

QCM Diagramme de Séquence

Résumé des réponses générales sur les trois premiers diagrammes UML

1. Diagramme de Cas d'Utilisation

- **Définition** : Il décrit les interactions entre un système et ses utilisateurs (acteurs).
- **Éléments principaux** : Acteurs, cas d'utilisation, relations entre eux.
- **Acteurs** : Principal (initie l'interaction) et Secondaire (assiste dans le processus).
- **Relations** :
 - **Inclusion** : Un cas d'utilisation réutilise un autre (obligatoire).
 - **Extension** : Un cas d'utilisation ajoute un comportement optionnel.
 - **Généralisation** : Un cas d'utilisation est une version spécialisée d'un autre.
- **Importance** : Il permet d'identifier les besoins fonctionnels du système.
- **Erreurs courantes** : Trop de détails techniques, confusion entre acteurs et utilisateurs.

2. Diagramme de Classes

- **Définition** : Il représente la structure statique du système en modélisant les classes et leurs relations.
- **Éléments principaux** : Classes, attributs, méthodes, relations (association, héritage, composition...).
- **Classes vs Objets** : Une classe est un modèle, un objet est une instance concrète.
- **Relations** :
 - **Association** : Lien simple entre classes.
 - **Agrégation** : Une classe contient d'autres objets mais peut exister sans eux.
 - **Composition** : Une classe contient des objets qui ne peuvent exister sans elle.
 - **Héritage** : Une classe fille hérite des caractéristiques d'une classe parent.

- **Encapsulation** : Modificateurs de visibilité :
 - **Public (+)** : Accessible partout.
 - **Privé (-)** : Accessible uniquement dans la classe.
 - **Protégé (#)** : Accessible dans la classe et ses sous-classes.
- **Interface vs Classe Abstraite** :
 - **Interface** : Définit uniquement des méthodes à implémenter.
 - **Classe Abstraite** : Peut contenir des méthodes définies et abstraites.

3. Diagramme de Séquence

- **Définition** : Il décrit la communication dynamique entre les objets dans le temps.
- **Éléments principaux** : Acteurs, objets, lignes de vie, messages, activation.
- **Messages** :
 - **Synchrone** : L'expéditeur attend la réponse avant de continuer.
 - **Asynchrone** : L'expéditeur n'attend pas de réponse.
 - **Retour** : Message de réponse à un appel précédent.
- **Création et Destruction d'objets** :
 - **Création** : Flèche pointant vers la ligne de vie de l'objet.
 - **Destruction** : Croix à la fin de la ligne de vie.
- **Fragments combinés** : Pour représenter des interactions complexes :
 - **alt** (alternative) : Choix entre plusieurs scénarios.
 - **loop** (boucle) : Répétition d'actions.
 - **par** (parallèle) : Exécution simultanée.
- **Utilité** : Précise l'enchaînement des interactions et aide à comprendre le comportement du système.