

Travail Pratique 6 : Système de Gestion de Bibliothèque (POO)

Objectif

Utiliser l'héritage et le polymorphisme pour créer un système de gestion de bibliothèque.
Appliquer les concepts de surcharge (overloading), redéfinition (overriding), getters et setters.
Créer un diagramme UML pour illustrer la structure du programme.

Contexte

Vous allez concevoir un système pour gérer les livres et les abonnés d'une bibliothèque. Ce système doit inclure différentes catégories de ressources et types d'abonnés avec des méthodes spécifiques pour chaque type. Durée : 1h30.

Diagramme UML

Avant de commencer le codage, réalisez un diagramme UML de classes illustrant les relations suivantes :

- Une classe Ressource qui sert de classe parent.
- Des classes enfants Livre et Magazine qui héritent de Ressource.
- Une classe Abonne qui représente un abonné.
- Des classes enfants AbonneEnfant et AbonneAdulte qui héritent de Abonne.

Instructions de Codage

1. Classe Ressource

- Attributs privés : String titre et String auteur.
- Constructeur qui initialise titre et auteur.
- Getters et Setters pour titre et auteur.
- Méthodes : afficherInfos() et afficherInfos(boolean detail) (surcharge).

2. Classe Livre

- Hérite de Ressource.
- Attribut : String genre.
- Constructeur utilisant super.
- Getters et Setters pour genre.
- Redéfinition de afficherInfos().

3. Classe Magazine

- Hérite de Ressource.
- Attribut : String periodicite.
- Constructeur utilisant super.
- Getters et Setters pour periodicite.
- Redéfinition de afficherInfos().

4. Classe Abonne

- Attributs : String nom et int age.
- Constructeur pour nom et age.
- Getters et Setters pour nom et age.
- Méthode afficherInfos().

5. Classe AbonneEnfant

- Hérite de Abonne.
- Attribut : String categorie.
- Constructeur utilisant super.
- Getters et Setters pour categorie.
- Redéfinition de afficherInfos().

6. Classe AbonneAdulte

- Hérite de Abonne.
- Attribut : String abonnement.
- Constructeur utilisant super.
- Getters et Setters pour abonnement.
- Redéfinition de afficherInfos().

7. Classe Main

- Instanciez des objets de type Livre, Magazine, AbonneEnfant et AbonneAdulte.
- Test du polymorphisme sur les Ressources et les Abonnés.
- Test de la surcharge avec afficherInfos(boolean detail).

Exemple de Sortie Attendue

```
Livre: "Le Petit Prince", Auteur: Antoine de Saint-Exupéry, Genre: Fiction Magazine:  
"Science & Vie", Auteur: Rédaction, Périodicité: Mensuel Abonne Enfant: Alice, Âge:  
12, Catégorie : Élève Abonne Adulte: M. Dupont, Âge: 45, Abonnement : Annuel  
Affichage rapide (surcharge): Le Petit Prince
```

Questions pour aller plus loin

1. Pourquoi est-il utile de redéfinir afficherInfos() dans chaque classe ?
2. Expliquez comment le polymorphisme fonctionne dans ce TP.
3. Expliquez le rôle de chaque attribut et méthode dans le diagramme UML.
4. Quelle est la différence fondamentale entre la surcharge et la redéfinition de méthode, et où les avez-vous appliquées dans ce TP ?