

Nom :
Prénom :
Numéro d'étudiant :
Groupe :

Contrôle continu 3

Tous documents sur support papier autorisés. Durée : 1h15

L'ensemble des réponses sera à donner sur les feuilles d'énoncé. Ne pas dégrapher les feuilles.

La figure 1 vous présente un extrait d'une représentation des fichiers dans un système d'exploitation générique. L'interface `Int_Fichier` expose les opérations disponibles sur tous les fichiers. La classe `Abs_Fichier` comporte les informations et les comportements (parties de codes de méthodes) communs à tous les fichiers. Les objets de `Abs_Fichier` sont divisés selon le critère `TypeFichier` qui indique si un fichier est un conteneur (par exemple un répertoire) ou non. La classe `FichierSimple` représente les fichiers ordinaires (textes, images, liens symboliques, etc.). Ceux-ci sont décrits par une taille (en octets) et une extension (telle que ".jpg", ".txt", ".odg", etc.). La classe `Répertoire` représente les répertoires. Un répertoire contient des fichiers.

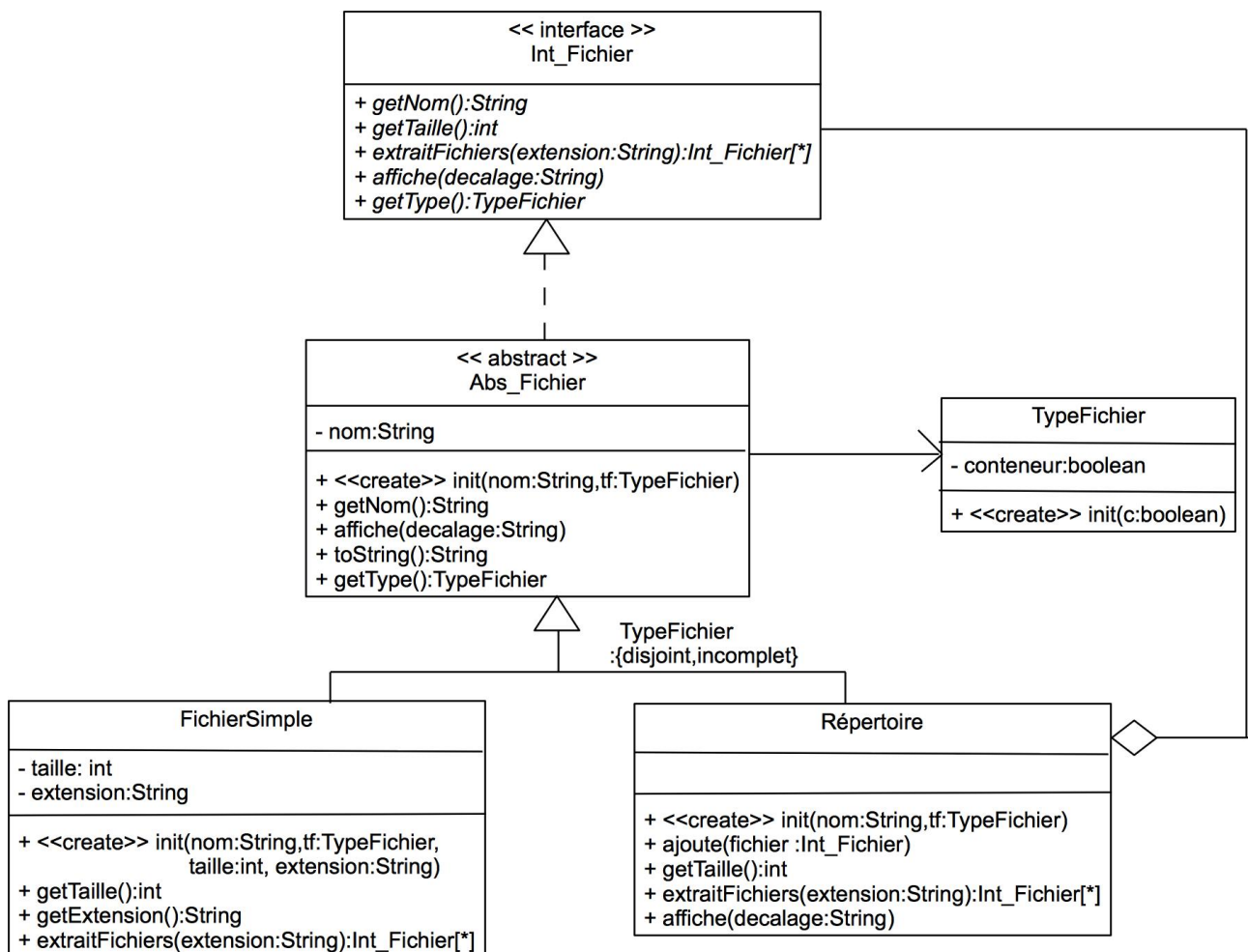


FIGURE 1 – Diagramme de classes à compléter

Question 1. Complétez le diagramme par les informations suivantes :

a- Représentez sur le diagramme le fait qu'un répertoire connaît les fichiers qu'il contient, mais qu'un fichier ne sait pas s'il est dans un répertoire.

b- ajoutez des noms de rôles et des cardinalités sur les associations. Ecrivez-les sur le diagramme et justifiez-les dans l'espace ci-dessous.

Question 2. En vous appuyant sur les contraintes indiquées sur le groupe de relations de spécialisation/généralisation, précisez si :

a- on peut ajouter une sous-classe « Fichier Lien » dans la spécialisation suivant le critère `TypeFichier` et pourquoi.

b- on peut ajouter une sous-classe instanciable commune à `FichierSimple` et `Répertoire` et pourquoi.

Question 3. Etant donné les instructions suivantes (les constructeurs sont construits de manière classique et suivent l'ordre de déclaration sur le diagramme) :

```
TypeFichier tfs = new TypeFichier(false);
TypeFichier trep = new TypeFichier(true);
Int_Fichier f1 = new FichierSimple("photoOurs",tfs,400,"jpg");
Int_Fichier f2 = new FichierSimple("photoFleur",tfs,300,"png");
Int_Fichier f3 = new FichierSimple("photoOcean",tfs,500,"png");
Int_Fichier f4 = new FichierSimple("photoEcume",tfs,200,"jpg");
Repertoire r1 = new Repertoire("r1",trep);
r1.ajoute(f1);    r1.ajoute(f2);
Repertoire r2 = new Repertoire("r2",trep);
r2.ajoute(f3); r2.ajoute(r1);  r2.ajoute(f4);
```

Proposez un diagramme d'instance présentant les objets construits, les liens et les attributs valués.

Question 4. Ecrivez l'interface `Int_Fichier`.

Question 5. Ecrivez l'entête de la classe `Abs_Fichier` et l'entête de la classe `FichierSimple`.

Question 6. Ecrivez l'entête et l'attribut de la classe `Repertoire`. Initialisez l'attribut.

Question 7. Ecrivez la méthode `getTaille` dans les classes où elle vous semble utile (appuyez-vous sur le diagramme). Pour les fichiers simples, elle retourne la valeur de l'attribut. Pour les fichiers répertoires, elle correspond à la somme des tailles des éléments contenus, à tous les niveaux d'imbrication. Par exemple, pour les objets créés précédemment, la taille de `r2` est 1400. Indiquez clairement dans quelle classe vous mettez quel code.

Question 8. Ecrivez la méthode `extraireFichier(String extension)`, partout où elle vous semble utile (appuyez-vous sur le diagramme). Pour les fichiers simples, elle retourne une liste contenant l'objet courant s'il a l'extension requise (sinon une liste vide). Pour les répertoires, elle retourne une liste contenant tous les fichiers du répertoire et de ses sous-répertoires jusqu'au plus profond niveau d'imbrication, qui ont l'extension requise. Par exemple, pour les objets créés précédemment, l'instruction `r2.extraireFichier("jpg")` retourne `[photoOurs, photoEcume]`. Indiquez clairement dans quelle classe vous mettez quel code. Suivant la manière dont vous vous y prenez, vous pouvez être amenés à fusionner des listes, pour cela vous pouvez utiliser la méthode `addAll(List<..> autreListe)` des classes `ArrayList` ou `Vector` (qui sont des `List`).

Question 9. Ecrivez la méthode `affiche()` qui affiche sur la console un fichier, partout où elle vous semble utile (appuyez-vous sur le diagramme). S'il s'agit d'un répertoire, le contenu est affiché de manière indentée. Par exemple, pour les objets créés précédemment l'affichage sera le suivant :

```
r2
  photoOcean
  r1
    photoOurs
    photoFleur
  photoEcume
```

Question 10. Proposez, dans la classe `Repertoire`, deux méthodes de tri des `Int_fichiers` du `Repertoire` :

- La première trie suivant les noms des fichiers par ordre alphabétique (ordre naturel),
- La seconde trie suivant le type de fichier en plaçant les conteneurs avant les non conteneurs.

On désire effectuer ce tri sur les éléments directement contenus dans le répertoire (donc pas récursivement sur les éventuels sous-répertoires). Mettez en place tout le code nécessaire pour le fonctionnement de ces méthodes de tri.