

POO - TDTP 7

TD

Exercice 1

Quels résultats fournit le programme suivant ?

```
class A {
    public void affiche() {
        System.out.print("Je suis un A ");
    }
}

class B extends A {
}

class C extends A {
    public void affiche() {
        System.out.print("Je suis un C ");
    }
}

class D extends C {
    public void affiche() {
        System.out.print(" Je suis un D ");
    }
}

class E extends B {
}

class F extends C {
}

class Test {
    public static void main(String arg[]) {
        A a = new A(); a.affiche(); System.out.println();
        B b = new B(); b.affiche(); a = b; a.affiche(); System.out.println();
        C c = new C(); c.affiche(); a = c; a.affiche(); System.out.println();
        D d = new D(); d.affiche(); a = d; a.affiche(); c = d; c.affiche(); System.out.println();
        E e = new E(); e.affiche(); a = e; a.affiche(); b = e; b.affiche(); System.out.println();
        F f = new F(); f.affiche(); a = f; a.affiche(); c = f; c.affiche();
    }
}
```

Certaines possibilités d'affectation entre objets des types classes A, B, C, D, E et F ne figurent pas dans le programme ci-dessus. Pourquoi ?

Exercice 2

On prend les classes suivantes :

Etudiant, Personne, EtudiantTravailleur, Enseignant, EtudiantSportif et Travailleur.

1. dessinez une arborescence cohérente pour ces classes en la justifiant,
2. où se situeront les champs suivants : salaire, emploiDuTemps, anneeDEtude, nom, age et sportPratique, estAgregé.
3. Écrivez une méthode qui prend un tableau de personnes en argument et retourne la somme des salaires de ces personnes.

TP

Exercice 1 : Tester l'exo2 du TD

Exercice 2 : Bibliothèque

Pour la gestion d'une bibliothèque, on vous demande d'écrire une application traitant des documents de nature diverse : des livres, qui peuvent être des romans ou des manuels, des revues, des dictionnaires, etc.

Tous les documents ont un numéro d'enregistrement (un entier) et un titre (une chaîne de caractères). Les livres ont, en plus, un auteur (une chaîne de caractères) et un nombre de pages (un entier). Les romans ont éventuellement un prix littéraire (une chaîne de caractères, par exemple : Goncourt, Médicis, Femina, etc.), tandis que les manuels, en plus d'un auteur, ont un niveau scolaire (un entier). Les revues ont un mois et une année (des entiers) et les dictionnaires ont une langue (une chaîne de caractères convenue, comme "anglais", "allemand", "espagnol", etc.). Tous les divers objets en question ici (livres, revues, dictionnaires, romans, etc.) doivent pouvoir être manipulés en tant que documents.

Sur papier : proposez une hiérarchie pour les classes Document, Livre, Roman, Manuel, Revue, Dictionnaire. Dites dans quelle classe vous déclarez quel attribut.

Sur machine : on n'implémentera que les classes Document, Livre et Roman.

Écrivez un constructeur (ou plusieurs, si cela vous semble nécessaire) pour les classes Document, Livre et Roman. Les documents doivent être automatiquement numérotés en commençant par le numéro 1. Vous utiliserez `super(...)` dans les constructeurs des classes filles.

Écrivez pour les classes Document, Livre et Roman une méthode `toString()` produisant une description sous forme de chaîne de caractères des objets. Pensez à utiliser `super.toString()` dans les classes filles.

La classe `Bibliotheque` : une bibliothèque sera représentée par un tableau de documents.

Définissez une classe `Bibliothèque`, avec un tel tableau comme attribut et les méthodes et constructeurs :

- Un constructeur qui crée une bibliothèque ayant la capacité fournie en paramètre (nombre maximum de documents),
- Une méthode qui affiche tous les ouvrages de la bibliothèque,
- Une méthode qui renvoie le *i*ème document,
- Une méthode qui ajoute/supprime un document fourni en paramètre et qui renvoie `false` en cas d'échec.