

UEF4.3. Programmation Orientée Objet

EXAMEN FINAL

02 heures-Documents interdits

Questions de cours (4pts)

1. Soient A et B deux classes telles que B dérive de A. Comment se déroule la création d'un objet de type B ? (Citez les différentes étapes)
2. Le polymorphisme se base sur deux règles essentielles. Lesquelles ?

Exercice 1 (6pts)

Dans une bibliothèque universitaire, nous avons plusieurs types d'adhérents définis par les classes suivantes :

```
interface Contractuel{}
```

```
class Adherent {  
    public void emprunter() {  
        System.out.println("je peux  
        emprunter des livres");  
    }  
}
```

```
class Enseignant extends Adherent {  
    public void emprunter () {  
        System.out.println("la durée de mes  
        emprunts est de 15 jours");  
    }  
}
```

```
Class EnseignantPermanent extends  
Enseignant {  
    public void emprunter () {  
        System.out.println("je peux  
        emprunter 5 livres pour 15 jours  
        ");  
    }  
}
```

```
class EnseignantVacataire extends  
Enseignant implements Contractuel{  
    public void emprunter () {  
        System.out.println("je peux  
        emprunter 3 livres pour 15 jours  
        ");  
    }  
}
```

```
class Etudiant extends Adherent {  
    class EtudiantGraduation extends  
    Etudiant {  
        public void emprunter () {  
            System.out.println("je peux  
            emprunter 2 livres pour 10 jours  
            ");  
        }  
    }  
}
```

```
Class EtudiantPostGraduation  
extends Etudiant {  
}
```

```
Class EtudiantMagister extends  
EtudiantPostGraduation {  
    public void emprunter () {  
        System.out.println("je peux  
        emprunter 2 livres pour 15 jours  
        ");  
    }  
}
```

```
Class EtudiantDoctorat extends  
EtudiantPostGraduation implements  
Contractuel {  
    public void emprunter () {  
        System.out.println("je peux  
        emprunter 3 livres pour 15 jours  
        ");  
    }  
}
```


Question 1 : Qu'affiche le programme suivant ?

```
public static void main(String args[]) {
    Adherent P1 = new Adherent ();
    Adherent P2 = new EnseignantPermanent();
    Enseignant P3 = new EnseignantVacataire ();
    Adherent P4 = new EtudiantPostGraduation();
    Etudiant P5 = new EtudiantDoctorat ();
    EtudiantPostGraduation P6 = new EtudiantMagister();
    EtudiantGraduation P7 = new EtudiantGraduation();
    EtudiantPostGraduation P8 = new EtudiantDoctorat ();
    P1.emprunter(); P2.emprunter(); P3.emprunter(); P4.emprunter ();
    P5.emprunter(); P6.emprunter(); P7.emprunter(); P8.emprunter();
}
```

Question 2 : Dans la suite des instructions suivantes, indiquez pour chacune :

- si elle est correcte ou si elle provoque une erreur à la compilation ou à l'exécution.
- si on peut résoudre le problème de la compilation par un cast (et donnez ce cast), et si oui, s'il reste une erreur à l'exécution ou non.

```
Adherent a1 = new Adherent ();
Contractuel a2 = new EnseignantVacataire ();
Contractuel a3 = new EtudiantDoctorat ();
Etudiant a4 = new EtudiantGraduation ();
Etudiant a5 = a3;
a5 = a4;
a2 = a3;
Enseignant a6 = new Enseignant ();
a4 = a6;
EnseignantVacataire a7 = a6;
a6 = a2;
EtudiantPostGraduation a8 =(EtudiantPostGraduation) a4;
```

Exercice 3 (10pts) : Conception d'un système de gestion de fichier

On désire réaliser un programme permettant de gérer un système de fichiers composé d'un ensemble de fichiers, et disposant d'un espace de stockage disponible (cet espace est un entier initialisé à une valeur au départ). Ce système de gestion permet d'ajouter ou supprimer un fichier ou d'effectuer une recherche en utilisant le nom d'un fichier ou un mot clé. Le résultat de la recherche sera un tableau des fichiers trouvés.

Un fichier peut être un fichier simple ou un répertoire qui contient d'autres fichiers. Il est caractérisé par son emplacement, son nom, sa taille (de type entier), sa date de création et de modification, son état (lecture seule ou caché), et son extension (si ce n'ai pas un répertoire).

Un fichier simple peut être un exécutable (un programme), un document qui contient du texte ainsi que des indications de typographie (police de caractères et couleur), un fichier

texte brut (sans indication de typographie), un fichier multimédia (une image, un fichier audio ou un fichier vidéo), ou bien un fichier compressé (.zip, .rar, ...etc) codé selon un procédé qui le rend plus petit que le (les) fichier(s) d'origine non codé(s). Afin que les fichiers compressés puissent être exploitables par l'ordinateur, ils doivent fournir la méthode de décompression qui devra être utilisée pour les lire. De même, les fichiers audio et les fichiers vidéo, étant obtenus à partir d'un signal numérique en utilisant une méthode de compression, doivent fournir la méthode de décompression permettant à l'ordinateur de les lire.

Lorsqu'il s'agit d'un fichier simple, il est possible de connaître la liste de ses mots clés qui sera utilisée lors de la recherche. Dans le cas d'un répertoire, cette recherche est effectuée non seulement sur le nom du fichier mais aussi sur les noms des fichiers qui le composent (deux fichiers au niveau racine d'un répertoire ne peuvent pas avoir le même nom).

Questions

1. Quelle structure de données proposez-vous pour :
 - a. Conserver les fichiers dans le système, indexés par leurs emplacements.
 - b. Conserver les fichiers d'un répertoire triés par leurs noms.
 Justifiez vos réponses.
2. Proposez une modélisation orientée objet pour concevoir ce système de gestion de fichiers :
 - a. Tracez le diagramme des classes en spécifiant les différentes classes et relations avec les flèches adéquates : utilisation --->, héritage —> ou implémentation --->. Précisez le type de chaque classe (abstraite, interface, énumération).
 - b. Détaillez les attributs et les méthodes de chaque classe dans un tableau (Ne pas inclure les constructeurs et les accesseurs/modificateurs (getters et setters)). Le tableau devra avoir la forme suivante :

Nom de la classe	
(spécifiez si elle est abstraite, si elle dérive d'une autre classe ou implémente une interface même prédéfinie dans Java)	
Liste attributs	
Mode d'accès Type_attribut nom_attribut	Désignation
...	...
Liste des méthodes	
Mode d'accès Type_retour Signature_méthode	Désignation
...	...

3. Implémenter en Java le code de la méthode `ajouterFichier(Fichier f)` permettant d'ajouter un fichier au système. Nous supposons qu'une erreur sera lancée si la taille du fichier est supérieure à l'espace disponible.