

LOG2410 : Conception logicielle

Conception à base de patrons I

TP4

Présenté par :

Stéphanie Mansour (1935595)

Nanor Janjikian (1901777)

Groupe : B1

Hiver 2019

Question 1 :

1. Patron Composite :
   1. Intention du patron composite :

Porte sur la structure du design permet de donner uniformité au traitement d’éléments composites. Ce patron permet de concevoir une structure d’arbre, soit des arbres a n sous-éléments. Un objet composite, en programmation d’objet, est constitué d’un ou de plusieurs objets similaires (avec des fonctionnalités similaires). Un patron composite est utilisé pour manipuler un groupe d’objets de la même façon que s’il s’agissait d’un seul objet. Les objets ainsi regroupés doivent posséder des opérations communes, c'est-à-dire un "dénominateur commun".

* 1. Structure :
     1. **Composant**
        1. est l'abstraction pour tous les composants, y compris ceux qui sont composés
        2. déclare l'interface pour le comportement par défaut
     2. **Feuille**
        1. représente un composant n'ayant pas de sous-éléments
        2. implémente le comportement par défaut
     3. **Composite**
        1. représente un composant pouvant avoir des sous-éléments
        2. stocke des composants enfants et permet d'y accéder
        3. implémente un comportement en utilisant les enfants
     4. **Client**
        1. manipule les objets de la composition à travers l'interface de la classe Composant

1. Identifiez toutes les abstractions présentent dans la conception du TP4, et pour chacune, identifiez les responsabilités spécifiques qui lui ont été assignées.

* Abstract : virtuelle pure
  + - AbsAudioFile.h :
    - AbsTransform.h :

1. Dans l’implémentation actuelle du système PolyVersion, quel objet ou classe est responsable de la création de l’arbre des composantes.

* La classe CompositeTransform

Partie 3 :

1. Intention proxy : dupliquer afin dappliquer les modifications sans changer original objet. Agir comme un “bouclier”(intermediaire) pour l’objet et le contrôler.
2. Structure :

Partie 4 :

* 1. Intention patron iterateur : fourni une abstraction sur les objets afin de « masquer » leur comportement interne. => « stabilise » les manipulations.
  2. La classe : vector in (TransformContainer.h)

1. Private : not acces de l’extérieur

Static ne pas modifier