

Programmation Orientée Objet 1

Examen de Mi-session : 30 %

Hiver 2023

Durée de l'évaluation : 2 périodes (100 minutes)

Identification de l'étudiant

Nom : _____

Numéro d'étudiant : _____

Date : _____

Résultat : _____

Consignes

- Les cellulaires et l'accès à l'Internet ne sont pas permis pendant l'examen.
- **Les étudiants ne sont pas autorisés à communiquer entre eux.**
- Le plagiat, la tentative de plagiat ou la coopération à un plagiat lors d'une épreuve sommative entraîne la note zéro (0). Dans le cas de récidive, dans le même cours ou dans un autre cours, l'étudiant peut se voir octroyer un « 0 » pour le cours concerné.

Répartition des points

Cette évaluation est sur 100 points répartis comme suit :

- Question 1 60 points
- Question 2 40 points

TOTAL: 100 POINTS

Instructions :

Pour chaque question, créez un projet Eclipse nommé "MidTermQn" où n est le numéro de la question. Dans chaque question, vous devez créer un petit main (classe et fonction) pour effectuer un test très basique sur vos classes.

À la fin de l'examen, compressez tous ces dossiers de projet (y compris les fichiers Word / PDF de capture d'écrans) dans un seul fichier zip et téléchargez-le dans le devoir LEA pour l'examen de mi-session.

Question 1 : (60 points)

Vous devez coder une classe `Employe` pour un système RH. Les objets de cette classe doivent avoir des attributs :

1. `id` - un entier positif qui est un identifiant unique de l'employé
2. `nom` - le nom de famille de l'employé
3. `prenom` - le prénom de l'employé
4. `salaire` - un double supérieur ou égal à 0.

Remarque : Le nom et le prénom ne doivent contenir que des lettres latines minuscules et majuscules, espaces ou des "-".

A. Les attributs `id`, `nom` et `prenom` doivent être **immuables**.

B. Quatre cas d'exceptions doivent être correctement levés par la classe :

`EmployeeDataException` code 1 - `id` non valide (`id` inférieur ou égal à zéro)

`EmployeeDataException` code 2 - `nom` non valide

`EmployeeDataException` code 3 - `prenom` non valide

`EmployeeDataException` code 4 - `salaire` invalide (`salaire` inférieur à zéro)

L'`ExceptionDataException` doit avoir le message approprié pour chaque cas.

C. Les objets de la classe doivent être triables.

L'ordre par défaut entre les objets est l'ordre standard de `id` comme d'entiers.

Deux ordres secondaires doivent être fournies :

C.1. Ordre alphabétique par `nom` et `prenom` puis `id`.

C2. `Salaire` puis `id`.

D. Créez une classe `Main` pour vérifier que votre classe `Employe` marche bien.

E. Créez un fichier Word ou PDF avec les captures d'écran montrant que votre solution s'exécute correctement.

Question 2 : (40 points)

2.1 Représentez correctement les classes Cercle et Ellipse en respectant le principe de substitution de Liskov. Les objets de la classe Ellipse sont définis par deux attributs : semi-axe vertical (sav) et semi-axe horizontal (sah). Ces deux attributs doivent être du type double et plus grands que zéro. Les objets de la classe Cercle sont définis par un seul attribut : le rayon (r). Cet attribut doit être du type double et plus grand que zéro. Un cercle est une ellipse dont semi-axe vertical est égal au semi-axe horizontal. Les objets de ces deux classes doivent être mutables.

2.2 Les deux classes doivent offrir une méthode getAttributs() qui retourne la liste de ses attributs et une méthode toString() pour sur-écrire (override) la méthode toString(). Les deux classes doivent aussi avoir une méthode pour calculer l'aire de ces objets :

aire de l'ellipse = $\pi \cdot \text{sav} \cdot \text{sah}$ et

aire du cercle = $\pi \cdot r^2$.

Vous devez utiliser correctement une classe abstraite Conique pour organiser ces méthodes.

2.3 Toutes les exceptions doivent être correctement définies et jetées.

2.4 Créez un fichier Word ou PDF avec les captures d'écran montrant que votre solution s'exécute correctement.

Remarque: Le respect à toutes les consignes est fondamental. Si vous vous éloignez des consignes, votre note sera pénalisée.