

TP 1 Java – Classes et Héritage

Exercice 1 :

Ecrire une classe pour effectuer des conversions (dans les 2 sens) entre nombres sexagésimaux (durées en heures, minutes, secondes) et nombres décimaux (durées en heures décimales). Pour cela, réaliser une classe permettant de représenter une durée. Elle comportera :

Des champs privés représentant la valeur sexagésimale.

Un constructeur recevant 3 arguments de type « int » représentant une valeur sexagésimale qu'on supposera normalisée (secondes et minutes entre 0 et 59). Aucune limitation sur les heures.

Un constructeur recevant un argument de type "double" représentant une durée en heures.

Une méthode "getDecimal" fournissant la valeur en heures décimales associée à l'objet.

Des méthodes "getHeure", "getMinute" et "getSeconde" fournissant les trois composantes du nombre sexagésimal associé à l'objet.

Exercice 2 :

Le système d'informations de l'aéroport de Cardiff est en cours de refonte. Le DBA (Database Architect), à défaut d'un diagramme de classe, vous fournit les informations suivantes :

Un avion possède un matricule, un pays d'immatriculation, un type (commercial, tourisme...), et un aéroport d'attache représenté par sa ville. Chaque avion est décrit par un nom (Airbus A380, Boeing 747...), un poids (en tonnes), une capacité de passagers, une distance maximale, et une adresse ICAO.

Un technicien de la compagnie a un nom, un prénom, une date de naissance, un matricule, une adresse, un numéro de téléphone, et un salaire. De plus, il est assujéti à un examen médical annuel.

Chaque avion doit passer plusieurs tests techniques pour certifier de son bon fonctionnement. Chaque test technique possède un numéro d'identification, un libellé (test motorisation, test étanchéité réservoir...), et un seuil d'acceptabilité (valide, non valide).

Chaque vol est effectué par un commandant de bord et un copilote, et concerne un unique avion. Un vol possède un aéroport de départ (ville), et un aéroport d'arrivée (ville). Il est également caractérisé par une date et une heure de départ, et une date et une heure d'arrivée.

Implémentez ces informations en langage Java.

Utilisez une classe de test pour vérifier la cohérence de l'application.

Exercice 3 :

Créez une classe `FormeGeometrique` avec les méthodes suivantes :

- `calculerAire()`: Cette méthode permet de calculer l'aire de la forme géométrique.
- `afficherDetails()`: Cette méthode doit afficher des détails sur la forme géométrique, par exemple, son type.

Créez trois sous-classes de `FormeGeometrique` : `Rectangle`, `Cercle`, et `Triangle`.

Chacune de ces sous-classes devrait avoir un constructeur pour initialiser les propriétés spécifiques à chaque forme géométrique, par exemple, la longueur et la largeur pour un rectangle, le rayon pour un cercle, etc.

Implémentez la méthode `calculerAire()` dans chaque sous-classe pour calculer l'aire de la forme géométrique respective.

Implémentez la méthode `afficherDetails()` dans chaque sous-classe pour afficher des détails spécifiques à la forme géométrique.

Créez une classe `CalculAire` avec une méthode `main` pour tester vos classes. Dans cette classe, créez des instances de différentes formes géométriques, calculez leur aire, et affichez leurs détails.