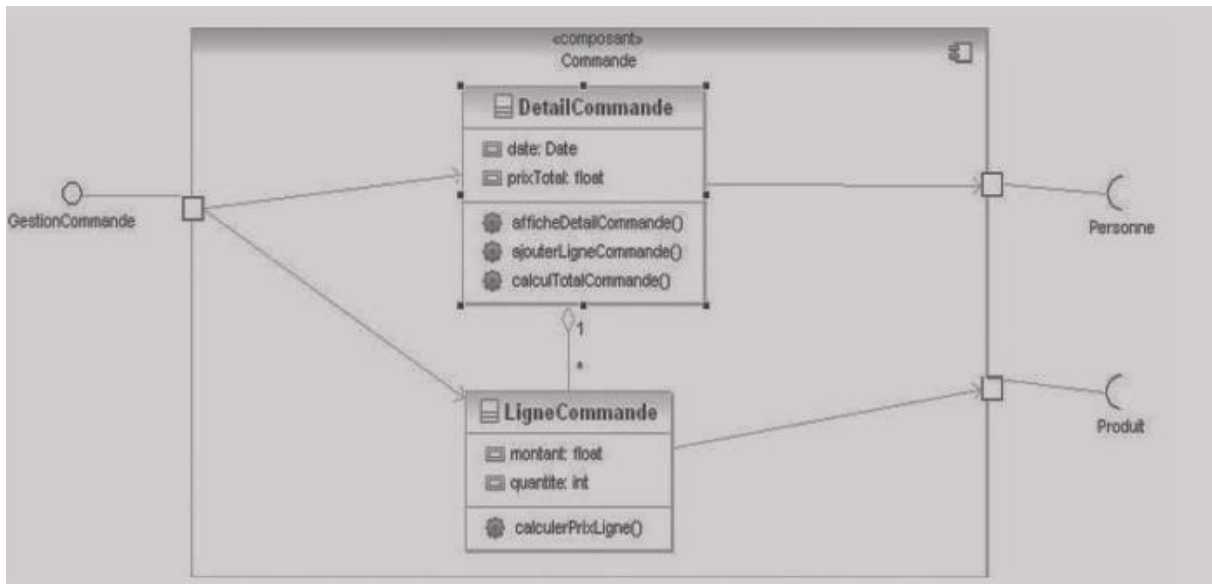
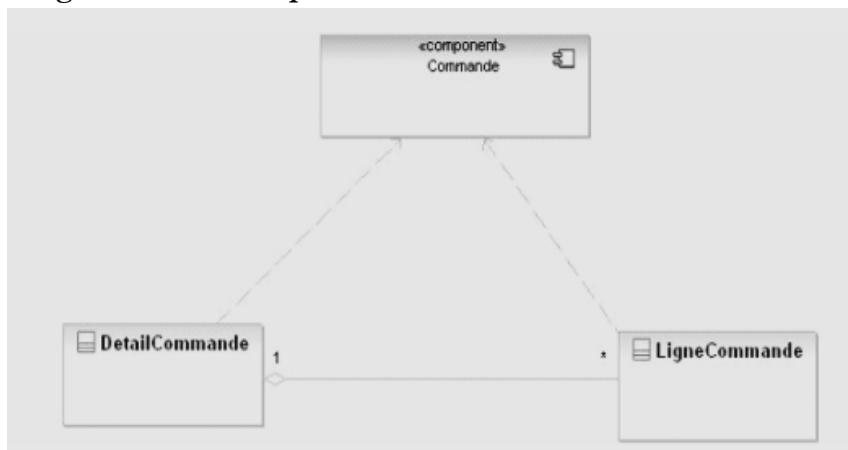


TD 3 : Diagrammes de composants & déploiement

Exercice 1 :

1. Elaborer le diagramme de composants correspondant aux énoncés suivants :
 - a) Le composant **fraudeAgent.dll** contient les classes **agentDesFraudes**, **reglementDesFraudes** et **RechercheDePattern**.
 - b) Le composant **image.java** dépend de l'interface **ImageObserver** du composant **component.java**.
 - c) Une application constituée des fichiers suivants :
 - un code source **registre.cpp**
 - un programme exécutable **registre.exe**
 - des librairies dynamiques **personne.dll** et **cours.dll**. Les librairies à liens dynamiques sont utilisées lors de l'exécution d'une application.
2. Soient les diagrammes de composants suivants :



Désigner la différence entre les deux diagrammes.

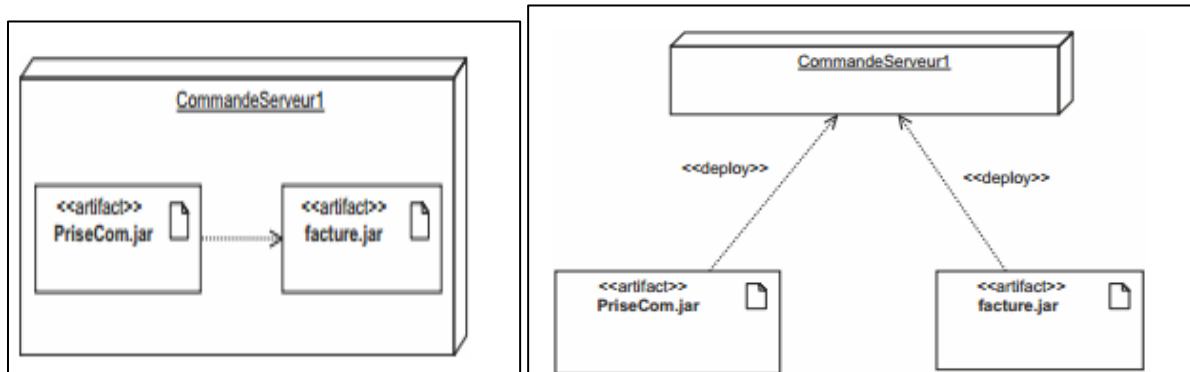
Exercice 2 : Une **architecture matérielle** est constituée d'un serveur, d'un kiosque et d'une console. Le serveur utilise une tour de disques durs montés en RAID. La console communique avec le serveur par une liaison Ethernet 100b et avec la console par une liaison RS232C. Les caractéristiques du serveur sont :

- Processeur Pentium
- Mémoire vive : 4 Go
- Ecran 17 pouces

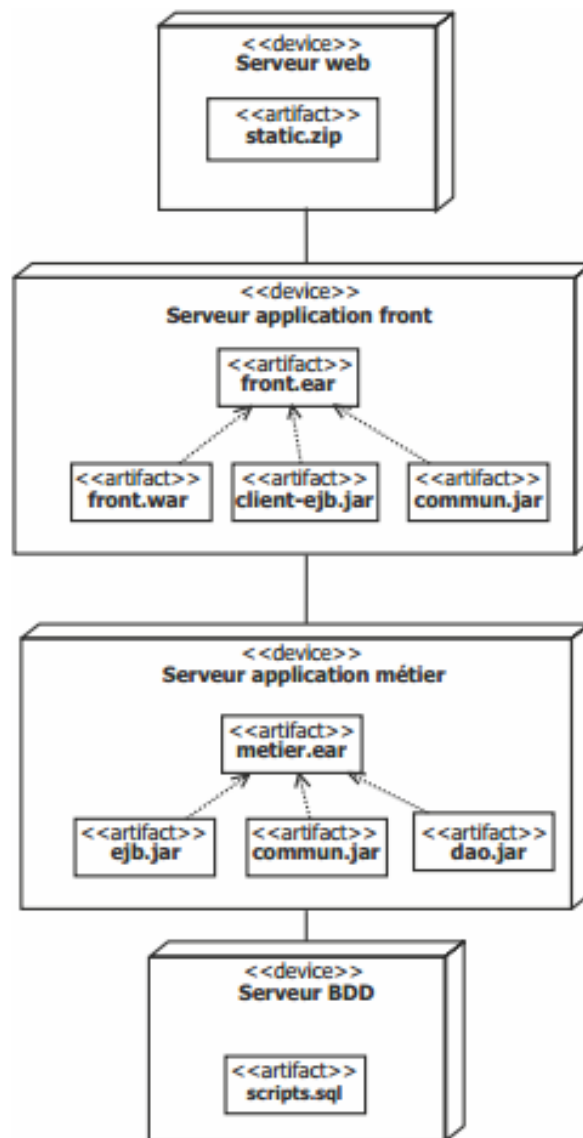
Question : Représentez le diagramme de déploiement correspondant.

Exercice 3 :

1. Quelle est la différence entre les deux diagrammes suivants ?



2. Soit le diagramme de déploiement suivant :



- Expliquer la structuration de ce diagramme.

Solution :

Exercice 1 :

2. Désigner la différence entre les deux diagrammes ci-dessous :

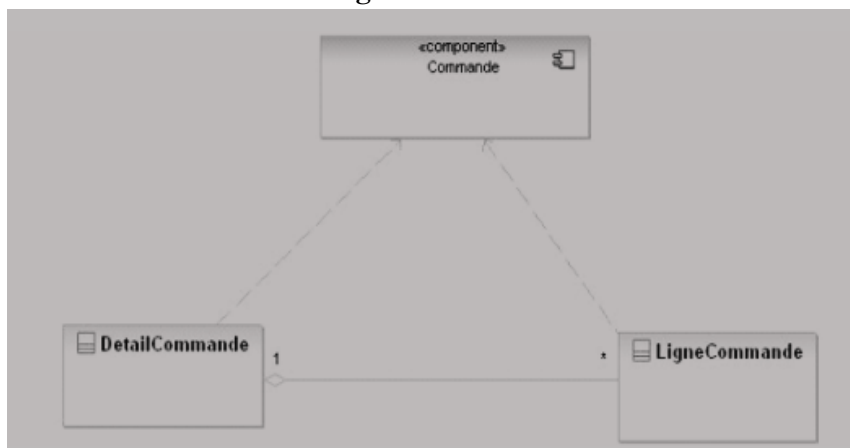


Diagramme 1 : Une autre **représentation interne du composant** peut être aussi utilisée en ayant recours aux **dépendances**. Ainsi, les classificateurs qui composent le composant sont reliés à celui-ci par une relation de dépendance.

Les relations entre les classificateurs (association, composition, agrégation) sont aussi modélisées.

Néanmoins, si elles sont trop complexes, elles peuvent être représentées sur un diagramme de classe relié au composant par une note.

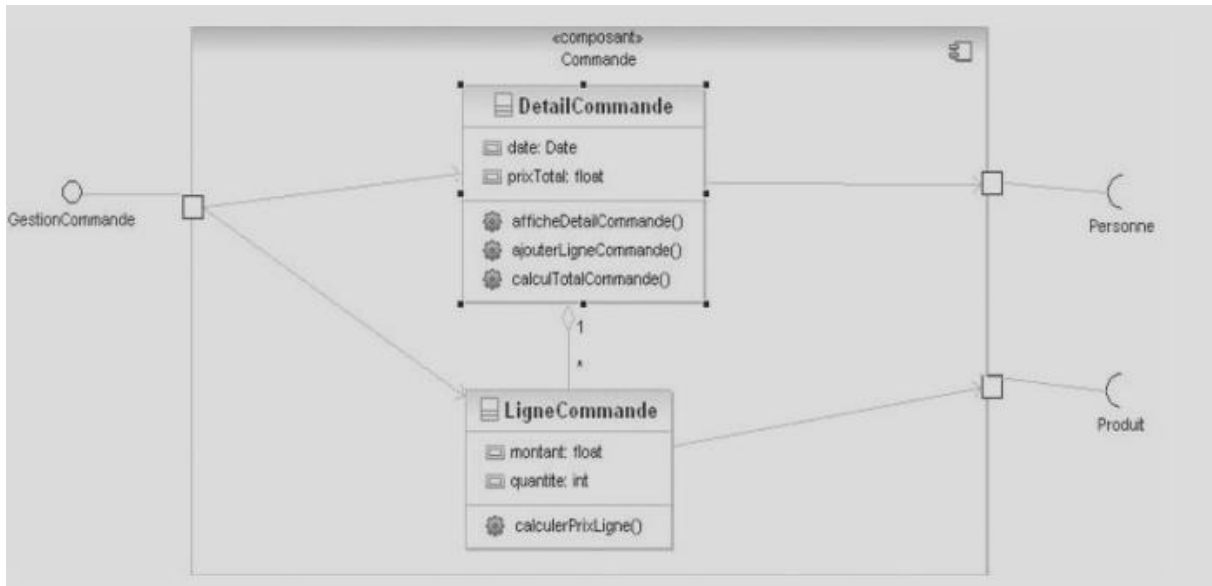
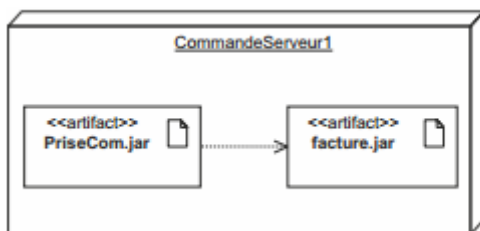


Diagramme 2 : Le composant Commande est constitué de deux classes (classificateur) reliées par une agrégation : DetailCommande et LigneCommande. L'interface fournie GestionCommande est accessible de l'extérieur *via* un port et permet d'accéder *via* les connecteurs aux opérations des deux classes DetailCommande et LigneCommande (ajouterLigneCommande, calculTotal-Commande, calculPrixLigne). L'interface requise Personne est nécessaire pour l'affichage du détail de la commande et est accessible *via* un port du composant Commande. L'interface requise Produit est nécessaire pour le calcul du prix de la ligne de commande et est accessible *via* un port du composant Commande.

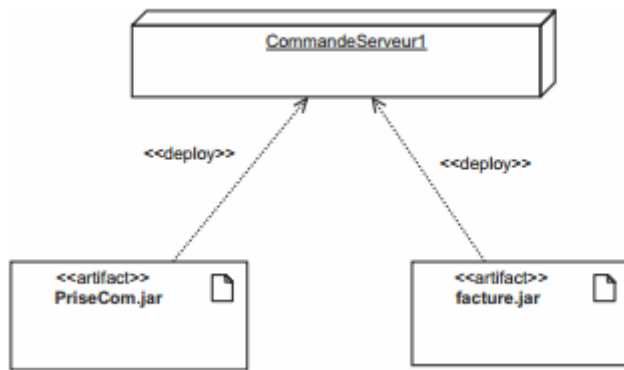
Exercice 3:

1.

- a) **Représentation inclusive** – Dans cette représentation, un artefact est représenté à l'intérieur du nœud auquel il se situe physiquement.



- b) **Représentation avec un lien de dépendance typé « deploy »** – Dans ce cas l'artefact est représenté à l'extérieur du nœud auquel il appartient avec un lien de dépendance entre l'artefact et le nœud typé avec le mot-clé « deploy ».



2.

Dans ce diagramme, plusieurs composants sont déployés.

- **Un serveur web** où se trouvent les éléments statiques du site dans une archive : images, feuilles de style, pages html (static.zip).
- **Un serveur d'application « front »** sur le quel est **déployée l'archive « front.ear »** composée de l'application web « **front.war** » et d'autres composants nécessaires au fonctionnement de cette archive web comme « **clientejb.jar** » (classes permettant l'appel aux EJB) et « **commun.jar** » (classes communes aux deux serveurs d'application).
- **Un serveur d'application métier** sur lequel sont déployés les composants : « .jar ». Ils sont packagés dans l'archive « **metier.ear** ». Deux autres archives sont nécessaires au fonctionnement des EJB : « dao.jar » (classes qui permettent l'accès à la base de données) et « commun.jar » (classes communes aux deux serveurs d'application).
- Un serveur BDD (base de données) sur lequel sont stockées des procédures stockées PL/SQL : « scripts.sql ».

