

Examen

Exercice 1: (7 points)

Question 1 : Quelle est la définition correcte d'un intergiciel ?

- A. Un matériel réseau qui connecte physiquement les serveurs dans un centre de données.
- B. Un logiciel qui agit comme une passerelle entre les applications, outils et bases de données pour offrir des services unifiés.
- C. Un système d'exploitation spécifique pour les serveurs cloud.
- D. Un langage de programmation permet la création de sites web dynamiques.

Question 2 : Dans le cadre de l'intégration d'applications patrimoniales (legacy systems), quel composant est utilisé pour faire le pont entre l'interface originelle et une nouvelle interface normalisée ?

- A. Un mandataire (Proxy).
- B. Un courtier (Broker).
- C. Une enveloppe (Wrapper).
- D. Un bus à messages.

Question 3 : Parmi les combinaisons de caractéristiques de communication (topologie, prévisibilité), laquelle est actuellement considérée comme « vide » ou non réalisable avec les techniques actuelles ?

- A. Fixe, imprévisible.
- B. Fixe, prévisible.
- C. Variable, imprévisible.
- D. Variable, prévisible.

Question 4 : Dans un système d'intergiciel, quel schéma de communication correspond à l'envoi d'une requête où le client se bloque en attendant la réponse ?

- A. La communication asynchrone.
- B. La communication par publication-abonnement (publish-subscribe).
- C. La communication synchrone.
- D. La communication pair à pair (P2P).

Question 5 : Concernant l'appel de procédure à distance, pourquoi ne peut-on généralement pas passer de pointeurs en paramètres ?

- A. Car le réseau ne peut pas transmettre d'adresses mémoire physiques.
- B. Car les processus appelant et appelé s'exécutent dans des espaces d'adressage différents.
- C. Car les langages de programmation modernes interdisent les pointeurs.
- D. Car cela pose des problèmes de sécurité au niveau du bus système.

Question 6 : Quelle technique côté serveur permet d'éviter le coût de création fréquente de threads lors de multiples appels RPC simultanés ?

- A. La création dynamique systématique de threads.

- B. L'utilisation d'un processus unique séquentiel.
- C. La gestion d'une réserve (pool) de threads créés à l'avance.
- D. La suppression des threads pour utiliser uniquement des processus lourds.

Question 7 : Quel est le rôle du langage de définition d'interface (IDL) dans le développement d'applications RPC ?

- A. Il sert à écrire le code métier de l'application.
- B. Il permet de gérer les fichiers de configuration réseau.
- C. Il spécifie les types et modes de passage des paramètres pour générer automatiquement les souches (stubs).
- D. Il remplace le système d'exploitation pour la gestion des sockets.

Exercice 2 (6 points)

L'entreprise **SmartLogistix**, spécialisée dans la logistique internationale, souhaite construire une application réseau innovante pour gérer sa flotte de camions et ses entrepôts connectés. L'application doit collecter des données en temps réel, analyser les itinéraires pour optimiser les coûts et générer des rapports mensuels pour la direction.

Voici les besoins spécifiques du projet :

1. Les camions sont équipés de capteurs (réseaux WSN) qui envoient la température, la localisation GPS et le niveau de carburant.
2. L'application doit alerter le conducteur immédiatement si un problème technique est détecté.
3. Toutes les données doivent être centralisées dans le Cloud pour être analysées par des algorithmes d'IA afin de prévoir la maintenance des véhicules.
4. L'entreprise possède déjà une base de données relationnelle pour les clients et les factures, mais les nouvelles données des capteurs sont non structurées et volumineuses (Big Data).
5. Les data scientists ont besoin d'accéder aux données brutes pour tester de nouveaux modèles d'apprentissage automatique, tandis que les managers ont besoin de tableaux de bord propres pour la prise de décision.

Questions :

1. Pour la génération des rapports mensuels destinés à la direction, quel type de traitement consistant à décomposer les données en unités plus petites est recommandé ?

- | | |
|---------------------------------|-----------------------------|
| A. Le traitement en temps réel. | B. Le traitement en ligne. |
| C. Le traitement par lots. | D. Le traitement distribué. |

2. SmartLogistix doit stocker des données non structurées provenant des capteurs. Quel type de Système de Gestion de Base de données est recommandé pour gérer ce type de données ?

- | | |
|---|-------------------------------|
| A. SGBD Relationnel (RDBMS) uniquement. | B. SGBD Hiérarchique. |
| C. Base de données NoSQL. | D. Système de fichiers local. |

3. Parmi les types de bases de données NoSQL, lequel est spécifiquement cité dans le texte comme adapté pour stocker les données horodatées de manière séquentielle ?

- | | |
|---|---------------------------------------|
| A. Les bases de données de documents. | B. Les bases de données clés-valeurs. |
| C. Les bases de données de séries chronologiques (Time-series). | D. Les bases de données graphiques. |

4. L'entreprise cherche une architecture de stockage qui permette à la fois la flexibilité d'un Data Lake (data scientists) et la rigueur d'un Entrepôt de données (les managers). Quelle solution hybride est conseillée ?

- A. Le Data Mart.
- B. Le Data Lakehouse.
- C. Le Cloud Computing public.
- D. L'ETL traditionnel.

5. Selon la conclusion du document, quel est l'avantage majeur du Cloud Computing pour la gestion des Big Data de SmartLogistix ?

- A. Il permet d'éliminer totalement le besoin en administrateurs de base de données.
- B. Il permet de gérer d'énormes charges de travail et de combiner les plateformes en un seul système adaptable.
- C. Il garantit que les données ne seront jamais stockées sur des serveurs étrangers.
- D. Il transforme automatiquement les données manuelles en données électroniques sans intervention humaine.

Exercice 3: (7 points)

Nous cherchons à produire un logiciel d'édition pour des documents structurés.

<pre> <section> <titre>TD</titre> <contenu> Les exercices ne sont pas indépendants. <section> <titre>Exercice 1</titre> <contenu> <section> <titre>Question 1</titre> <contenu>En premier lieu,...</contenu> </section> <section> <titre>Question 2</titre> <contenu>Maintenant, modélisez...</contenu> </section> </contenu> </section> <section> <titre>Exercice 2</titre> <contenu>Il est à noter que...</contenu> </section> </contenu> </section> </pre>	TD Les exercices ne sont pas indépendants. Exercice 1 Question 1 En premier lieu,... Question 2 Maintenant, modélisez... Exercice 2 Il est à noter que...
---	---

Les documents sont structurés à l'aide de balises à la façon des documents XML. A gauche on trouve le code source du document et à droite, une manière de l'interpréter pour en faire quelque chose de plus présentable. Les balises <section> permettent de structurer le document en parties et en sous parties. Les balises <titre> et <contenu> définissent des compartiments de données pour chaque section. Le compartiment « contenu » peut contenir des sous-sections. Chaque cadre dans les figures représente une vue.

1 Patron « composite » (vue-vue)

Pour visualiser le document ou seulement une section, on peut avoir recours à une vue du code, mais on veut également une représentation WYSIWYG (What You See Is What You Get) comme celle ci-dessus à droite. Pour ce faire, on considère que chaque section peut l'objet d'une vue et que la vue en question peut contenir des sous-vues représentant les sous-sections.

Dans la vue représentée à gauche, chacune des vues est décomposée en sous-vues. Dans la vue représentée à droite, on a plutôt choisi de représenter chacun des exercices au moyen de leur code, sans les décomposer. Ces modes de représentation sont tous les deux possibles dans le logiciel que l'on cherche à concevoir.

Question : Sans prendre en compte la manière dont est structuré le code source du document, en se restreignant à la seule organisation des vues entre-elles, proposez une modélisation de la structure hiérarchique des vues. Vous pourrez vous appuyer sur le patron « composite ».

TD	TD
Les exercices ne sont pas indépendants.	Les exercices ne sont pas indépendants.
Exercice 1	Exercice 1
<i>Question 1</i>	<i>Question 1</i>
En premier lieu,...	En premier lieu,...
<i>Question 2</i>	<i>Question 2</i>
Maintenant, modélisez...	Maintenant, modélisez...
Exercice 2	Exercice 2
Il est à noter que...	Il est à noter que...

2. Patron « observer » (vue-modèle)

L'application que nous voulons développer doit pouvoir gérer plusieurs vues simultanément. Par exemple, on pourra avoir en même temps une vue du code source complet et une vue WYSIWYG ou tout autre intermédiaire. Le code source du document est appelé le modèle. Dès que ce dernier est modifié, on veut que les vues se redessinent de manière adéquate.

Question : Proposez une modélisation UML qui rende compte de cette fonctionnalité. Vous pourrez utiliser le patron de conception «observer » en considérant les vues comme des observateurs du modèle

3. Patron « strategy »(vue-contrôleur)

En plus des vues et du modèle du document, on souhaite disposer de moyens d'interaction avec le système pour modifier le modèle. Pour ce faire, on se dote de contrôleurs associés aux vues. Les contrôleurs réagissent aux événements extérieurs (clics sur des boutons, frappes clavier...) et modifient le modèle en réponse aux événements en question. La manière dont un contrôleur agit sur le modèle dépend de la vue à laquelle il est associé. Par exemple, une clic de souris sur une vue du code source pourra permettre de déplacer le curseur alors qu'un clic semblable sur une vue hiérarchique permettra d'accéder à certaines de ses propriétés. Ainsi, les contrôleurs sont semblables à des algorithmes qui, en fonction du contexte défini par la vue, déclencheront des opérations différentes,

Question : Utilisez le patron « strategy » pour proposer une modélisation du lien entre vues et contrôleurs.

4. Patron « observer »(contrôleur-modèle)

Il peut arriver que l'état du modèle ait un impact sur les contrôleurs. Par exemple, il se peut qu'en fonction du modèle, certaines options ne soient pas disponibles (menus, cases à cocher...). Les contrôleurs doivent donc être avertis de toutes les modifications du modèle.

Question : Utilisez le patron « observer » pour modéliser ce lien entre modèles et contrôleurs. Nous considérerons qu'il y a deux types de contrôleurs : pour gérer la souris et pour gérer le clavier.

5. Fusion de l'utilisation de plusieurs patrons de conception

Dans les exercices précédents, nous avons modélisé différentes parties d'un système en nous focalisant successivement sur les relations entre :

- vues et sous-vues ;
- vues et modèle ;
- vues et contrôleurs ;
- contrôleurs et modèles.

Question : Combinez maintenant les différents sous-modèles élaborés pour former un modèle global du système complet. Ce modèle utilise trois patrons de conception : composite, observer et strategy, à chaque fois pour obtenir des fonctionnalités différentes mais complémentaires. En outre, on prendra ici en compte le fait que lorsqu'un contrôleur reçoit un événement, il demande au modèle de se mettre à jour.

Solution

Exo 1

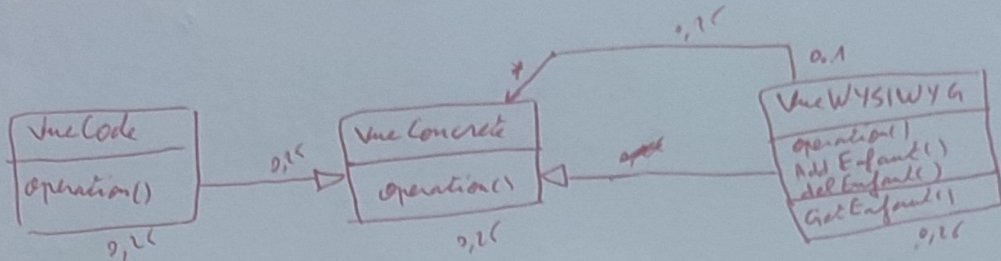
1-B 2-C 3-D 4-C 5-B 6-C 7-C

Exo 2

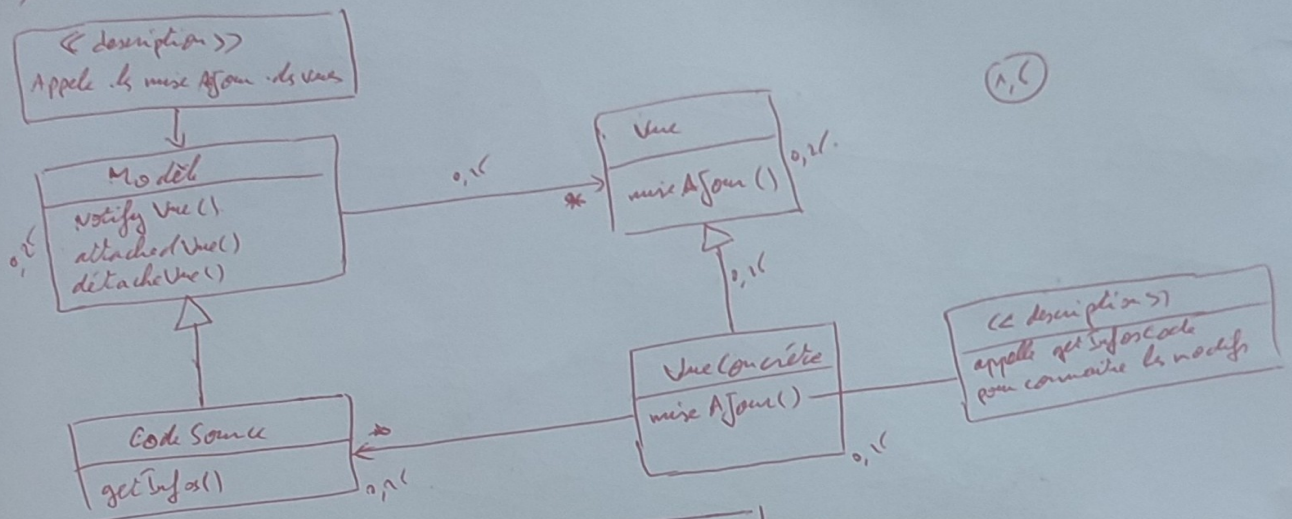
1-C 2-C 3-C 4-B 5-B

Exo 3

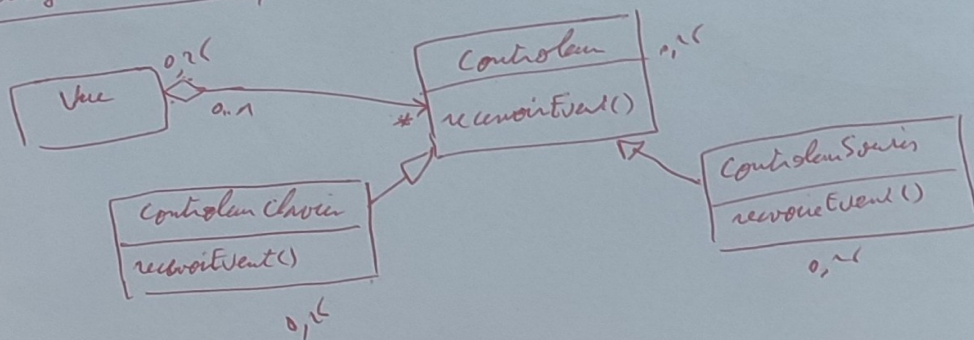
1)



2)



3)



4)

