question 1

### L’intention du patron Decorator

### Structure du patron Decorator dans PolyIcone3D

## Question 2

### Identification des classes primitives

## Question 3

### Pourquoi le Decorator s’applique aux primitives et non à tout objet 3D

# Conteneur et Patron Iterator

## Question 1

### L’intention du patron Iterator

### Identifier la classe de conteneur et les classes des Iterators

## Question 2

### Expliquez le rôle de l’attribut statique m\_emptyContainer

## Question 3

### Conséquence du changement de la classe de conteneur

## Question 4

### Surcharge des opérateurs \* et ->