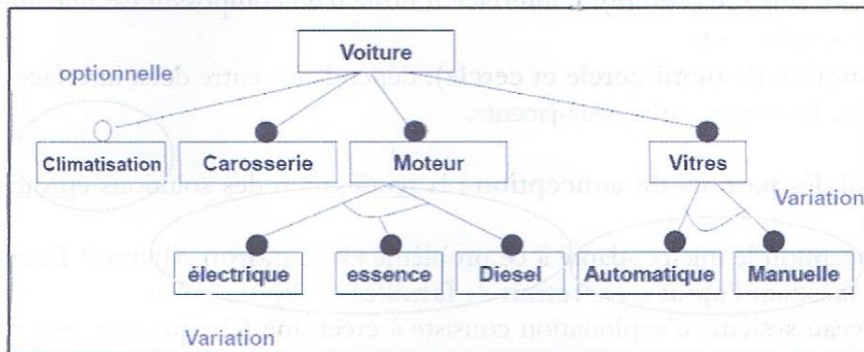


Corrigé Type du Contrôle de Génie Logiciel 2

Questions de Cours (6 Points)

1.

- Définition d'une ligne de produits logiciels (LdP): Une LdP logiciels est un **ensemble de systèmes partageant un ensemble de propriétés communes** et satisfaisant des besoins spécifiques pour un **domaine particulier**. (1 Pt)
- L'intérêt principal des lignes de produits logiciels : favorise la **réutilisation** et la **configuration en masse** de systèmes logiciels ce qui permet le développement d'une multitude de produits logiciels avec un **gain** considérable en termes de **coût**, de **temps** et de **qualité**. (1 Pt)
- Exemple de ligne de produits logiciels sous forme de diagramme FODA : LdP Voiture (1 Pt)



2. Quel énoncé est vrai à propos des lignes de produit logiciels : (1 Pt)

d) Aucun de ces énoncés.

3. Quel énoncé est vrai à propos des styles architecturaux ? (1 Pt)

c) Aucun de ces énoncés.

4. Quel énoncé est vrai à propos du style architectural MVC (Model View Controller) ? MVC permet la séparation entre :

b) Les données, la présentation et les traitements du système concerné. (1 Pt)

Exercice 1 (6 Points): Les réponses doivent être détaillées.

1.

a)

- Ce diagramme est une représentation combinée du diagramme de déploiement avec le diagramme de composants d'une application web. (0.75 Pt)

Il est composé de deux parties physiques déployées sur deux nœuds: (0.75 Pt)

- La première partie constitue la partie cliente sur lequel est déployé l'artéfact (i.e. composant) « navigateur web ».
- La deuxième partie étant un Serveur, sur lequel est déployé deux artéfacts, à savoir « Tomcat » et « MySQL ».

Ces deux parties interagissent entre elles via une dépendance d'Internet (HTTP) selon le protocole de communication TCP/IP.

- L'intérêt du modèle : modélise l'architecture physique du système concerné, plus précisément, il montre la **répartition de l'architecture logique (logicielle) sur l'architecture physique (matérielle)**. (0.75 Pt)

b) Les composants de base de ce modèle sont : (0.75 Pt)

- **Nœud** : une ressource sur laquelle des artéfacts (i.e. composants) peuvent être déployés pour être exécutés. (Ici, on a deux nœuds client et serveur).
- **Artéfact** : est la spécification d'une entité physique d'information utilisée ou produite par le système concerné. Les artéfacts sont équivalents aux composants (Ici, Navigateur web, Tomcat et MySQL).
- **Les Chemins de communication** : des associations entre deux nœuds au travers lesquelles les informations sont échangées. (Ici, la liaison TCP-IP / HTTP).

- **La flèche en pointillée** : modélise le déploiement de l'artéfact sur le nœud, elle porte le stéréotype « deploy » et pointe vers le nœud.

c) C'est une architecture Client/ Serveur à 2 tiers. (1 Pt)

2.

a) Ce diagramme représente le **diagramme de composants** d'une application d'inscription cours en ligne. Il est constitué de **quatre composants**, à savoir : « InscriptionCours », « Comptabilité », « CatalogueCours » et « Annuaire ». Les composants « Comptabilité », « CatalogueCours » et « Annuaire » **fournissent** chacun d'eux une seule **interface qui sont respectivement**: « Facturation », « Cours » et « FichePersonne ». Ces interfaces sont requises par le composant « InscriptionCours ». (1 Pt)

L'intérêt du diagramme de composants : permet de modéliser **architecture logicielle** d'un système informatique.

b) Les éléments de ce type de diagramme sont : (1 Pt)

- **Composant** : est une unité logicielle autonome au sein d'un système global ou d'un sous-système.

- **Interface requise (demi-cercle)**: L'interface requise d'un composant est une interface **dont il a besoin pour fonctionner**.

- **Interface offerte ou fournie (cercle)**: L'interface fournie d'un composant est une interface qui **fournit un service** aux autres composants.

- **Dépendance (jonction de demi-cercle et cercle)**: dépendance entre deux interfaces requise et offerte spécifie le lien entre les composants sous-jacents.

Exercice 2

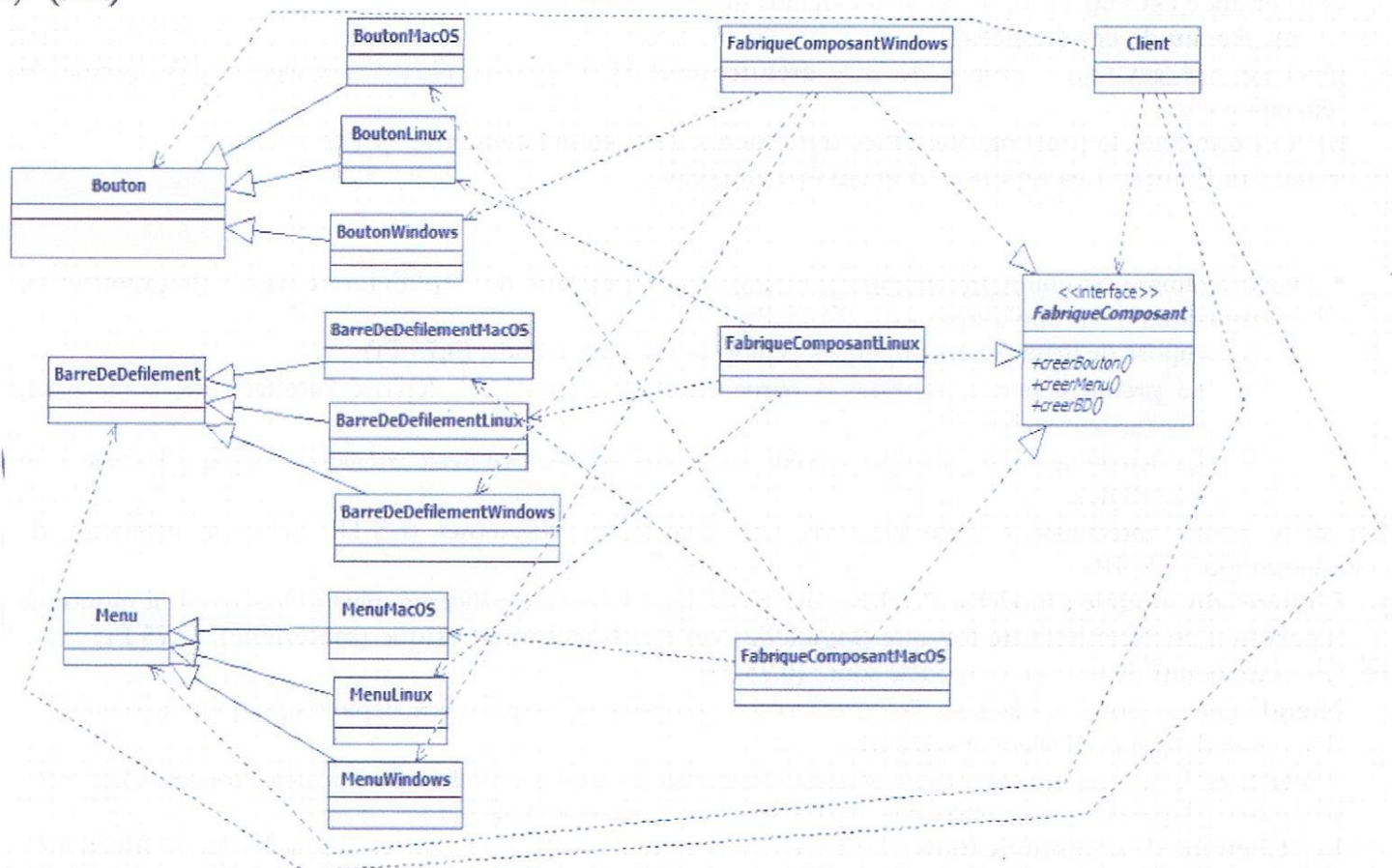
a) L'intérêt principal des **patrons de conception** : la **réutilisation** des solutions **éprouvées** pour des problèmes **récurrents**. (1 Pt)

b) Le patron de conception le mieux adapté à ce problème est le **patron Abstract Factory** puisque il offre une structure flexible facilitant **l'ajout** ou le **retrait de familles d'objets**. (1 Pt)

L'ajout d'un nouveau système d'exploitation consiste à créer une fabrique concrète pour ce système ainsi que les produits concrets correspondants. (1.5 Pt)

Type de ce patron : patron de création. (0.5 Pt)

c) (3 Pt)



d) Oui il existe d'autres moyens pour la modélisation de ce problème, tel que les lignes de produits logiciels (1 Pt).