

Modélisation statique

Accueil / Mes cours / MOD1 / Sections / Section 1 / MOD1

Commencé le Monday 14 April 2025, 09:05

État Terminé

Terminé le Monday 14 April 2025, 10:25

Temps mis 1 heure 20 min

Question **1**

Terminé

Noté sur 0,50

🚩 Marquer la question

Le pattern Singleton garantit que :

- a. Il est impossible d'instancier la classe
- b. Plusieurs instances peuvent exister mais une seule est accessible
- c. Chaque objet est unique mais peut être cloné
- d. Une seule instance d'une classe est créée

Question **2**

Terminé

Noté sur 0,50

🚩 Marquer la question

Dans le pattern Strategy, le comportement est :

- a. Encapsulé dans une famille de classes interchangeables
- b. Défini en dur dans la classe principale
- c. Distribué par des visiteurs indépendants
- d. Partagé entre les instances par le biais du pattern Singleton

Question **3**

Terminé

Noté sur 0,50

🚩 Marquer la question

Quel est le principal avantage du pattern Singleton ?

- a. Augmente le couplage entre les classes
- b. Simplifie la sérialisation d'objets
- c. Contrôle centralisé de l'accès à une ressource partagée
- d. Facilite l'héritage multiple

Question **4**

Terminé

Noté sur 0,50

🚩 Marquer la question

Le pattern Visitor est utilisé pour :

- a. Optimiser les performances des algorithmes récursifs
- b. Séparer un algorithme de la structure de données sur laquelle il opère
- c. Permettre une communication entre objets distants
- d. Créer une seule instance globale

Question **5**

Terminé

Noté sur 0,50

🚩 Marquer la question

Le pattern Visitor est particulièrement utile quand :

- a. On veut partager l'état entre plusieurs objets
- b. On veut éviter les classes abstraites
- c. On veut centraliser la gestion d'un seul objet
- d. On veut ajouter des opérations sur une hiérarchie de classes sans les modifier

Question **6**

Terminé

Noté sur 0,50

🚩 Marquer la question

Le pattern Composite est utilisé pour :

- a. Permettre une communication bidirectionnelle entre objets
- b. Limiter l'accès à une seule instance d'un objet
- c. Simplifier l'ajout d'un algorithme externe
- d. Représenter une hiérarchie d'objets sous forme d'arbre

Question **7**

Terminé

Noté sur 0,50

🚩 Marquer la question

Dans le pattern Composite, une "feuille" est :

- a. Un objet qui visite d'autres objets
- b. Un objet singleton racine
- c. Un objet sans enfants
- d. Une interface partagée

Question **8**

Terminé

Noté sur 0,50

🚩 Marquer la question

Quel est l'avantage du pattern Composite ?

- a. Permet de traiter de manière uniforme les objets simples et les compositions
- b. Remplace les pointeurs faibles
- c. Supprime complètement l'héritage
- d. Facilite la synchronisation multithread

Question **9**

Terminé

Noté sur 0,50

🚩 Marquer la question

Le pattern Strategy permet :

- a. De changer dynamiquement le comportement d'un objet
- b. De créer une hiérarchie d'objets sous forme d'arbre
- c. De visiter une structure de données
- d. D'assurer qu'un objet est unique dans le système

Question **10**

Terminé

Noté sur 0,50

🚩 Marquer la question

Dans un diagramme de classes, une association avec multiplicité 0..* signifie :

- a. Qu'il doit y avoir au moins une instance
- b. Qu'il ne peut y avoir qu'une seule instance liée
- c. Que les instances sont créées automatiquement
- d. Qu'une classe peut être liée à zéro ou plusieurs instances de l'autre

Question **11**

Terminé

Noté sur 0,50

🚩 Marquer la question

Quel est le rôle d'un attribut dans une classe UML ?

- a. Représenter une donnée ou une propriété de l'objet
- b. Indiquer un état de l'objet
- c. Servir de clé primaire
- d. Représenter une méthode

Question **12**

Terminé

Noté sur 0,50

🚩 Marquer la question

Quelle notation permet de représenter une méthode publique dans un diagramme de classes UML ?

- a. Une flèche
- b. Le signe plus (+) suivi du nom de la méthode
- c. Le signe moins (-) suivi du nom de la méthode
- d. Un astérisque (*)

Question **13**

Terminé

Noté sur 0,50

🚩 Marquer la question

Dans un diagramme de classes, une composition implique :

- a. Une relation plus faible qu'une agrégation
- b. Une relation entre interfaces uniquement
- c. Que la vie de l'objet contenu dépend de l'objet contenant
- d. Un simple partage d'objets

Question **14**

Terminé

Noté sur 0,50

🚩 Marquer la question

Dans un diagramme de classes, une visibilité "privée" est représentée par :

- a. Un astérisque (*)
- b. Un dièse (#)
- c. Un signe moins (-)
- d. Un signe plus (+)

Question **15**

Terminé

Noté sur 0,50

🚩 Marquer la question

Dans un diagramme de classes, une association entre deux classes peut être :

- a. Symétrique par défaut
- b. Unidirectionnelle ou bidirectionnelle
- c. Uniquement bidirectionnelle
- d. Uniquement unidirectionnelle

Question **16**

Terminé

Noté sur 0,50

🚩 Marquer la question

Dans un diagramme de classes, le symbole de généralisation (héritage) est représenté par :

- a. Une ligne simple
- b. Une flèche en pointillés
- c. Un losange plein
- d. Une flèche avec un triangle blanc pointant vers la superclasse

Question **17**

Terminé

Noté sur 0,50

🚩 Marquer la question

Dans UML, une agrégation est représentée par :

- a. Un losange blanc
- b. Un triangle
- c. Un losange noir
- d. Un rectangle

Question **18**

Terminé

Noté sur 0,50

🚩 Marquer la question

Quelle est la différence entre agrégation et composition ?

- a. La composition permet un lien plus faible
- b. L'agrégation interdit le partage d'objets
- c. Il n'y a pas de différence
- d. La composition implique une relation de vie forte entre les objets

Question **19**

Terminé

Noté sur 0,50

🚩 Marquer la question

Quelle est la différence principale entre une interface et une classe abstraite ?

- a. Une interface ne contient que des méthodes sans implémentation
- b. Une interface peut être instanciée
- c. Une interface peut contenir des attributs privés
- d. Une classe abstraite ne peut pas définir de constructeur

Question **20**

Terminé

Noté sur 0,50

🚩 Marquer la question

En C++, que se passe-t-il si une classe hérite de deux classes qui ont une méthode du même nom ?

- a. Il faut résoudre le conflit explicitement dans la classe dérivée
- b. C'est interdit par le langage
- c. Le compilateur choisit au hasard
- d. La méthode de la première classe est toujours appelée

Question **21**

Terminé

Noté sur 0,50

🚩 Marquer la question

Quelle affirmation est vraie concernant l'héritage en Java ?

- a. Toute classe hérite implicitement de Thread
- b. Les interfaces ne peuvent pas être héritées
- c. Une classe Java ne peut hériter que d'une seule classe
- d. Java permet l'héritage multiple de classes

Question **22**

Terminé

Noté sur 0,50

🚩 Marquer la question

Dans quel cas est-il préférable d'utiliser une interface plutôt qu'une classe abstraite ?

- a. Quand on veut empêcher l'instanciation
- b. Quand plusieurs classes sans lien hiérarchique doivent partager un comportement commun
- c. Quand on veut définir un comportement complet
- d. Quand on veut hériter de plusieurs classes concrètes

Question **23**

Terminé

Noté sur 0,50

🚩 Marquer la question

Quel énoncé décrit le mieux le principe de substitution de Liskov ?

- a. Une classe dérivée doit pouvoir être utilisée partout où sa classe de base est attendue, sans modifier le comportement du programme
- b. Une classe doit toujours hériter d'une autre classe abstraite
- c. Les sous-classes doivent éviter d'utiliser les méthodes de leur super-classe
- d. Chaque classe doit avoir une seule responsabilité

Question **24**

Terminé

Noté sur 0,50

🚩 Marquer la question

Que signifie le principe Open/Closed ?

- a. Une classe doit pouvoir être modifiée à tout moment pour corriger un bug
- b. Le code doit être facilement modifiable par tous les développeurs
- c. Les entités logicielles doivent être ouvertes à l'extension mais fermées à la modification
- d. On doit éviter d'ouvrir les classes au public

Terminer la relectu

English version below.

Le système de gestion de livraison de drones permet d'orchestrer le transport d'objets pesables à l'aide de différentes catégories de drones. Ce système est conçu pour répondre à divers besoins de transport. Il devra coordonner des drones légers ainsi que des drones cargo. L'objectif est de gérer des livraisons de colis, incluant des colis spéciaux, comme des articles nécessitant une livraison urgente ou des colis devant être sécurisés en raison de leur nature fragile, sensible ou confidentielle. Les drones légers sont équipés d'un mécanisme de pince pour saisir et transporter un paquet, tandis que les drones cargo peuvent transporter d'autres drones, mais ne peuvent pas saisir d'autres objets. Certains colis doivent être livrés en urgence (par exemple, avec une date limite stricte), et d'autres devront être sécurisés (par exemple, nécessitant un code de déverrouillage pour faire atterrir le drone livreur). Les missions de livraison planifient l'acheminement de colis par un ou plusieurs drones, en tenant compte des contraintes de poids, d'autonomie et de rapidité.

– Dessiner sur la feuille fournie en annexe, la modélisation en UML correspondant au texte précédent. Vous y ferez apparaître les associations et leurs cardinalités. Vous pourrez justifier textuellement et brièvement vos choix d'association. Vous inscrirez en haut à gauche de votre feuille vos noms, prénoms et login. Une partie non négligeable de la note sera dédiée à la lisibilité du diagramme.

The drone delivery management system orchestrates the transport of weighable objects using different categories of drones. This system is designed to meet various transportation needs. It will coordinate both lightweight and cargo drones. The goal is to manage parcel deliveries, including special packages, such as items requiring urgent delivery or packages that must be secured due to their fragile, sensitive, or confidential nature. Lightweight drones are equipped with a gripper mechanism to grasp and transport a package, while cargo drones can transport other drones but cannot grasp other objects. Some packages must be delivered urgently (e.g., with a strict deadline), and others must be secured (e.g., requiring an unlock code to land the delivery drone). Delivery missions plan the delivery of packages by one or more drones, taking into account weight, range, and speed constraints.

– Draw the UML model corresponding to the previous text on the sheet provided in the appendix. You will show the associations and their cardinalities. You will be able to briefly justify your association choices textually. You will write your first and last names and login details in the top left corner of your sheet. A significant portion of the grade will be devoted to the readability of the diagram.