

## *Examen n° 1*

# PolytechAstro

L'examen est à réaliser en 1h30 en monôme. Seuls vos projets java (TP par exemple) et l'API de Java sont autorisés. Le TP doit être réalisé sur un poste fixe de la salle de TP.

Aucune question n'est acceptée. Si le cahier des charges vous semble flou, indiquez vos propres hypothèses en commentaire dans le code.

**Les codes sources doivent appartenir à un package portant le nom du monôme (-1pt si ce n'est pas le cas). Une archive *nomDuMonome.zip*, comprenant vos fichiers sources, votre *javadoc* et *QuestionsDeCours.txt*, est à déposer sur Madoc en fin de séance.**

### Exercice 1 : Planète [6pt]

L'entreprise *PolytechAstro* a besoin d'une application pour modéliser des planètes. Voici le cahier des charges :

- une planète est composée de plusieurs couches concentriques superposées :
  - le coeur interne
  - le coeur externe
  - le manteau
  - la croûte
- chaque couche a les propriétés suivantes :
  - une profondeur maximum et une profondeur minimum (en km)
  - un nom
  - une densité (g/cm<sup>3</sup>)
- par exemple, la Terre est composée des couches suivantes (de l'intérieur vers l'extérieur) :
  - le coeur interne : profondeur 5100-6378 (profondeur minimum 5100 et profondeur maximum 6378), densité 12.8. Le nom de la couche est donc *coeur interne*
  - le coeur externe : profondeur 2890-5100, densité 9.9
  - le manteau : profondeur 35-2890, densité 3.4
  - la croûte : profondeur 0-35, densité 2.2
- l'affichage d'une planète doit afficher toutes les propriétés de ses couches
- **le nombre de couches pour modéliser une planète pourra varier dans un futur proche. Votre modélisation doit donc être assez souple pour gérer facilement une telle évolution**
- **les propriétés d'une couche pourront varier dans un futur proche. Votre modélisation doit donc être assez souple pour gérer facilement une telle évolution**

Nous supposons que toutes les planètes contiennent les couches précédentes, mais avec des propriétés différentes (une planète a donc obligatoirement les 4 couches précédentes).

Remarquez que la profondeur maximum du coeur interne nous donne le rayon de la planète.

### Question :

1. Implémenter votre solution de modélisation d'une planète
2. Proposer une modélisation permettant d'instancier facilement les planètes (donner un exemple avec la planète Terre)
3. Tester votre code via une classe interne. Le test consistera à instancier la planète Terre et à afficher sa composition (propriétés de chacune de ses couches)

## Exercice 2 : Gestion de planètes [6pt]

L'entreprise *PolytechAstro* veut ajouter de nouvelles fonctionnalités à son logiciel de gestion de planètes. Voici le cahier des charges :

- une classe `GestionPlanete` doit permettre de gérer un ensemble de planètes (pas de doublon dans l'ensemble - deux planètes sont considérées comme égales si elles ont exactement les mêmes propriétés pour chaque couche). L'interface de `GestionPlanete` doit permettre de :
  - afficher l'ensemble des planètes en les ordonnant du plus petit au plus grand rayon (voir interface `Comparable<T>`)
  - ajouter une planète
  - supprimer une planète (la planète à supprimer sera passée en paramètre de cette fonction)

### Question :

1. Implémenter votre classe `GestionPlanete`
2. Ajouter de nouvelles planètes (Jupiter, Mars... en mettant des propriétés au hasard pour chaque couche)
3. Tester votre code dans une classe `Test` en vérifiant que les doublons ne sont pas acceptés dans votre ensemble et que l'affichage est correctement trié

## Exercice 3 : Clonage & Sérialisation [6pt]

L'entreprise *PolytechAstro* veut ajouter de nouvelles fonctionnalités à son logiciel de gestion de planètes. Voici le cahier des charges :

- une planète doit être `Cloneable`
- la classe `GestionPlanete` permet de sauvegarder/charger la liste des planètes en les sérialisant/dé-sérialisant dans un fichier via deux nouvelles méthodes :
  - `savePlanete(File file)` : sérialiser l'ensemble des planètes dans le fichier `file`
  - `loadPlanete(File file)` : recharger l'ensemble des planètes à partir du fichier `file`

### Question :

1. Implémenter les nouvelles méthodes de `GestionPlanete`
2. Tester votre code dans votre classe `Test`
3. Commenter entièrement la classe `GestionPlanete` et générer la javadoc

## Exercice 4 : Questions de cours [2pt]

Créer un fichier `QuestionsDeCours.txt` dans lequel vous noterez vos réponses.

### Question :

1. En pratique, à quoi sert la liaison dynamique ? (donner des exemples si nécessaires)
2. En pratique, à quoi servent les génériques en Java ?