**Questions & Réponses**

Date : 27 decembre 2024

Cours : POO

Fait par : SENE

Q **Qu'est-ce qu'une classe en programmation orientée objet ?**

R Une classe est un plan ou une structure qui définit les attributs et méthodes d’un objet. Elle permet de créer un ou plusieurs objets à partir d’un modèle commun.

Q **Quelle est la différence entre une classe et un objet ?**

R Une classe est un plan qui décrit les caractéristiques et comportements d’un objet. Un objet est une instance concrète de cette classe.

Q **Quels sont les quatre concepts fondamentaux de la POO ?**

R

* **Abstraction** : Cacher les détails complexes et exposer l’essentiel.
* **Encapsulation** : Regrouper les données et méthodes dans une classe.
* **Héritage** : Permettre à une classe d’utiliser ou d’étendre une autre classe.
* **Polymorphisme** : Une méthode peut avoir des comportements différents selon le contexte.

Q **Quels sont les avantages d’utiliser UML dans la conception d’un système ?**  
R UML fournit une notation graphique standardisée pour modéliser, documenter et communiquer la structure et le comportement d’un système orienté objet.

Q **Quelle est la différence entre un diagramme de classe et un diagramme de séquence ?**

R

* Le **diagramme de classe** représente les classes, leurs attributs, méthodes et relations.
* Le **diagramme de séquence** illustre les interactions temporelles entre objets dans un scénario spécifique.

Q **Qu’est-ce que l’héritage en POO ?**

R L’héritage est un mécanisme permettant à une classe enfant de réutiliser ou redéfinir les attributs et méthodes d’une classe parent.

Q **Quand utilise-t-on une interface plutôt qu’une classe abstraite ?**

R Une interface est utilisée pour définir un contrat strict que plusieurs classes peuvent implémenter. Une classe abstraite est utilisée pour modéliser un comportement commun avec des variantes possibles.

Q **Quelle est la différence entre agrégation et composition ?**

R

* **Agrégation** : Relation faible, où les objets liés peuvent exister indépendamment.
* **Composition** : Relation forte, où la destruction de l’objet principal entraîne celle des objets liés.

Q **Quels types de besoins identifie-t-on dans une analyse orientée objet ?**

R

* **Besoins fonctionnels** : Ce que l’application doit faire.
* **Besoins non fonctionnels** : Contraintes de performance, sécurité, documentation, etc.

Q **Quels outils sont recommandés pour créer des diagrammes UML ?**

R

* En ligne : Gliffy, Lucidchart, Creatly.
* Applications : Visio, OmniGraffle, UMLet.
* Open source : Dia, AlgoUML.
* IDE : Visual Studio Code, Eclipse (avec plugins UML).