

## Problème : entreprise du bâtiment

Le modèle UML sur lequel s'appuie le problème est en dernière page. Le modèle est incomplet mais il est là pour vous aider. Toutes les classes ne doivent pas être codées.

1. Intéressons-nous d'abord à la classe Adresse. Donner le squelette de la classe (déclaration + attributs nom, latitude, longitude).

2. Écrire le constructeur de Adresse qui prend en paramètre tous les attributs

3. Écrire les méthodes qui permettent de lire ou de modifier la latitude d'une Adresse

4. Instancier un objet de type Adresse : « mon entreprise », latitude : 45,7598494, longitude : 3,1085476

5. Écrire la méthode « naturelle » pour avoir un bel affichage « nom ( latitude ; longitude ) »

6. Pourquoi cette méthode est toujours disponible ?

7. Que faut-il faire pour rendre la mémoire lorsque cet objet n'est plus utilisé ?

8. Écrire une classe d'exception `IndisponibleException` associée au message d'erreur «Ouvrier indisponible »

9. Écrire l'interface `Specialite` (spécialité)

10. Intéressons-nous à la classe Ouvrier. Écrire le squelette (déclaration + attributs) de la classe. Le compteur est incrémenté à chaque instantiation d'ouvrier et permet d'identifier de manière unique un ouvrier.

11. Comment déclare-t-on et initialise-t-on compteur ?

12. Est-il nécessaire de donner une valeur par défaut aux attributs en Java ?

13. Écrire les méthodes qui permettent de lire les valeurs des attributs nom, valide et compteur de la classe Ouvrier.

14. Écrire les constructeurs de la classe Ouvrier sans duplication de code

15. Écrire une méthode travailler() qui affiche sur la sortie standard « nom » travailler à adresse Latitude ; Longitude) (voir la question 4) si l'ouvrier est disponible ou lève une exception « IndisponibleException » si ce n'est pas le cas

16. Les spécialités sont la plomberie, le chauffage et l'électricité. L'entreprise voudrait identifier des ouvriers spécialisés et ainsi utiliser la modélisation naturelle : plombier, chauffagiste, plombier-chauffagiste et électricien. Quelles déclarations de classe faire pour implémenter Plombier et Plombier-Chauffagiste ?

La modélisation précédente (statique) n'est pas satisfaisante car elle ne permet pas de prendre en compte l'évolution des compétences des ouvriers.

17. Les ouvriers sont encadrés par des chefs d'équipe qui sont eux-mêmes des ouvriers spécialisés. Donner la déclaration de la classe Chef ainsi que l'attribut Equipe

18. Implémenter le constructeur de la classe Chef.

19. Implémenter la méthode travailler(). Si le chef est disponible, il travaille lui-même puis il demande aux membres de son équipe de travailler. Si le chef n'est pas disponible, une exception est levée..

20. Instancier une vache, la faire communiquer et produire. Afficher un message s'il n'est pas possible de « récupérer » du lait.

21. Les animaux peuvent produire() mais également certains types de parcelles comme la jachère (le foin), la culture (céréales ...), un verger (fruit). Comment faire en UML puis en Java pour prendre en compte cette possibilité ?

22. La vache peut se trouver en paturage notamment l'été ou bien en étable si elle est malade ou pendant l'hiver. Comment repérer l'endroit / le lieu où la vache est à un instant donné ?

