

PARTIE 2

DIAGRAMMES DE STRUCTURE

2 DIAGRAMMES DE STRUCTURE

Les diagrammes de structure permettent de visualiser, spécifier, construire et documenter l'aspect statique du système.

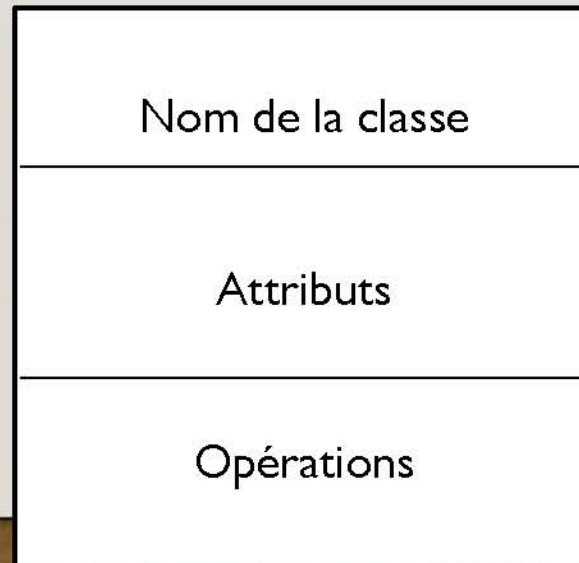
- Diagramme de classe
- Diagramme d'objet
- Diagramme de paquetage
- Diagramme de composant
- Diagramme de déploiement
- Diagramme de profil

3 DIAGRAMME DE CLASSE

- Le diagramme de classe est Utilisé pour capturer la structure statique du modèle conceptuel du système, en décrivant ses classes et les relations statiques entre elles.
- Le diagramme de classe montre un ensemble de classes, d'interfaces et de relations.
- Le diagramme de classe est considéré comme une fondation pour les diagrammes de composant et de déploiement.

4 ELÉMENTS DU DIAGRAMME DE CLASSE

- La classe est un modèle pour créer des objets, produire la spécification des attributs et des opérations d'une instance de la classe.
- Les classes sont graphiquement représentées par des boites avec compartiments pour le nom de la classe, les attributs et les opérations et la capacité des propriétés, responsabilités, règles etc....



5 ELÉMENTS DU DIAGRAMME DE CLASSE

- **Interface:** spécifie un contrat qui consiste en un ensemble cohérent d'attributs publiques et d'opérations pour la classe. N'importe quelle instance d'une classe qui réalise l'interface doit respecter le contrat.

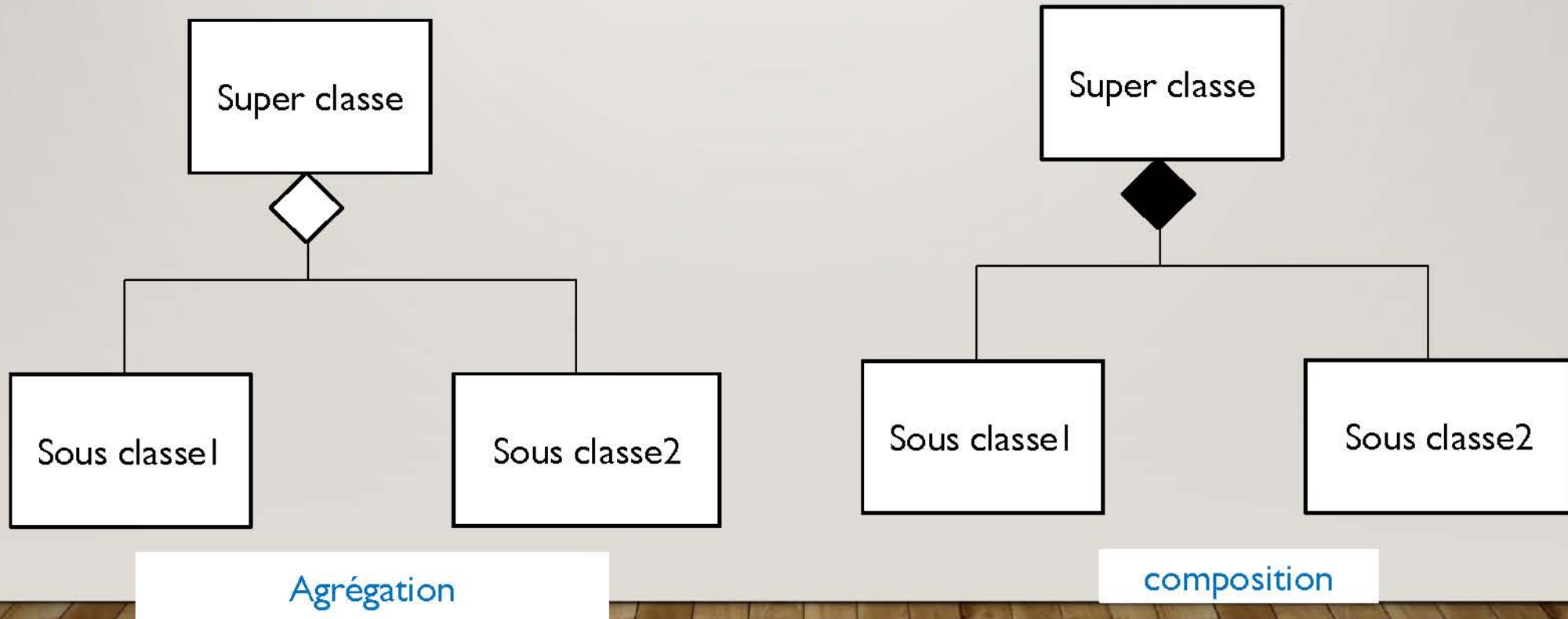
6 VISIBILITÉ

La visibilité contrôle l'accès aux fonctionnalités d'une classe. Elle est appliquée aux opérations et aux attributs.

- **Visibilité publique** : Tout élément pouvant accéder à la classe peut accéder à ses fonctionnalités avec une visibilité publique (**disponible à tous les utilisateurs**).
- **Visibilité privée**: Seules les opérations au sein de la classe peuvent accéder aux fonctionnalités avec visibilité privée (**caché à l'utilisation**).
- **Visibilité Protégée**: Uniquement les opérations au sein de la classe ou au sein des filles de la classe, peuvent accéder aux fonctionnalités avec une visibilité protégée (**disponible aux descendants via l'héritage**).
- **Paquetage**: Tout élément qui se trouve dans le même paquetage que le paquetage classe, ou dans un sous-paquetage imbriqué, peut accéder à l'un des ses fonctionnalités avec la visibilité des paquetages.

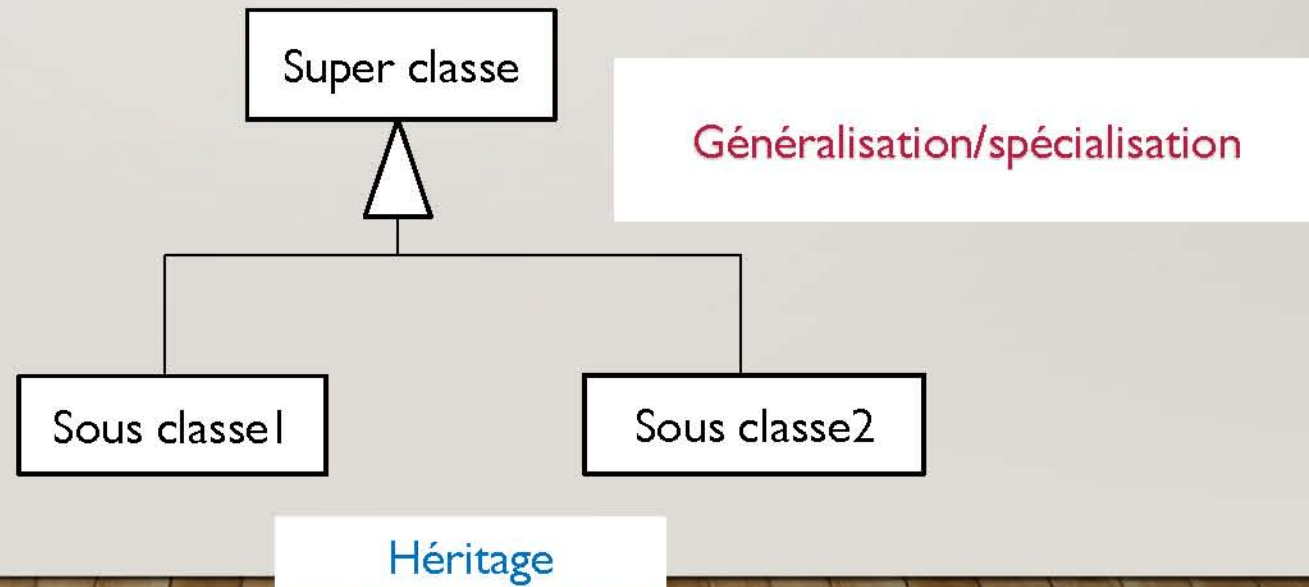
7 ELÉMENTS DU DIAGRAMME DE CLASSE

- **Relation:** c'est un concept qui spécifie quelques types de relations entre les classes.



8 ELÉMENTS DU DIAGRAMME DE CLASSE

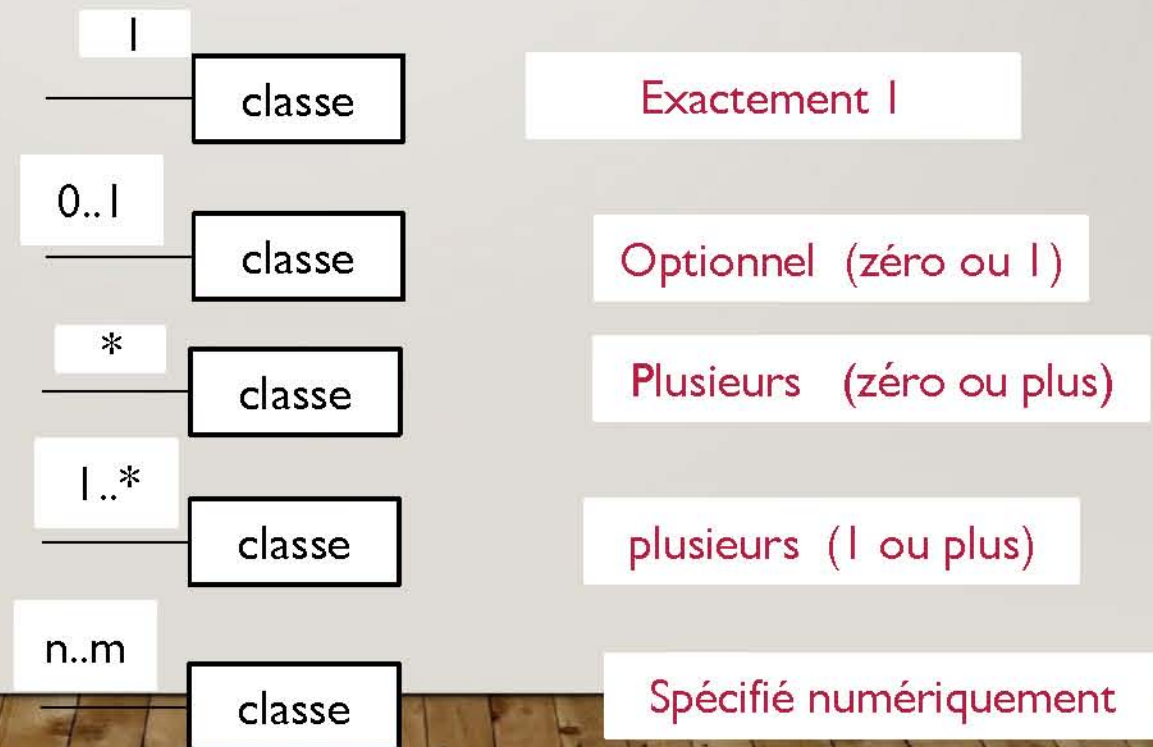
- **Généralisation** : elle relie les classes générales aux classes spécifiques.



ÉLÉMENTS DU DIAGRAMME DE CLASSE

9

- association: c'est une relation structurelle sur les classes montrant la structure physique des objets (par exemple une voiture possède 4 roues). L'association est une relation structurelle et statique entre deux ou plusieurs classes.



10 ELÉMENTS DU DIAGRAMME DE CLASSE

- **Dépendance**: énonce une entité qui utilise les informations et les services d'une autre entité (exemple une voiture utilise une station d'essence pour faire le plein de carburant).
- **Enumérateur**: utilisé pour spécifier un ensemble de valeurs disponibles (exemple **booléen** spécifie un énumérateur avec deux valeurs vrai et faux)

|| TECHNIQUES D'IDENTIFICATION DE CLASSES: ANALYSE VERBE/NOM

La première étape de l'analyse consiste à collecter des informations des sources suivantes :

- Spécification des besoins supplémentaires (s'ils existent),
- Cas d'utilisations,
- Glossaire du projet du projet .

12 TECHNIQUES D'IDENTIFICATION DE CLASSES: ANALYSE VERBE/NOM

Après avoir collecté la documentation, analyser-la de manière très simple en mettant en évidence les:

Noms exemple vol

Phrases nominales exemple numéro de vol

Verbes exemple allouer

Phrases verbales exemple vérifier la carte de crédit

Les noms et les phrases nominales peuvent indiquer des classes ou des attributs.

Les verbes et phrases verbales peuvent indiquer des responsabilités (méthodes) des classes.

13 TECHNIQUES D'IDENTIFICATION DE CLASSES: ANALYSE VERBE/NOM

Remarques:

1. Pour le terme difficile à comprendre durant ce processus, demander des clarifications d'un expert du domaine et ajouter le terme au glossaire.
2. Prendre une liste des noms, phrases nominales, verbes et phrases verbales et utiliser le glossaire du projet pour résoudre les synonymes et homonymes. Cette résolution permet la création d'une liste de classes candidates , attributs et responsabilités.

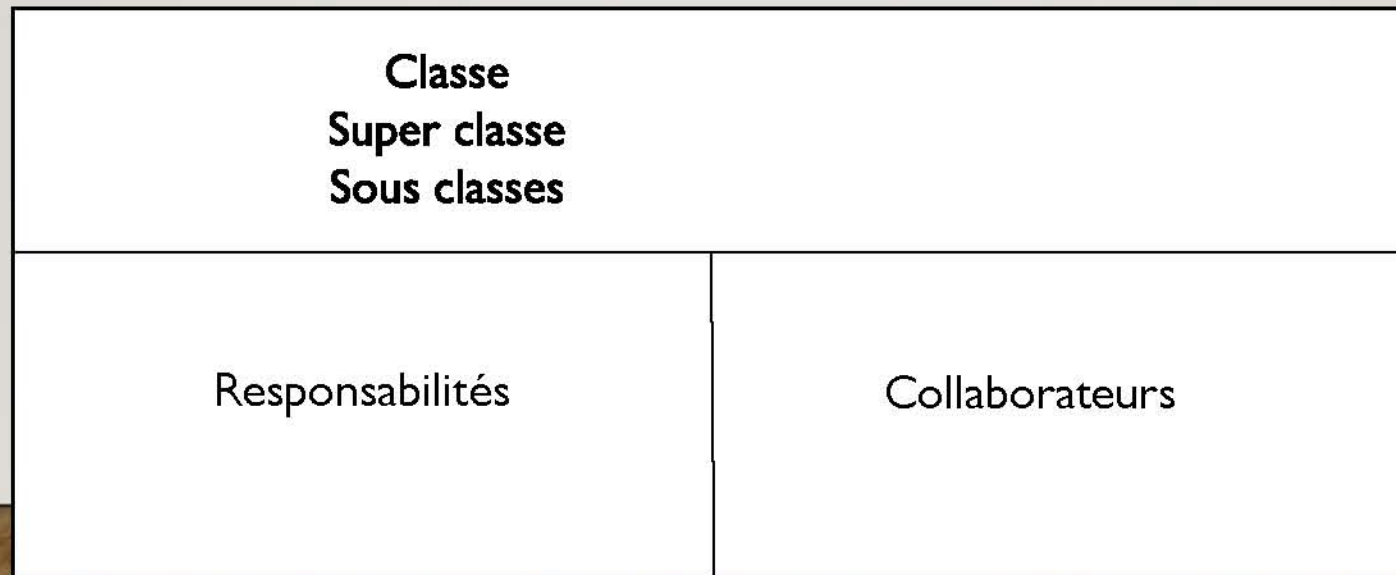
TECHNIQUES D'IDENTIFICATION DE CLASSES: ANALYSE CRC (CLASS, RESPONSABILITÉS, AND COLLABORATION)

Enseignant: BETT A. Mohamed

14

CRC est une technique de **brainstorming** dans laquelle vous capturez les choses importantes dans le domaine du problème sur des notes autocollantes.

Brainstorming est une technique qui consiste à réunir un groupe de collaborateurs afin qu'ils produisent collectivement un maximum d'idées nouvelles sur un thème donné.



15 PROCÉDURE D'ANALYSE CRC

La technique CRC doit être toujours utilisée en conjonction avec l'analyse nom/verbe des cas d'utilisation, besoins, glossaire et autres documents pertinents sauf si le système est très simple.

Phase 1: recueillir les informations (les participants sont des analystes, des parties prenantes et les experts du domaines).

Phase 2: analyser les informations (les participants sont des analystes orientée objet et des experts du domaine).

16 TECHNIQUES D'IDENTIFICATION DE CLASSES : RECHERCHE D'AUTRES RESSOURCES DE CLASSES

Du monde réel, on peut déterminer des classes

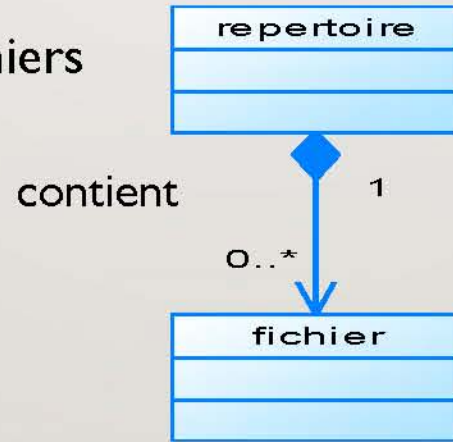
- Objets physiques peuvent être des classes (avion, hôtel etc.)
- La paperasse est une autre source riche de classes (factures, commandes etc.)
- Les Interfaces connues avec le monde extérieur comme les écrans, les claviers, les périphériques et autres systèmes peuvent être source de classes candidates spécialement pour les systèmes embarqués.

17 CRÉATION D'UN PREMIER MODÈLE

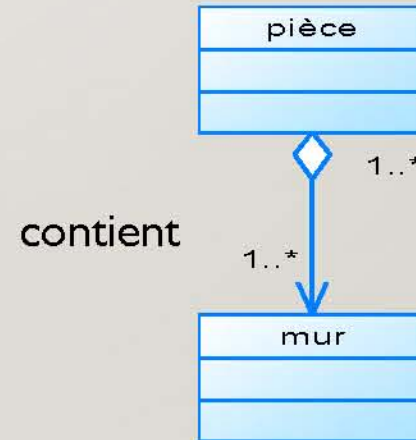
1. Comparer les résultats d'analyse nom/verbe avec les résultats CRC et les résultats des autres sources de classes,
2. Résoudre les synonymes et homonymes,
3. Les différences entre les résultats des différentes techniques montrent les zones d'incertitudes,
4. Consolider les résultats dans un premier modèle d'analyse,
5. Raffiner le modèle avec itération.

18 EXEMPLES DE DIAGRAMMES DE CLASSE

- Un répertoire contient des fichiers

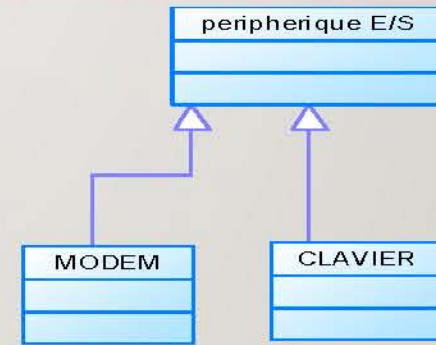


- Une pièce contient des murs

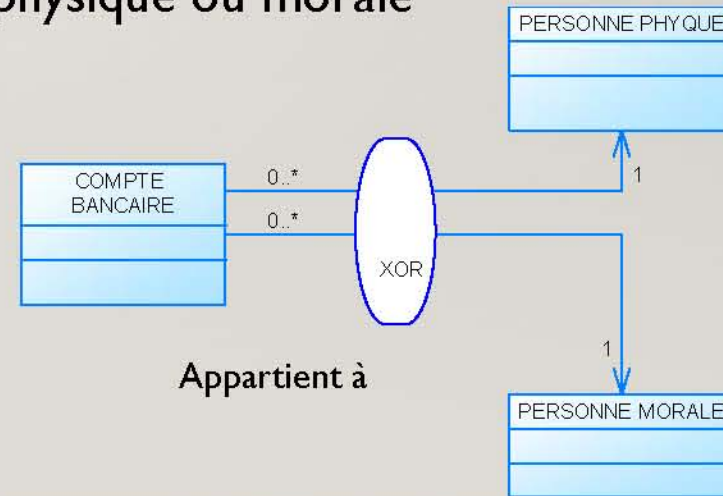


19 EXEMPLES DE DIAGRAMMES DE CLASSE

- Les modems et les claviers sont des périphériques

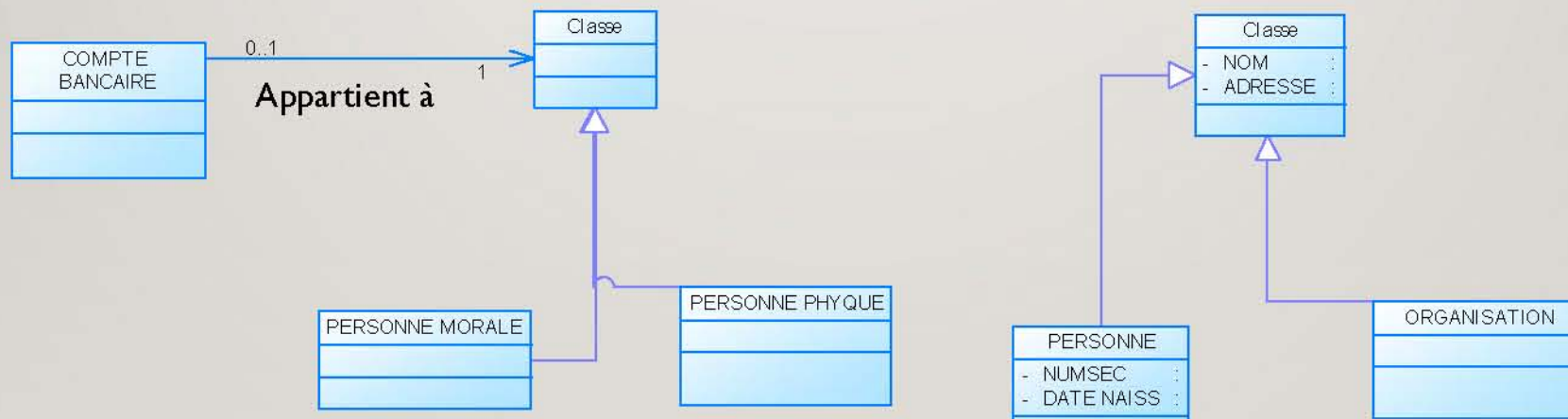


- Un compte bancaire peut appartenir à une personne physique ou morale



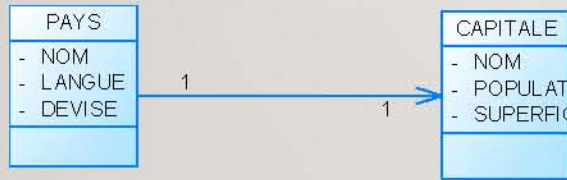
20 EXEMPLES DE DIAGRAMMES DE CLASSE

Utilisation d'une classe abstraite

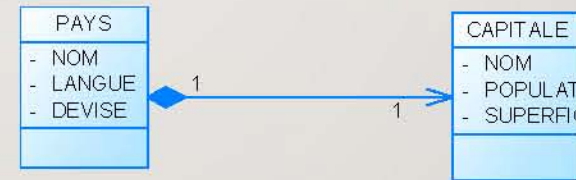


21 EXEMPLES DE DIAGRAMMES DE CLASSE

- Un pays a une capitale



Première solution

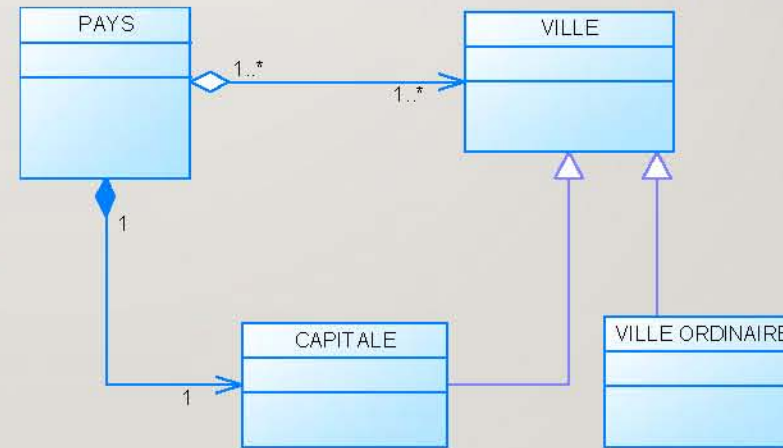
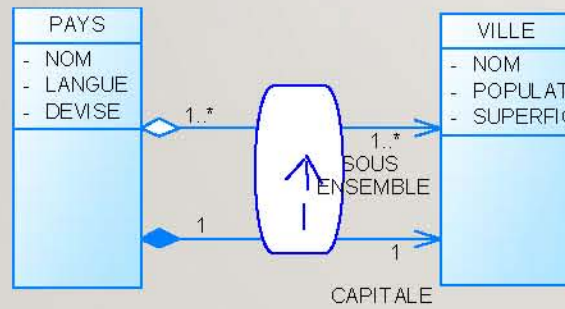


composition



Agrégation

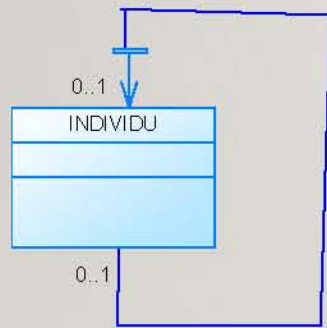
22 EXEMPLES DE DIAGRAMMES DE CLASSE



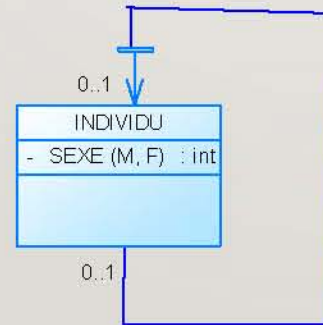
Classe abstraite ville

23 EXEMPLES DE DIAGRAMMES DE CLASSE

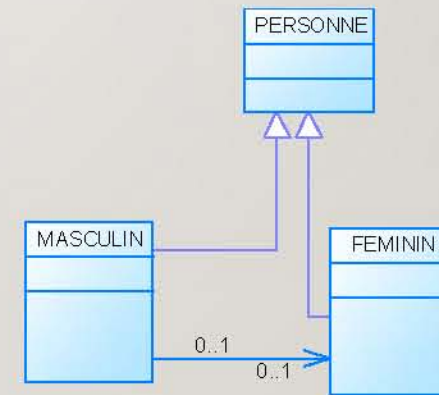
- Deux personnes peuvent être mariées



est marié à



est marié à

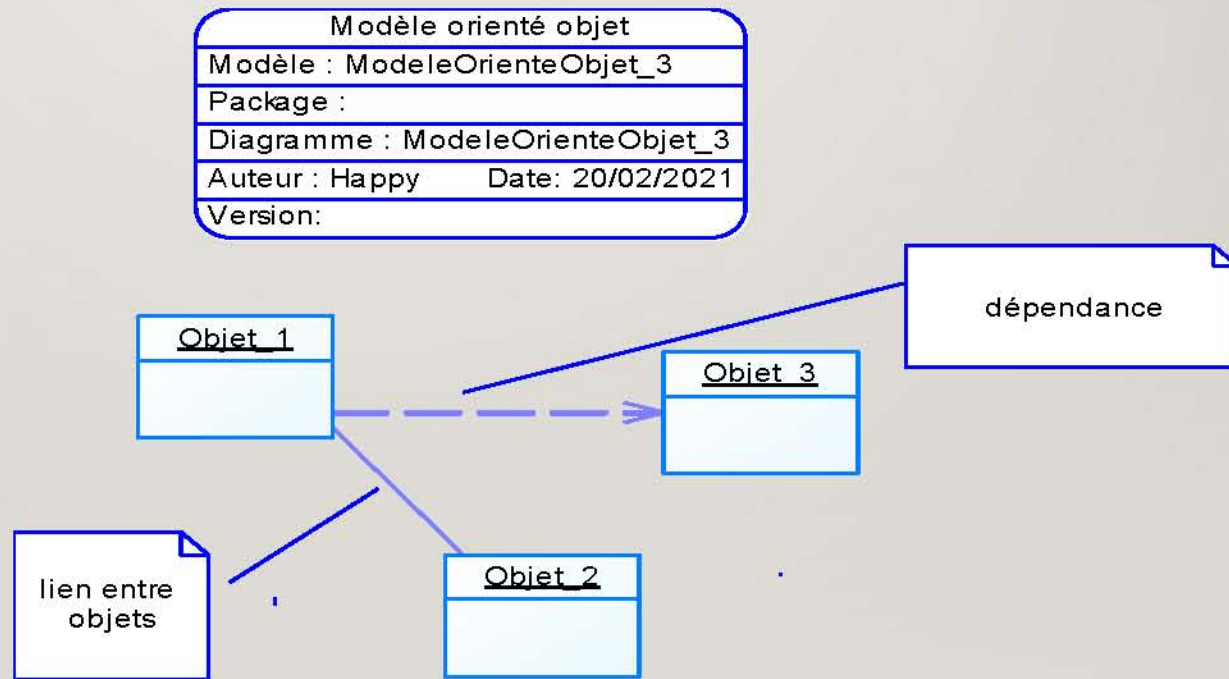


24 DIAGRAMME D'OBJET

- L'objet est une instance d'une classe.
- Un diagramme d'objets est un instantané des objets d'un système à un moment donné, y compris les relations (liens) entre eux.
- Le diagramme d'objets est utilisé chaque fois que cela est nécessaire pour modéliser ou examiner les valeurs des attributs et l'état de l'objet à différentes étapes de l'exécution du logiciel.
- Les diagrammes d'objets sont très utiles pour modéliser l'exécution (étape par étape) (d'un processus complexe ou d'une opération de calcul).

25 DIAGRAMME D'OBJET

Une association dans un diagramme de classe devient un lien dans un diagramme d'objet



26 DIAGRAMME DE PAQUETAGE

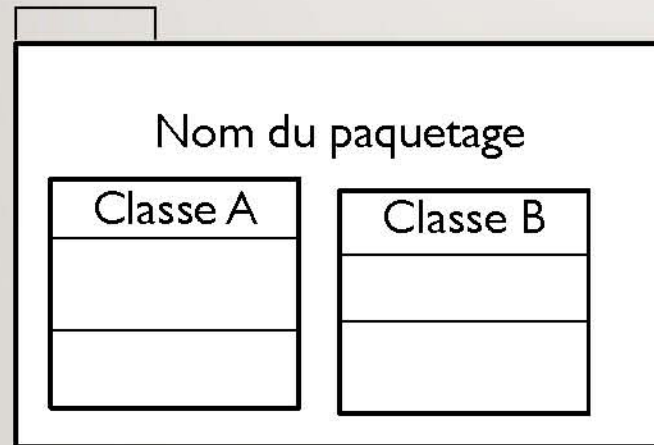
- Le diagramme de paquetage est un regroupement de différents éléments d'un système. Exemples classes, diagrammes, fonctions, interfaces etc.
- Le diagramme de paquetage est un mécanisme à usage général pour organiser les éléments et les diagrammes en groupes.
- Le but des diagrammes de paquetages est de clarifier le modèle en l'organisant.

27 DIAGRAMME DE PAQUETAGE

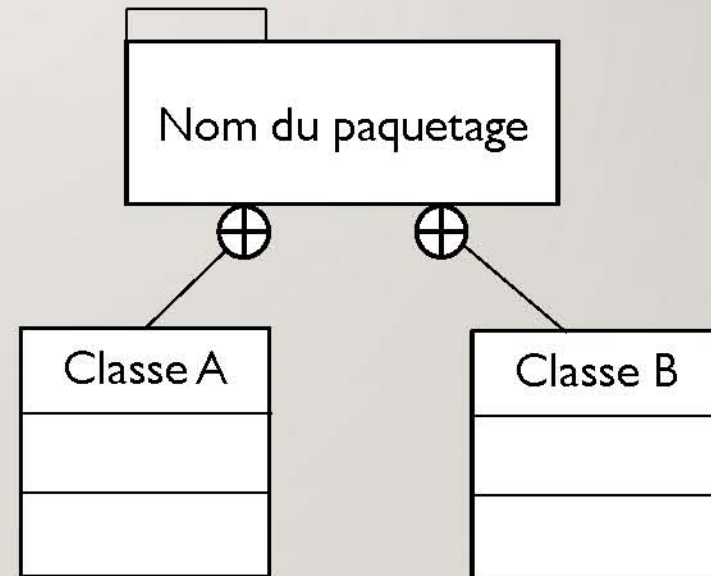
Les diagrammes de paquetage servent à plusieurs usages.

1. Ils regroupent les éléments sémantiquement liés.
2. Ils créent une frontière (frontière sémantique) dans le modèle.
3. Ils fournissent des unités de gestion de configuration.
4. Dans la conception, ils fournissent des unités pour le travail parallèle .
5. Ils fournissent un espace de noms encapsulé dans lequel tous les noms doivent être unique.

28 DIAGRAMME DE PAQUETAGE



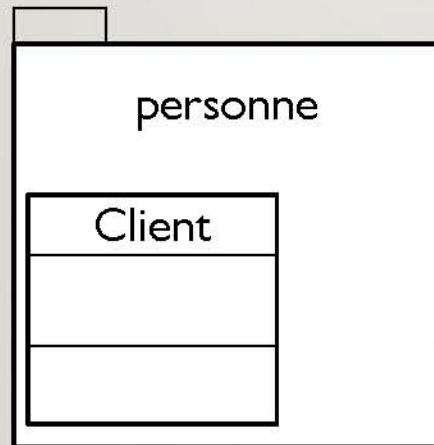
Les classes à l'intérieur



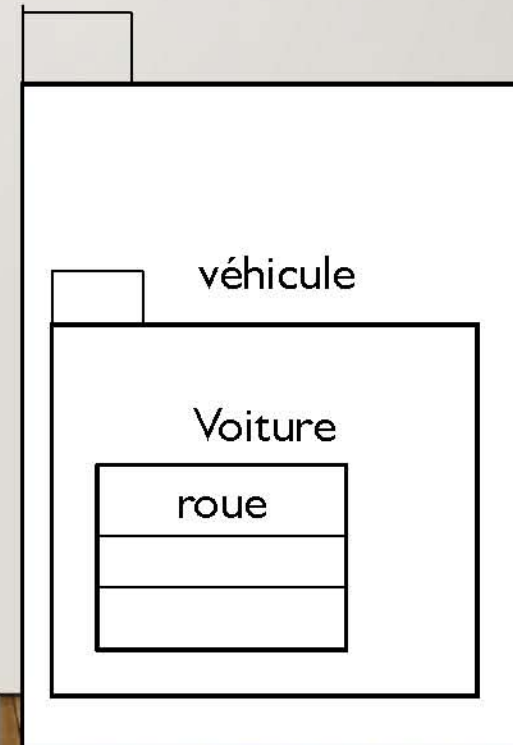
Les classes à l'extérieur

APPEL D'UN ÉLÉMENT D'UN PAQUETAGE

pour accéder à un élément dans l'espace de noms, il faut spécifier le nom de l'élément et le nom de l'espace de noms.



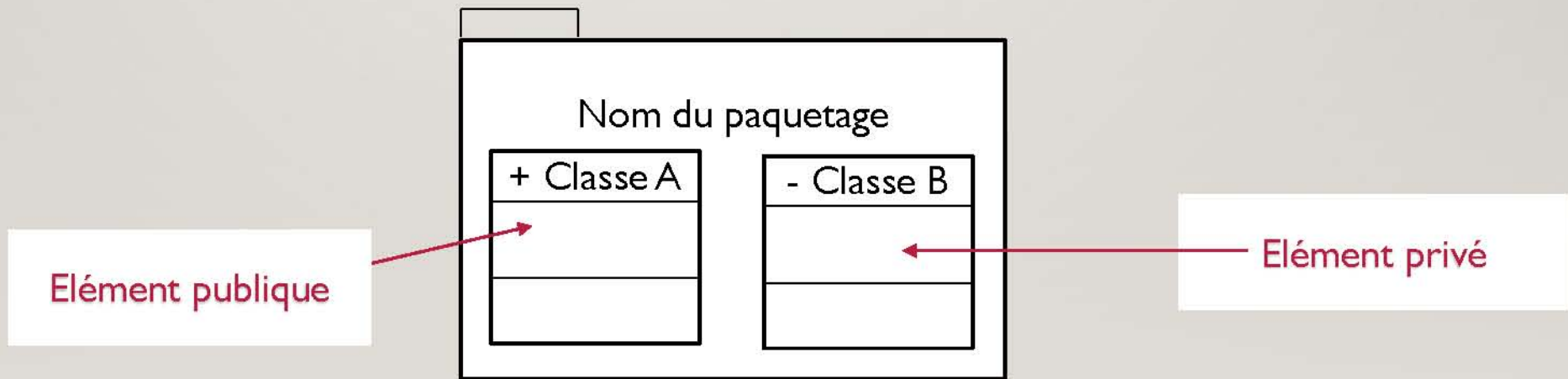
Personne :: client



Véhicule :: voiture :: roue

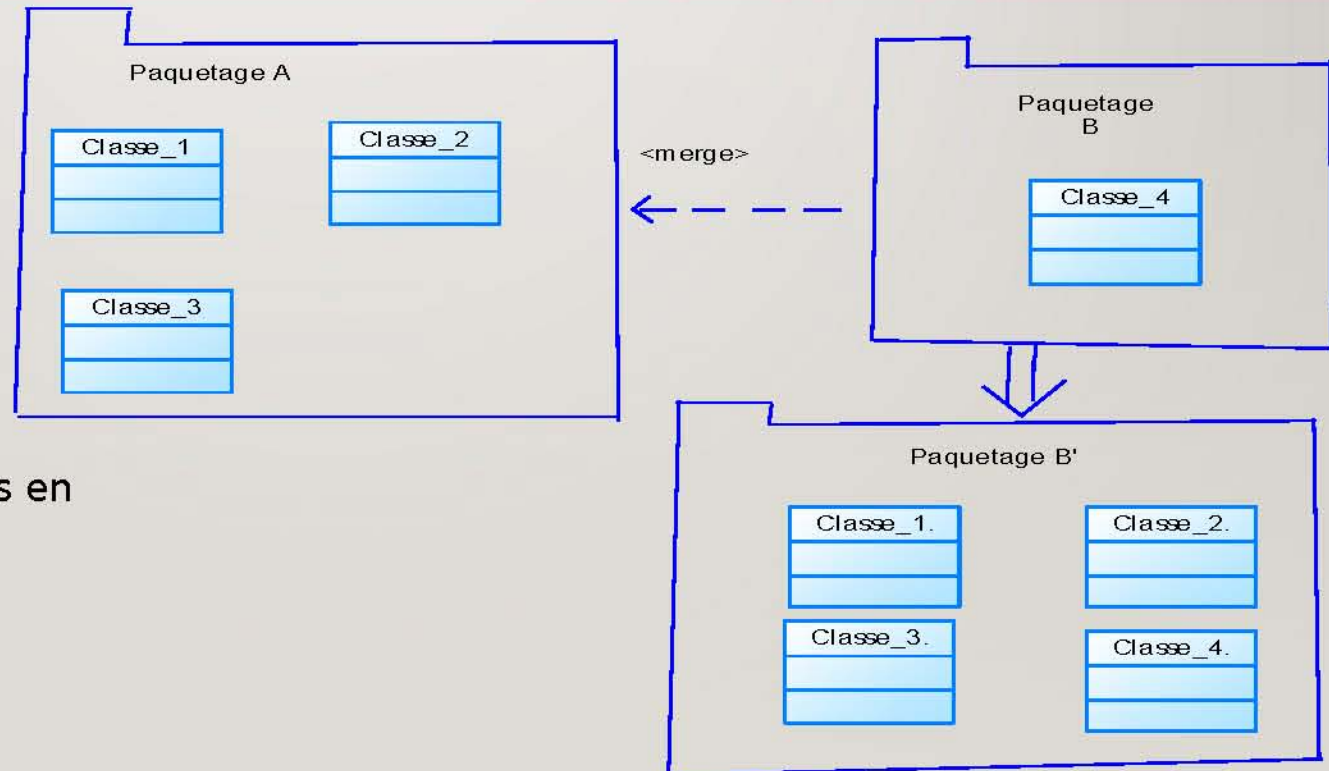
DÉPENDANCES ENTRE PAQUETAGES

- Pour la visibilité, chaque élément d'un paquetage est soit:
 1. Privé (désigné par -), encapsulé dans le paquetage et invisible de l'extérieur.
 2. Publique (désigné par +), visible et accessible de l'extérieur du paquetage.



31

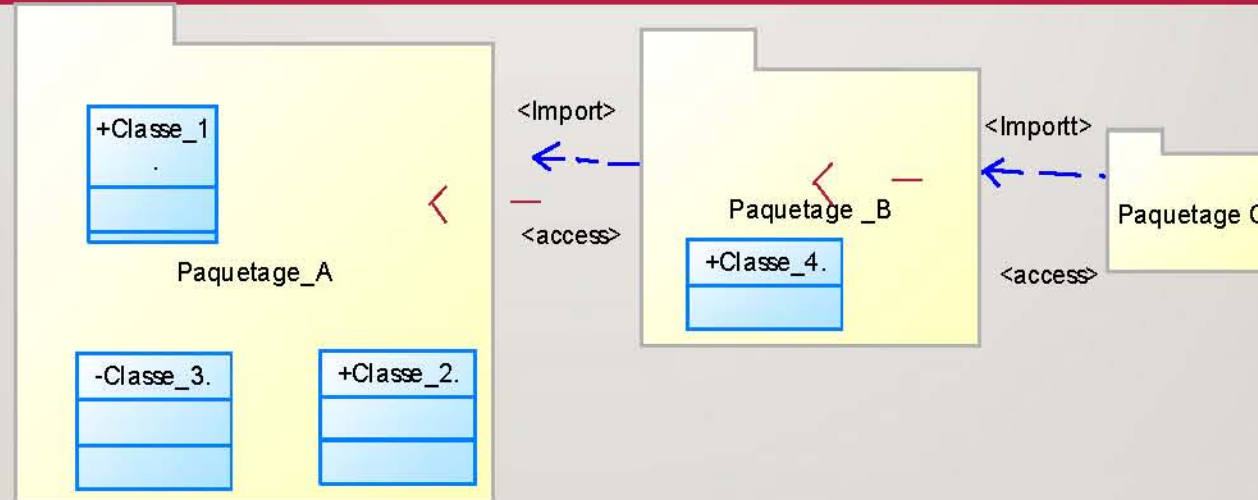
DÉPENDANCE DE PAQUETAGES: TYPE MERGE



Merge = fusion de 2 paquetages en un seul paquetage

A n'est pas modifié
B est écrasé

32 DÉPENDANCE DE PAQUETAGES: TYPE **IMPORT**



B importe classe1 et classe2 (pas classe3 visibilité privée)

C importe classe1, classe2 et classe4)

B a un accès à la classe1 et la classe2 (pas d' accès à la classe3)

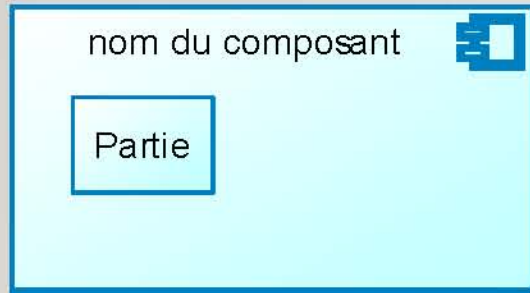
C a accès à la classe4 (pas à la classe 1 et à la classe 2) **pas de transmission**

DIAGRAMME DE COMPOSANT

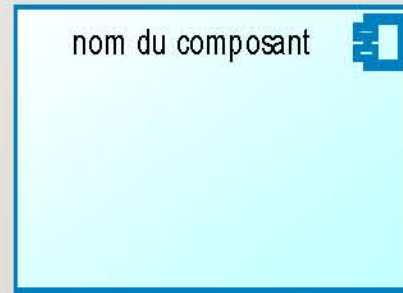
- Le diagramme de composant capture les interactions de haut niveau et les dépendances entre les composants logiciels qui représentent la structure physique de l'implémentation construite comme une partie de la spécification architecturale du système.
- Le diagramme de composant est un élément logiciel remplaçable et réutilisable qui fournit ou reçoit un service bien précis.
- Sa notion est proche de celle d'un objet dans le sens de la modularité et de la réutilisation.
- La granularité peut être différente.
- Le but du diagramme de composant est d'organiser le code source, construire une version exécutable et spécifier la base de données physique..

34 DIAGRAMME DE COMPOSANT

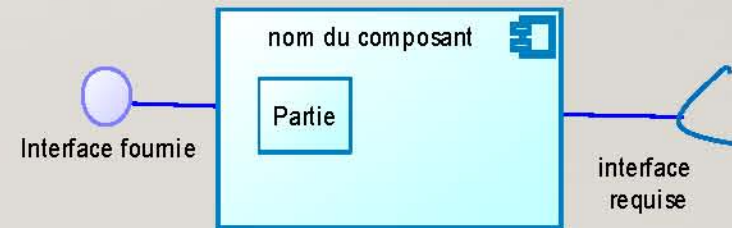
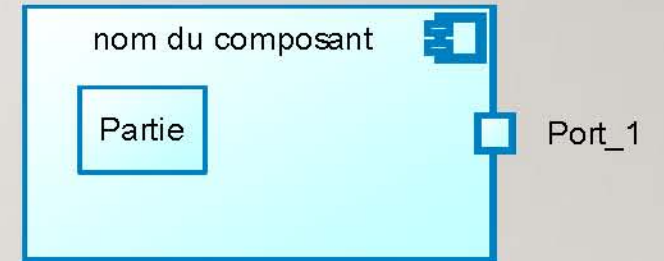
Représentation graphique



Avec une partie



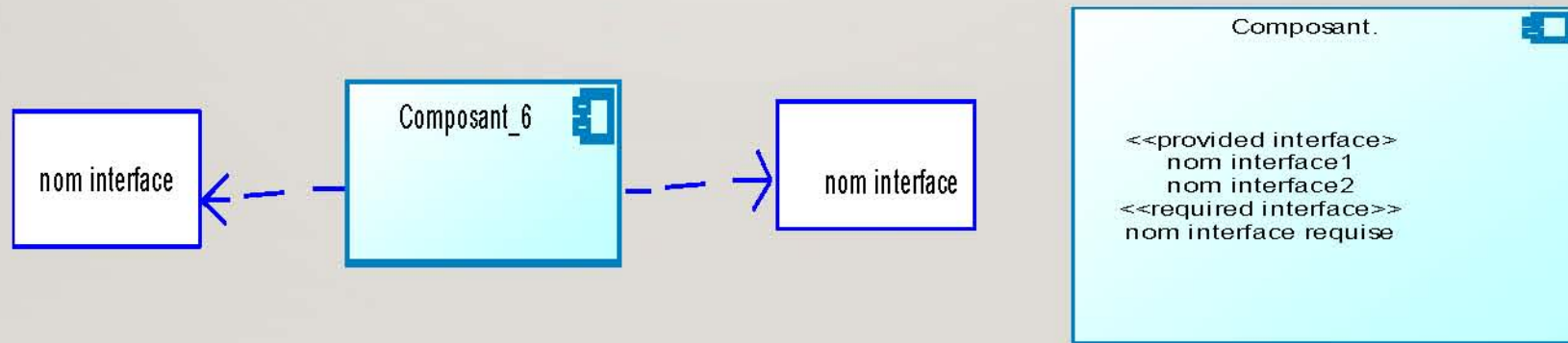
Avec partie et un port



Avec interfaces

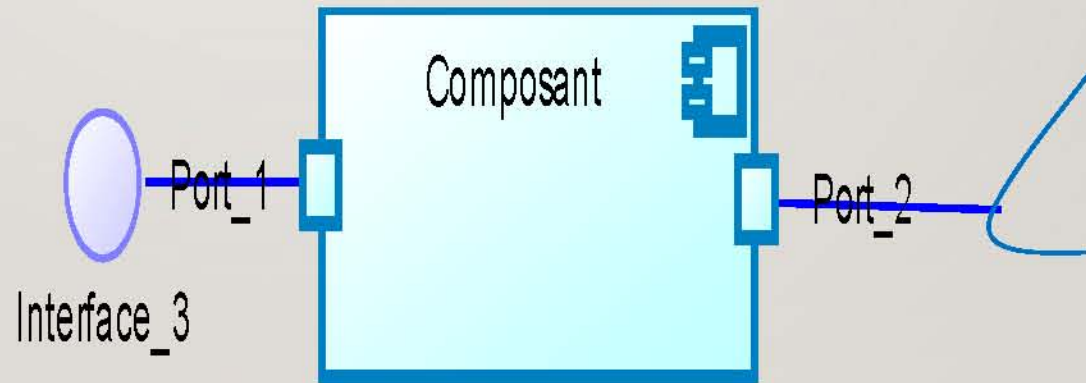
DIAGRAMME DE COMPOSANT

- Un composant fournis des services via des interfaces et peut être remplacé par un autre compatible.
- Deux types d'interfaces : interface requise et interface fournie.
 - L'interface requise fournis un service au composant dont il a besoin pour fonctionner.
 - L'interface fournie à travers laquelle le composant fourni lui-même un service.



36 DIAGRAMME DE COMPOSANT

Port = point de connexion entre le composant et son environnement, c'est la matérialisation de l'interface,



37 DIAGRAMME DE COMPOSANT

- **Boite noire**
 - aucune connaissance sur le contenu,
 - Seul accès par les interfaces,
- **boite blanche**
 - Spécifications des objets constituant le composant et en indiquant leurs relations.

DIAGRAMME DE DÉPLOIEMENT

- Le diagramme de déploiement permet aux concepteurs et aux ingénieurs réseau/performance de se concentrer sur le placement et la configuration des composants à l'exécution, incluant la topologie du hardware .
- Le diagramme de déploiement montre la configuration physique du système en terme de nœuds physiques et connexions physiques entre les nœuds comme une connexion réseau.
- Le diagramme de déploiement modélise la distribution de votre système software sur le hardware.

39 DIAGRAMME DE DÉPLOIEMENT

- Le diagramme de déploiement représente :
 1. la disposition physique des ressources matérielles constituant le système et montre la répartition des composants sur ces ressources matérielles.
 2. La nature des connexions de communication entre les différentes ressources matérielles.
- Le but du diagramme de déploiement est de spécifier la distribution des composants et d'identifier les performances comme résultat .

40 DIAGRAMME DE DÉPLOIEMENT

- Le diagramme de déploiement mappe l'architecture logicielle à l'architecture matérielle.
- Le processus de déploiement est un processus à 2 étapes:
 1. Dans le workflow de conception, on se concentre uniquement sur les nœuds ou instances de nœuds et connexion.
 2. Dans le workflow de l'implémentation, on se concentre sur l'attribution des instances de composants physiques aux instances de nœuds (formulaire d'instance) ou des composants aux nœuds (formulaire de descripteur).

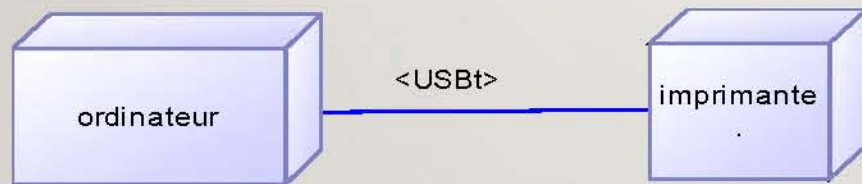
4 | DIAGRAMME DE DÉPLOIEMENT

- ❑ Le diagramme de déploiement du formulaire d'instances montre les instances de composants déployées sur des éléments matériels spécifiques.
- ❑ Le diagramme de déploiement de formulaire de descripteur montre les types de composants déployés sur les types de matériels.

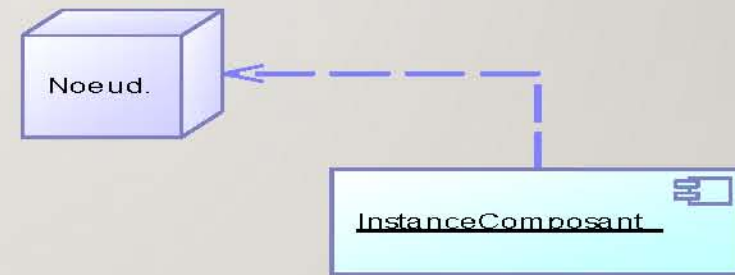
42 DIAGRAMME DE DÉPLOIEMENT



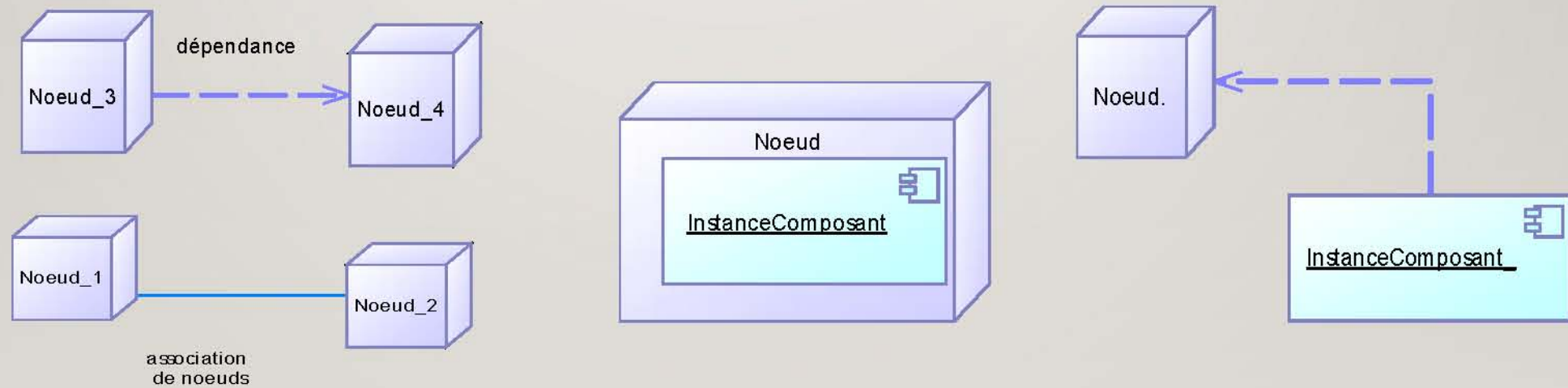
43 DIAGRAMME DE DÉPLOIEMENT



CHEMIN DE COMMUNICATION



44 DIAGRAMME DE DÉPLOIEMENT



45 DIAGRAMME DE PROFIL

- **Profil:** sorte de package permettant d'étendre un métamodèle de référence comme celui d'UML.
- **Stéréotype** : définit de nouveaux concepts ou termes spécifiques par extension de métamodèle UML . Peut avoir des propriétés appelées tag.
- **Extension:** la relation d'extension entre stéréotypes et méta classe UML est représentée graphiquement comme la relation de généralisation, mais avec une flèche noire.

46 DIAGRAMME DE PROFIL

- Créer pour :
 - développer des plateformes spécifiques au développement comme .NET et J2EE.
 - des projets spécifiques comme la modélisation des processus d'affaires.
 - Définir une variante d'UML pour un domaine spécifique. SyML (System Modeling Language) est un exemple récent de UML2.

UNE DEFINITION PROFIL PEUT INCLURE DES ELEMENTS UML COMME LES STEREOTYPES ET LES CONTRAINTES.

Le diagramme de profil permet l'extension de UML comme la visualisation de stéréotypes (hériter toutes les caractéristiques d'un méta-classe.

