

Design Pattern Composite

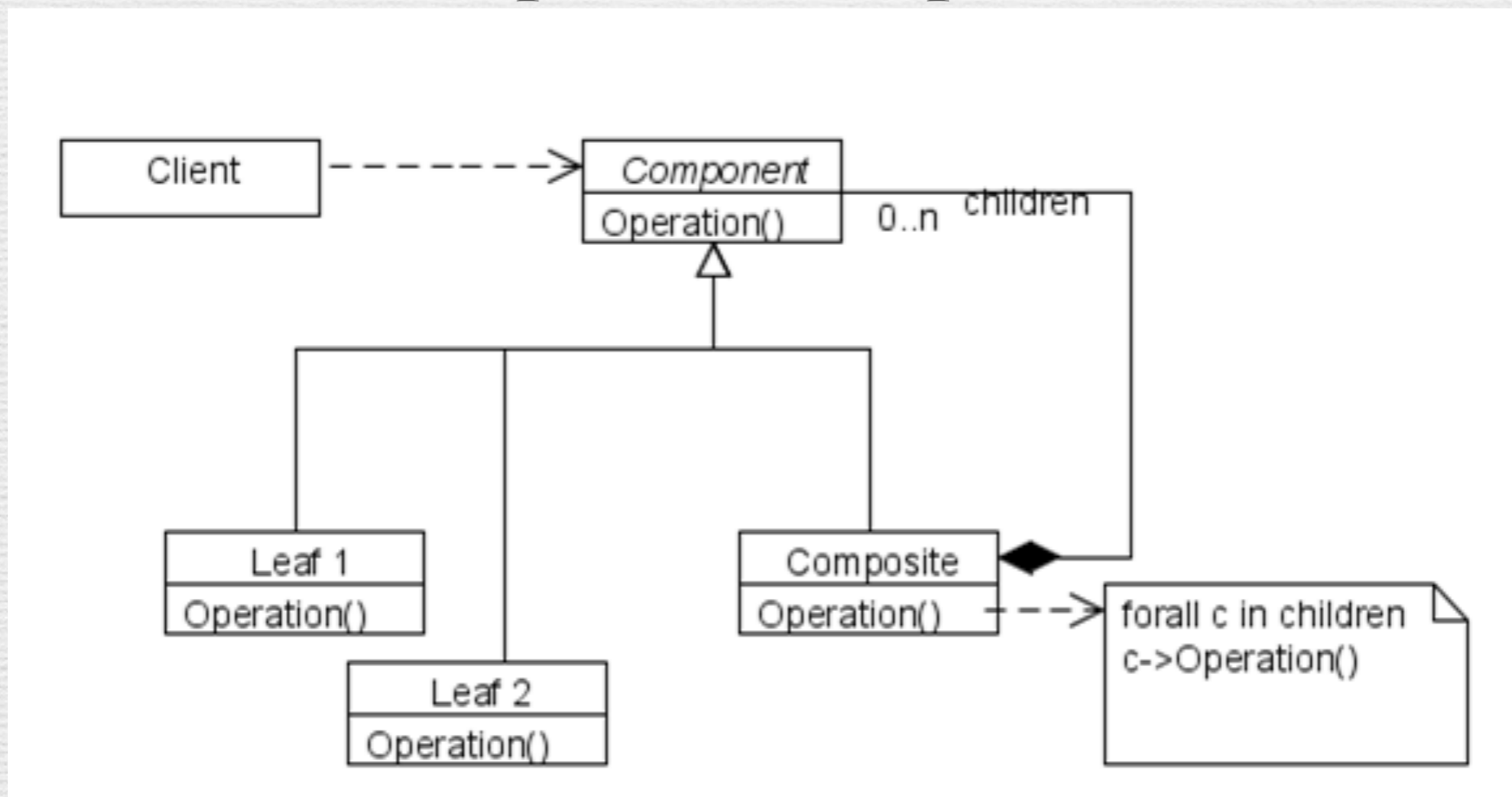
«Pattern Composite» : le problème

✓ Problème :

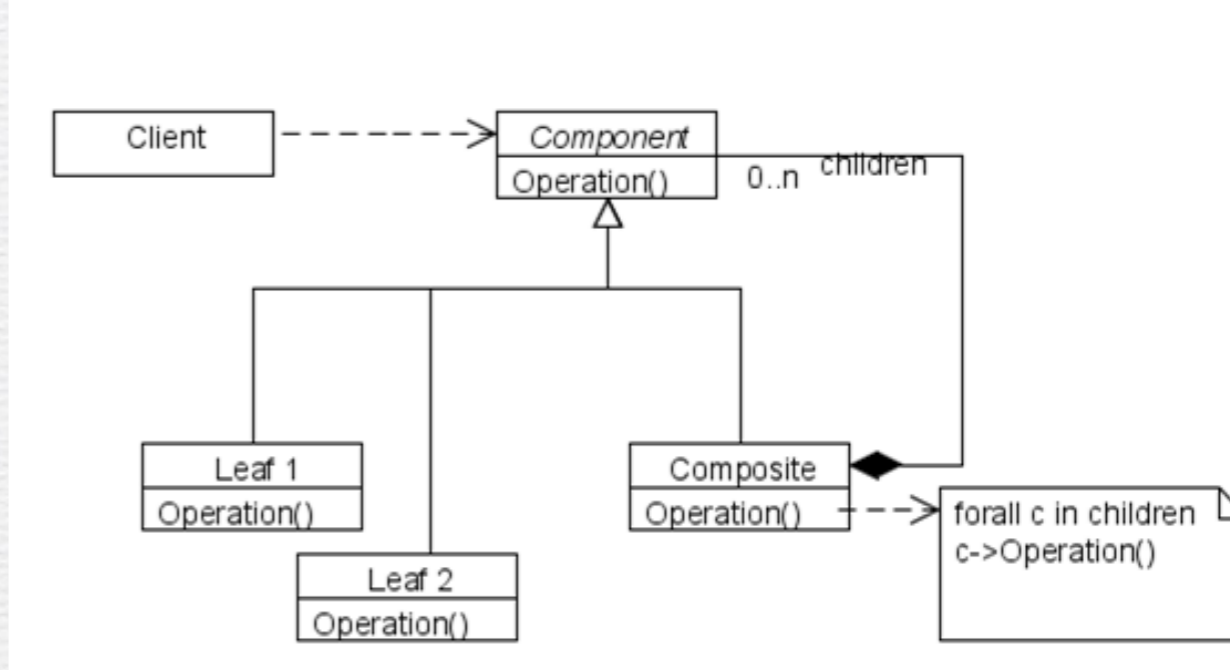
- ➡ Une application doit manipuler des collections d'objets dont certains sont «primitifs» et d'autres «composites».
- ➡ Ces objets doivent tous répondre à une méthode dont le comportement est différent selon que l'objet est composite ou non.

«Pattern Composite» : la solution

- ✓ Organiser les objets dans une structure d'arbre qui capture la hiérarchie «partie-contenant».
- ✓ Le client adresse alors tous les objets de manière uniforme. On parle de composition récursive.



«Pattern Composite» : les rôles



➔Component

- ▶ déclare l'interface des objets pris en compte dans la composition
- ▶ implémente le comportement par défaut des composants autant que possible

➔Composite

- ▶ définit le comportement des composants ayant des «enfants» dans la hiérarchie de la composition
- ▶ référence les composants fils (ils peuvent eux-même être composites!)

➔Leaf

- ▶ définit le comportement des objets primitifs dans la composition

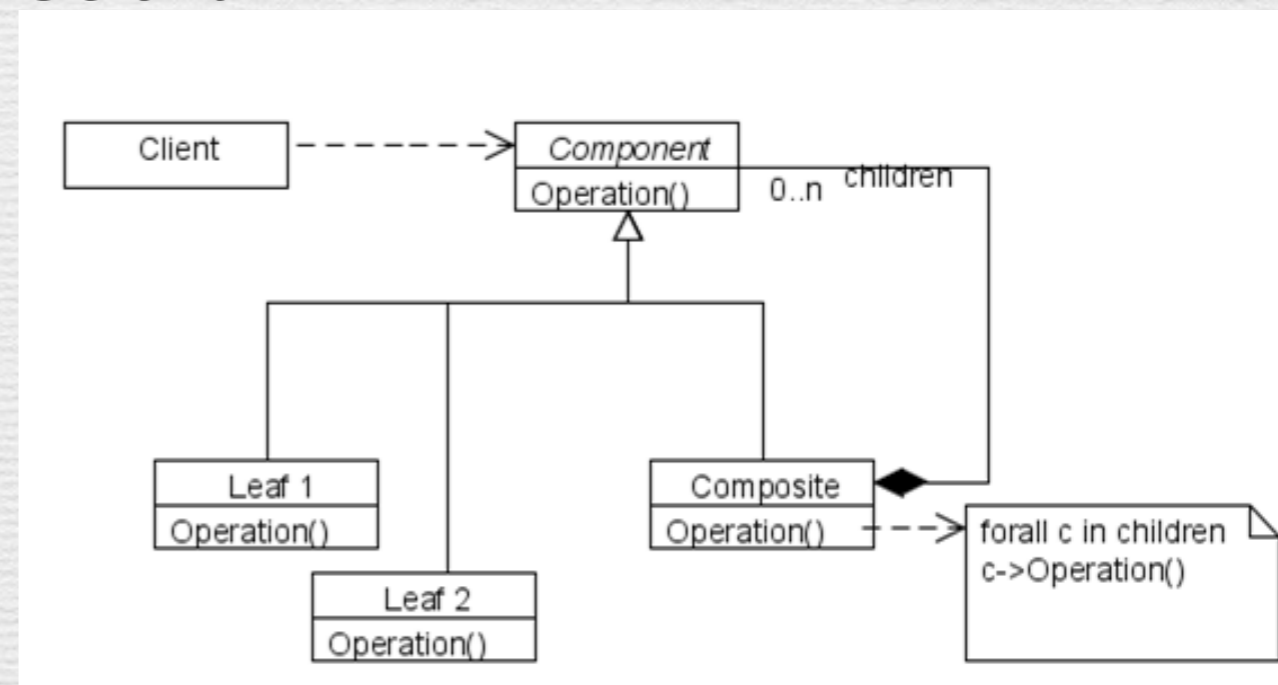
➔Client

- ▶ manipule les objets de la composition au travers de l'interface Component

Design Pattern Composite en action

- ✓ Notre entreprise vend des produits. Chaque produit a un prix, une référence et une description. Nous vendons des jeux videos. A certaines périodes de l'année nous vendons les jeux par lots. Le prix du lot est alors la somme des prix des jeux dans le lot réduite de 10%.
- ✓ Nous construisons notre catalogue comme un ensemble de produits que l'on peut imprimer.
- ✓ Tout produit créé a une référence qui est automatiquement déterminée à la création du produit.

A vous !



DP Composite : Résumé

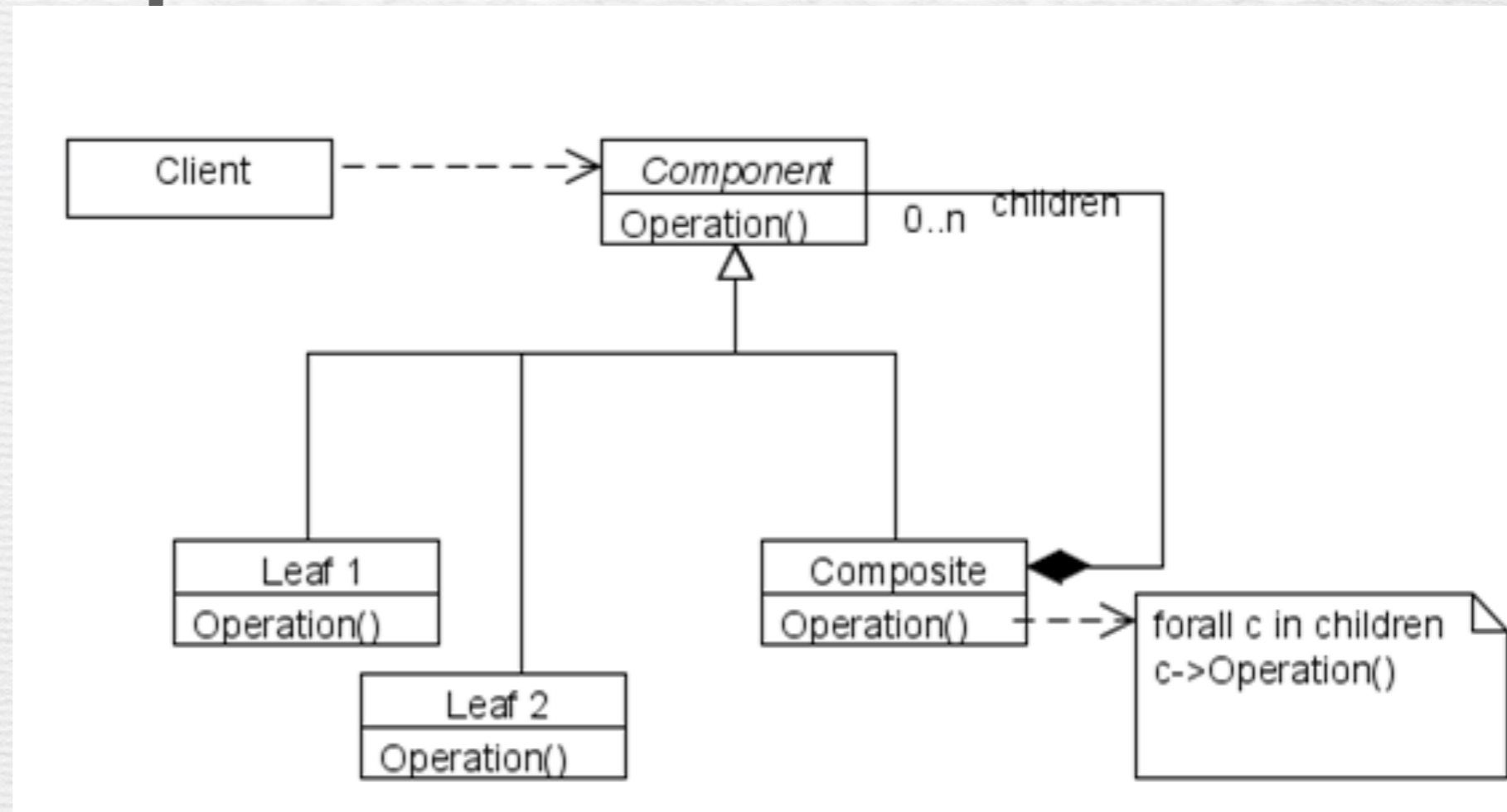
✓ Intention :

- ➔ Compose des objets en des structures arborescentes pour représenter des hiérarchies composant/composé.
- ➔ Permet au client de traiter d'une unique façon les objets et les combinaisons d'objets.

✓ Applicabilité : Utilisez le **Composite** lorsque :

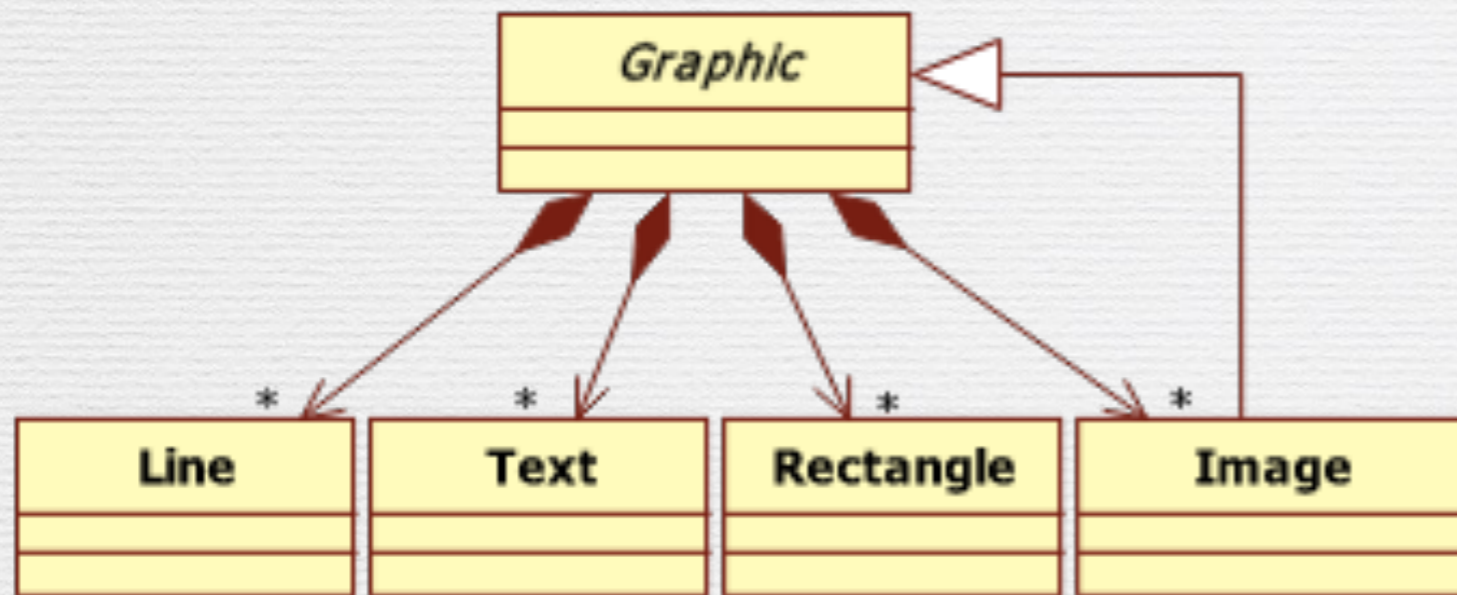
- ➔ Vous souhaitez représenter des hiérarchies de l'individu.
- ➔ Vous souhaitez que le client n'ait pas à se préoccuper de la différence entre "combinaisons d'objets" et "objets individuels". Les clients pourront traiter de façon uniforme tous les objets de la structure composite.

DP Composite : Résumé



✓ Points forts :

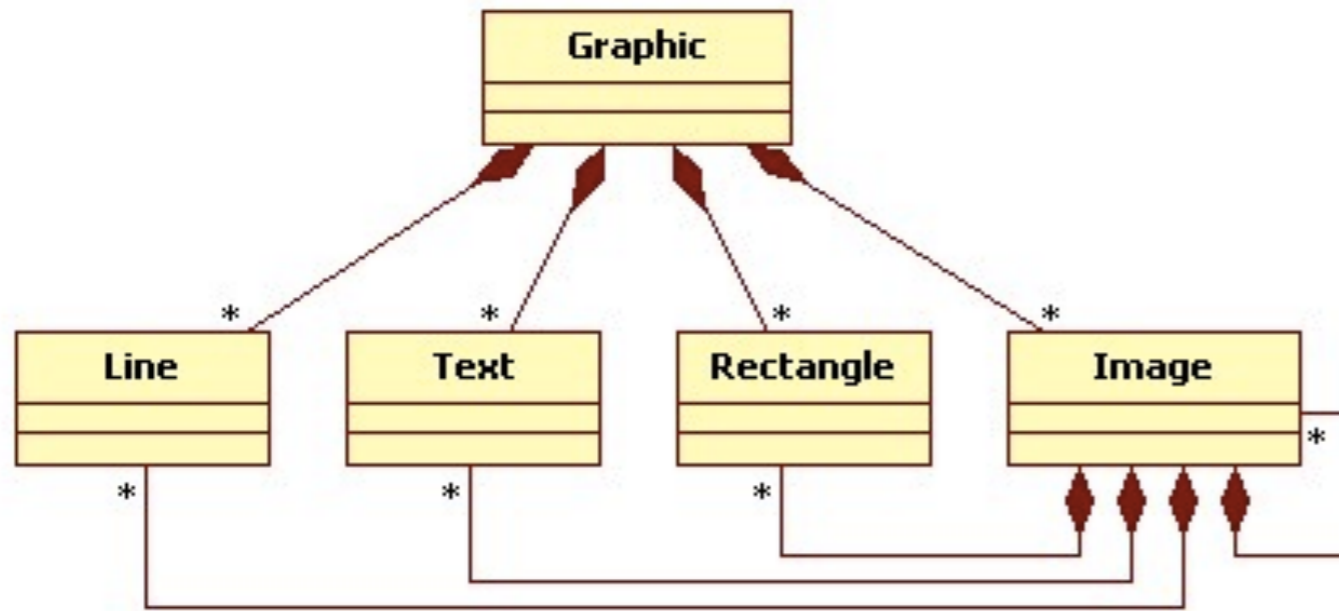
- ✓ 1. Découplage et extensibilité
 - ✓ 1.1 Factorisation maximale de la composition
 - ✓ 1.2 L'ajout ou la suppression d'une feuille n'implique pas de modification de code
 - ✓ 1.3 L'ajout ou la suppression d'un composite n'implique pas de modification de code
- ✓ 2. Protocole uniforme
 - ✓ 2.1 Protocole uniforme sur les opérations des objets composés
 - ✓ 2.2 Protocole uniforme sur la gestion de la composition
 - ✓ 2.3 Point d'accès unique pour la classe client



Quels sont les points faibles de ce modèle?

Modéliser un système permettant de dessiner un graphique. Un graphique est composé de lignes, de rectangles, de textes et d'images, une image pouvant être composée d'autres images, de lignes, de rectangles et de textes.

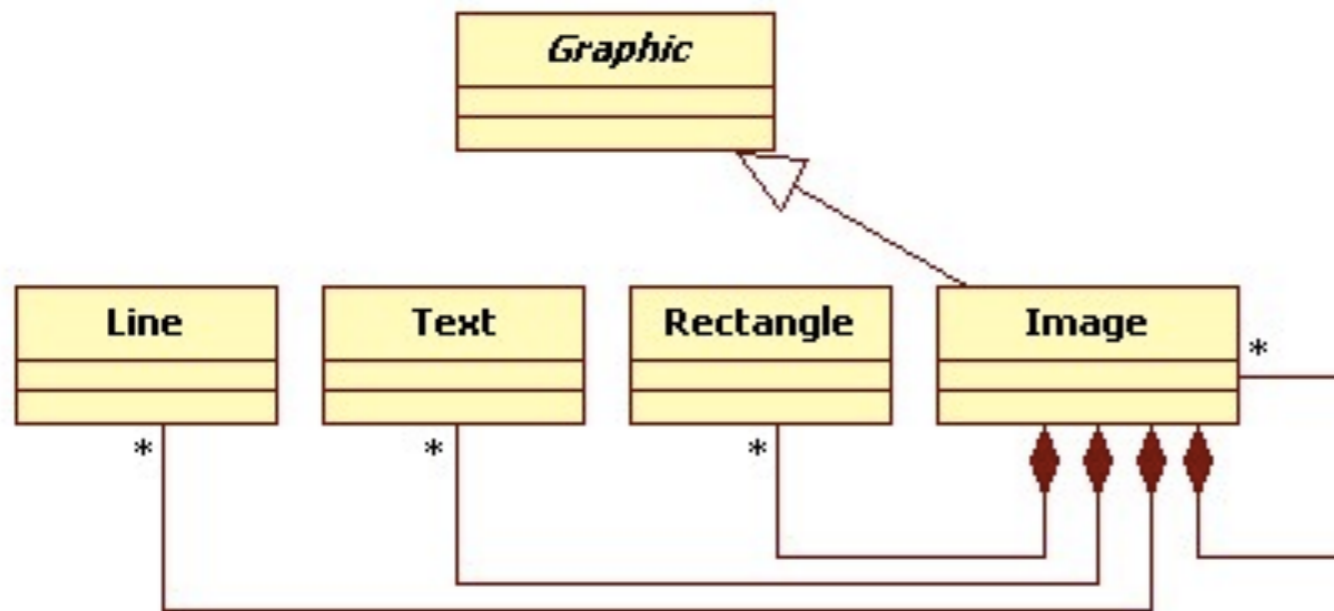
- 1. Découplage et extensibilité
 - 1.1 Factorisation maximale de la composition
 - 1.2 L'ajout ou la suppression d'une feuille n'implique pas de modification de code
 - 1.3 L'ajout ou la suppression d'un composite ne n'implique pas de modification de code
- 2. Protocole uniforme
 - 2.1 Protocole uniforme sur les opérations des objets composés
 - 2.2 Protocole uniforme sur la gestion de la composition
 - 2.3 Point d'accès unique pour la classe client



Quels sont les points faibles de ce modèle?

Modéliser un système permettant de dessiner un graphique. Un graphique est composé de lignes, de rectangles, de textes et d'images, une image pouvant être composée d'autres images, de lignes, de rectangles et de textes.

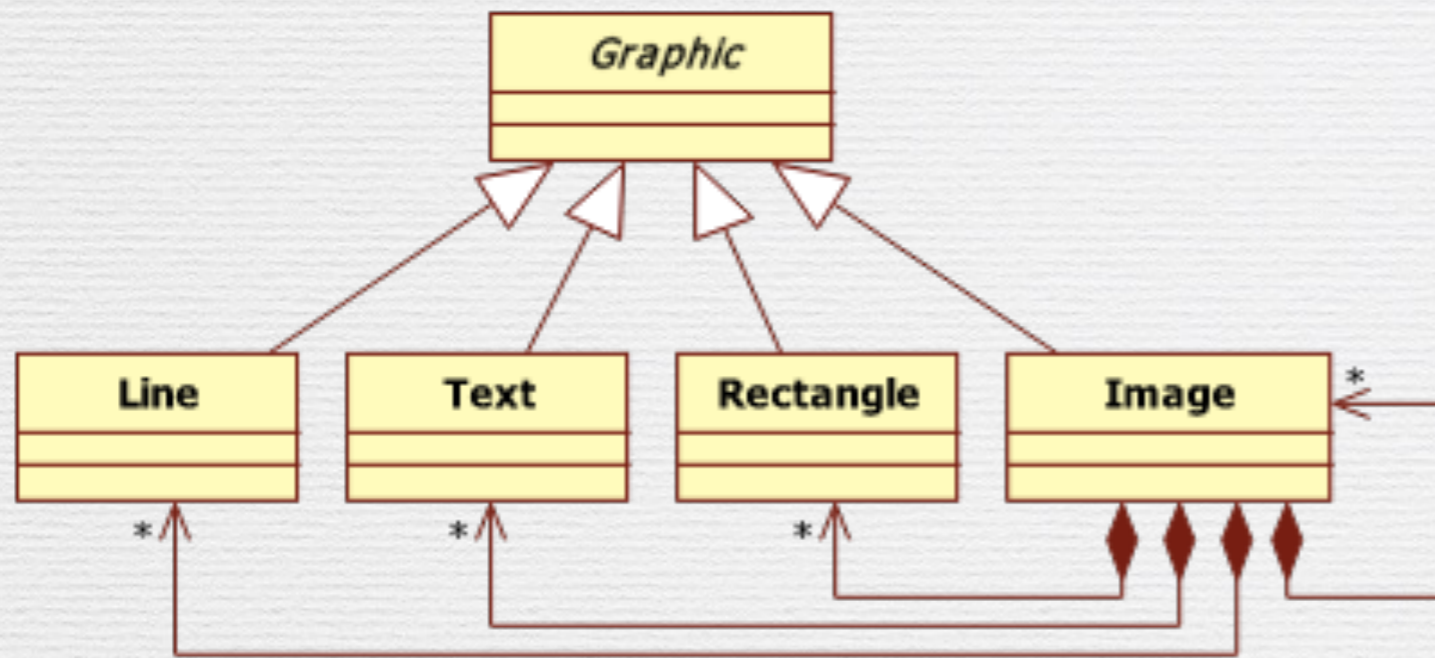
- 1. Découplage et extensibilité
 - 1.1 Factorisation maximale de la composition
 - 1.2 L'ajout ou la suppression d'une feuille n'implique pas de modification de code
 - 1.3 L'ajout ou la suppression d'un composite ne n'implique pas de modification de code
- 2. Protocole uniforme
 - 2.1 Protocole uniforme sur les opérations des objets composés
 - 2.2 Protocole uniforme sur la gestion de la composition
 - 2.3 Point d'accès unique pour la classe client



Quels sont les points faibles de ce modèle?

Modéliser un système permettant de dessiner un graphique. Un graphique est composé de lignes, de rectangles, de textes et d'images, une image pouvant être composée d'autres images, de lignes, de rectangles et de textes.

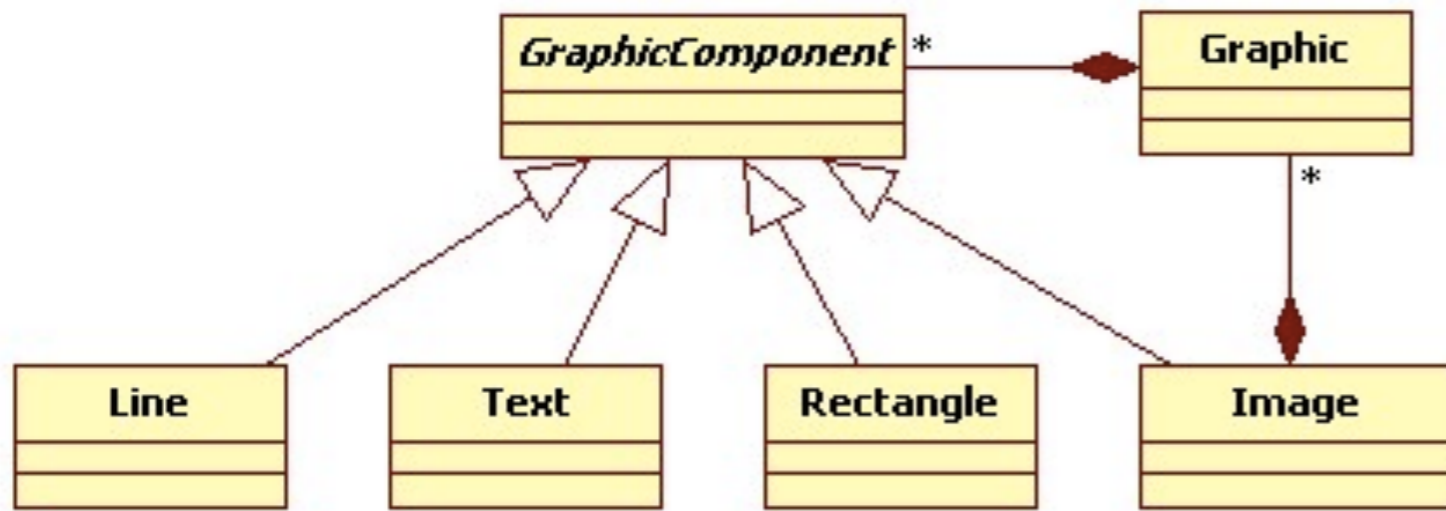
- 1. Découplage et extensibilité
 - 1.1 Factorisation maximale de la composition
 - 1.2 L'ajout ou la suppression d'une feuille n'implique pas de modification de code
 - 1.3 L'ajout ou la suppression d'un composite ne n'implique pas de modification de code
- 2. Protocole uniforme
 - 2.1 Protocole uniforme sur les opérations des objets composés
 - 2.2 Protocole uniforme sur la gestion de la composition
 - 2.3 Point d'accès unique pour la classe client



Quels sont les points faibles de ce modèle?

Modéliser un système permettant de dessiner un graphique. Un graphique est composé de lignes, de rectangles, de textes et d'images, une image pouvant être composée d'autres images, de lignes, de rectangles et de textes.

- 1. Découplage et extensibilité
 - 1.1 Factorisation maximale de la composition
 - 1.2 L'ajout ou la suppression d'une feuille n'implique pas de modification de code
 - 1.3 L'ajout ou la suppression d'un composite ne n'implique pas de modification de code
- 2. Protocole uniforme
 - 2.1 Protocole uniforme sur les opérations des objets composés
 - 2.2 Protocole uniforme sur la gestion de la composition
 - 2.3 Point d'accès unique pour la classe client



Quels sont les points faibles de ce modèle?

Modéliser un système permettant de dessiner un graphique. Un graphique est composé de lignes, de rectangles, de textes et d'images, une image pouvant être composée d'autres images, de lignes, de rectangles et de textes.

- 1. Découplage et extensibilité
 - 1.1 Factorisation maximale de la composition
 - 1.2 L'ajout ou la suppression d'une feuille n'implique pas de modification de code
 - 1.3 L'ajout ou la suppression d'un composite ne n'implique pas de modification de code
- 2. Protocole uniforme
 - 2.1 Protocole uniforme sur les opérations des objets composés
 - 2.2 Protocole uniforme sur la gestion de la composition
 - 2.3 Point d'accès unique pour la classe client