

Correction examen

Questions de cours (4 points)

- 0.5 1) Cette notion est l'encapsulation. En effet, certains attributs et méthodes ont pour but des traitements internes à l'objet et ne doivent pas être accessibles en dehors de celui-ci. Ces attributs et méthodes encapsulés sont dits privés.
- 0.5 2) Oui, l'agrégation est un cas particulier d'association associant un objet complexe aux objets qui le constituent.
- 0.5 3) Oui, la composition est une agrégation vérifiant des contraintes supplémentaires. 0.5
- 0.5 4) Le polymorphisme : signifie que la même méthode peut avoir des comportements différents dans des classes différentes
- 01 5) Une classe concrète possède des instances tandis qu'une abstraite ne peut pas en posséder.
- 01 6) Lors de l'envoi d'un message synchrone, l'émetteur du message attend le retour du destinataire avant de continuer sa propre activité, c'est-à-dire qu'il attend le message de retour émis par le destinataire. Dans le cas d'un message asynchrone, cette attente n'existe pas. Il n'y a alors pas message de retour.

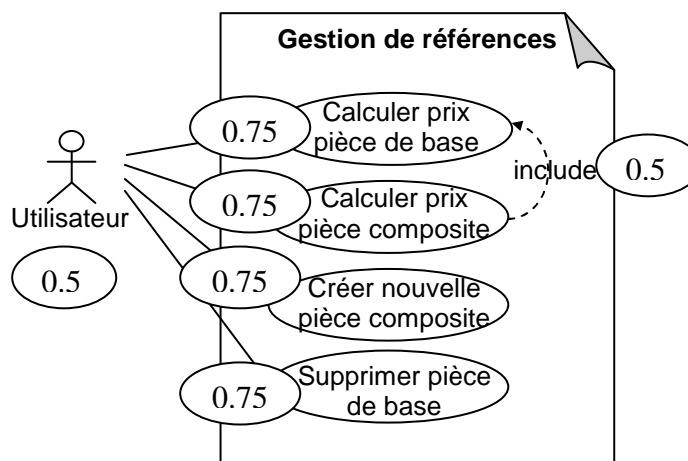
Remarques de correction :

Nous vous rappelons de plus que les solutions proposées ne sont pas uniques. La cohérence et la pertinence de vos diagrammes ainsi que le respect de la syntaxe UML comptent pour beaucoup dans votre note.

Exercice 1 – Diagramme des cas d'utilisation (4 points)

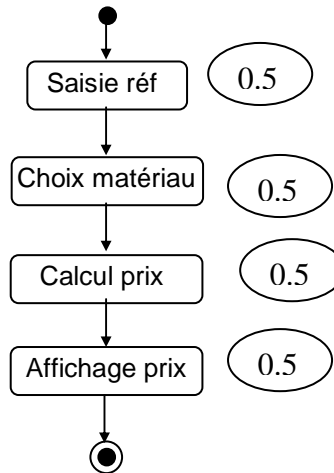
On rappelle que le diagramme des cas d'utilisation permet de représenter les interactions entre le système à développer et le monde extérieur.

Dans notre cas le seul acteur est l'utilisateur et les cas d'utilisation identifiables sont « calculer prix pièce de base », « calculer prix pièce composite », « créer nouvelle pièce composite » et « supprimer pièce de base ».

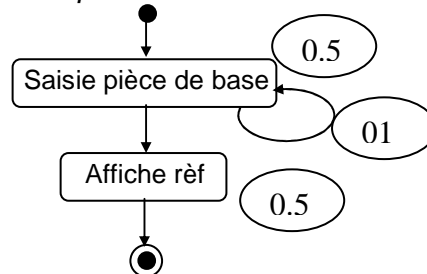


Exercice N°2 - Diagramme d'activité (04 points)

Pour le cas « Calculer prix pièce de base » :



Pour le cas « Créer nouvelle pièce composite » :



Exercice 3 – Diagrammes de séquence (8 points)

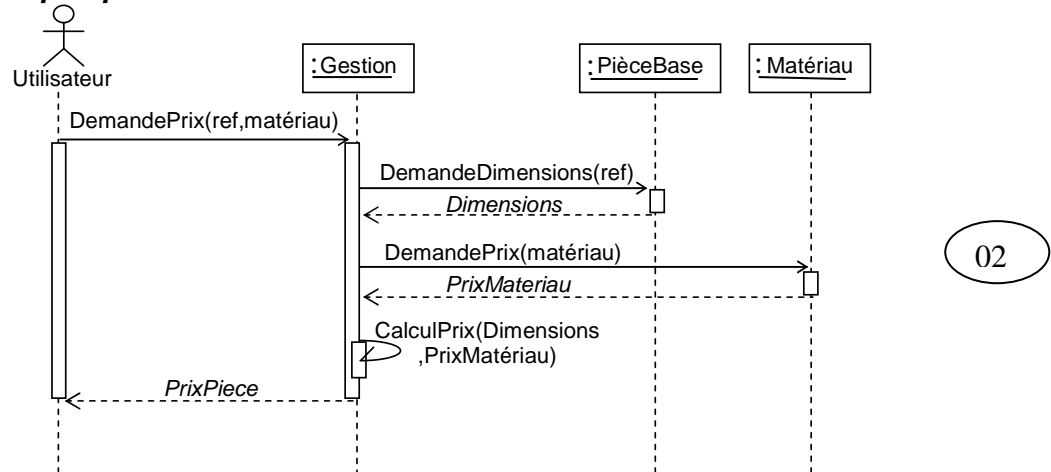
Dans la question précédente nous avons identifiés 4 cas d'utilisation. Il faut donc réaliser 4 diagrammes de séquence différents. Avant de donner les diagrammes il faut identifier quelles sont les classes qui appartiennent au système.

On peut identifier les classes suivantes :

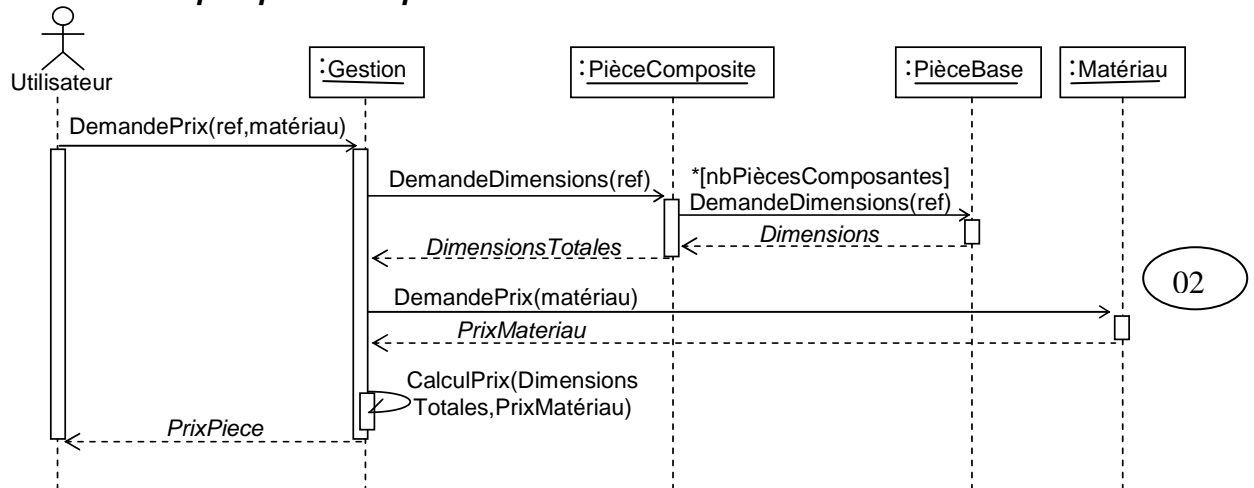
- une classe Pièce dont hérite la classe PièceBase et la classe PièceComposite, (0.5)
- une classe Matériau (0.5)
- une classe Gestion qui représente le cœur du système de gestion (0.5)
- un acteur Utilisateur (0.5)

Une fois ses classes identifiées on peut représenter les différents cas d'utilisation sous forme de diagramme de séquence.

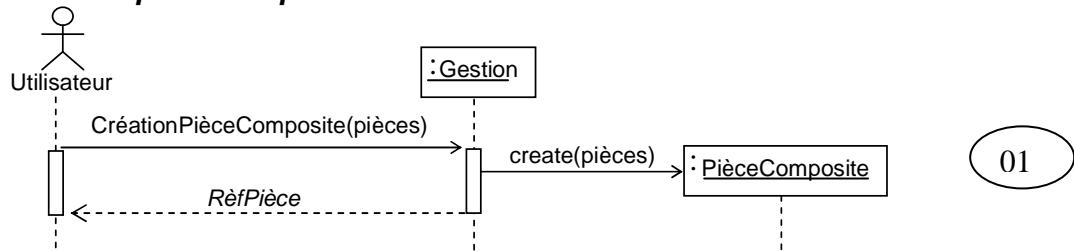
Cas « Calculer prix pièce de base »



Cas « Calculer prix pièce composite »

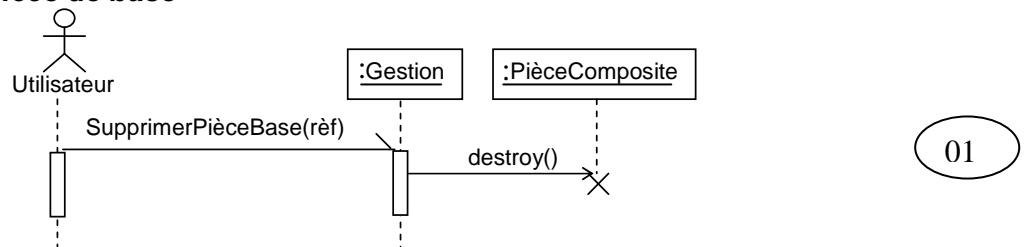


Cas « Créer nouvelle pièce composite »



Notez que ce n'est pas la classe *PièceComposite* qui est créée ici mais une instance de cette classe.

Cas « Supprimer pièce de base »



Notez que ce n'est pas la classe *PièceBase* qui est détruite ici mais une instance de cette classe.