

UML : Langage de modélisation objet unifié



Cours n°10 : Diagramme de paquetages

1) Notion de paquetage

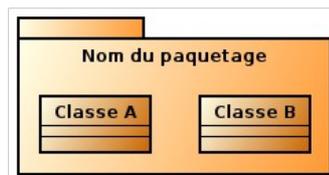
Lorsque nous sommes en présence d'un système de grande taille, il peut être intéressant de le décomposer en plusieurs parties (appelées paquetage).

Un paquetage est donc un regroupement de différents éléments d'un système (regroupement de classes, diagrammes, fonctions, interfaces...). Cela permet de clarifier le modèle en l'organisant. Il est représenté par un dossier avec son nom à l'intérieur :

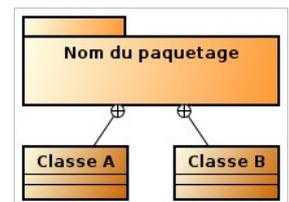


Il est possible de représenter les éléments du système appartenant au paquetage :

à l'intérieur de celui-ci :



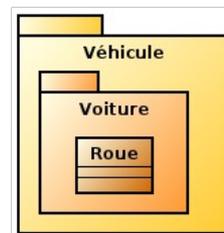
ou à l'extérieur :



Pour faire appel à un élément d'un paquetage, nous indiquons le nom du paquetage (espace de nommage) suivi de deux fois deux points (::) puis du nom de l'élément.



→ *Personne::Client*



→ *Véhicule::Voiture::Roue*

Les paquetages peuvent s'imbriquer (décomposition hiérarchique) mais pas se chevaucher. Un élément du système ne peut appartenir qu'à un et un seul paquetage. Chaque paquetage doit posséder un nom différent.

2) Rôle du diagramme de paquetages :

Le diagramme de paquetages est un diagramme structurel (statique) d'UML qui **représente** les **paquetages** (ou espaces de noms) composant un système, ainsi que **les relations** qui lient ces différents paquetages.

3) Dépendances entre paquetages :

3-1) Visibilité :

Chaque éléments d'un paquetage est soit :

- **privé**, c'est-à-dire encapsulé dans le paquetage et invisible à l'extérieur de celui-ci. Un élément privé est désigné par un signe - devant lui.
- **public**, c'est-à-dire visible et accessible de l'extérieur du paquetage. Un élément public est désigné par un signe + devant lui.



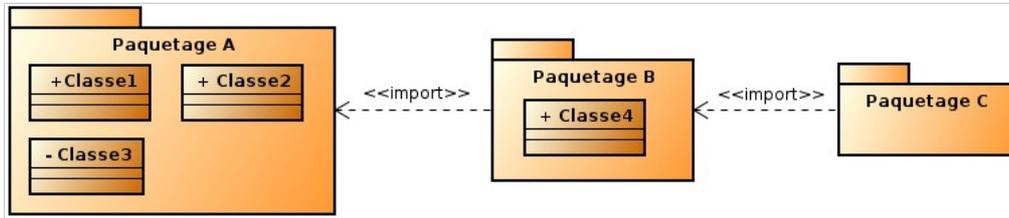
Par défaut, les éléments d'un paquetage sont publics.

3-2) Dépendance de type « import » :

Elle correspond à l'importation par un paquetage **B** de tous les éléments **publics** d'un paquetage **A**. Ces éléments :

- auront la visibilité « **public** » dans le paquetage **B** (et seraient donc aussi transmis à un paquetage **C** qui ferait une importation du paquetage **B**).
- seront accessibles au paquetage **B** sans avoir à utiliser explicitement le nom du paquetage **A**.

La dépendance de type « **import** » est représentée par une flèche pointillée muni du stéréotype **<<import>>**.

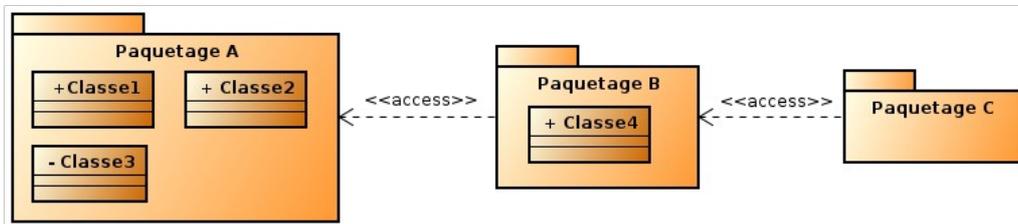


Le paquetage B importe **Classe1** et **Classe2** (pas **Classe3** qui a une visibilité de type **privée**). **Classe1** et **Classe2** ont une visibilité de type **public** dans paquetage B. Le paquetage C importe **Classe1**, **Classe2** et **Classe4**.

3-3) Dépendance de type « access » :

Elle correspond à l'accès par un paquetage B de tous les éléments publics d'un paquetage A. Ces éléments auront la visibilité **privé** dans le paquetage B, ils ne peuvent donc pas être transmis à un paquetage C qui ferait une importation ou un accès au paquetage B (pas de transitivité).

La dépendance de type « **access** » est représentée par une flèche pointillée muni du stéréotype **<<access>>**.

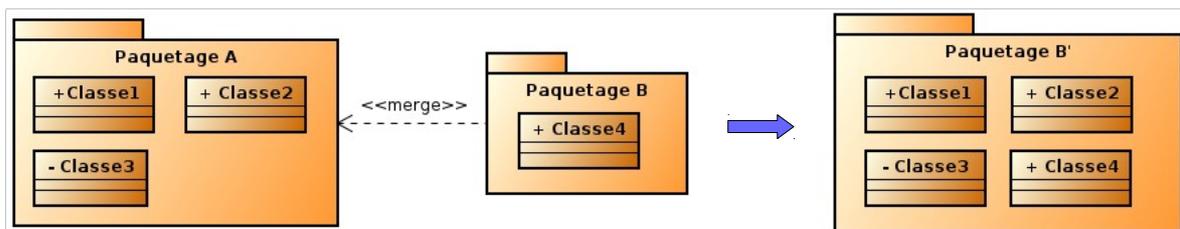


Le paquetage B a accès à **Classe1** et **Classe2** (pas à **Classe3** qui a une visibilité de type **privée**). **Classe1** et **Classe2** ont une visibilité de type **privé** dans paquetage B. Le paquetage C a accès à **Classe4** (pas à **Classe1** et **Classe2** qui ont une visibilité de type **privé** dans paquetage B).

3-4) Dépendances de type « merge » :

Elle correspond à la fusion de 2 paquetages en un seul.

La dépendance de type « **merge** » est représentée par une flèche pointillée muni du stéréotype **<<merge>>**.



Le paquetage A est fusionné dans le paquetage B (le paquetage A n'est pas modifié alors que le paquetage B est écrasé pour accueillir la fusion des 2 paquetages).