Examen 2014

Instructions générales

Les trois exercices seront rédigées sur trois feuilles séparées. Le seul document de cours autorisé est une feuille manuscrite de format A4 recto-verso. Les calculatrices et autres équipements électroniques (téléphones portables) ne sont pas autorisés. Le barème est donné à titre indicatif. Toute tricherie sera sévèrement sanctionnée. Durée : 1h45.

Exercice 1 : Spécifications algébriques (8 points)

Nous nous proposons de spécifier le type abstrait $Bibliothèque$ permettant de gérer une collection de livres. Ils sont rangés et accessibles uniquement d'après leur titre, ce qui implique que deux livres de cette bibliothèque ne peuvent avoir le même titre. La signature du type $Bibliothèque$ est la suivante :

$$\begin{aligned}
bibliothequeVide &: && \rightarrow Bibliotheque \\
emprunter &: Bibliotheque \times Titre && \rightarrow Bibliotheque \\
rendre &: Bibliotheque \times Titre \times Livre && \rightarrow Bibliotheque \\
existe &: Bibliotheque \times Titre && \rightarrow Booleen \\
nbLivres &: Bibliotheque && \rightarrow Entier
\end{aligned}$$

avec

- $emprunter$ qui retire le livre de la bibliothèque à partir de son titre,
- $rendre$ qui remet le livre dans la bibliothèque à partir de son titre,
- $existe$ qui indique si oui ou non le livre qui porte ce titre est présent dans la bibliothèque,
- $nbLivres$ qui indique le nombre de livres présents dans la bibliothèque.

(1.1) Quelles sont les opérations internes et les observateurs ?
(1.2) Écrivez les deux pré-conditions et les axiomes du type $Bibliotheque$.

Nous décidons d'ajouter un classement supplémentaire par auteur. Ainsi, les livres seront d'abord classés par auteur puis par titre pour chaque auteur. Ce nouveau classement permet d'avoir deux livre de même titre s'ils ont un auteur différent. Nous définissons ainsi un nouveau type abstrait $Bibliothek$ avec la signature devient la suivante :

$$\begin{aligned}
bibliothekVide &: && \rightarrow Bibliothek \\
emprunter &: Bibliothek \times Auteur \times Titre && \rightarrow Bibliothek \\
rendre &: Bibliothek \times Auteur \times Titre \times Livre && \rightarrow Bibliothek \\
existeAuteur &: Bibliothek \times Auteur && \rightarrow Booleen \\
existeTitre &: Bibliothek \times Auteur \times Titre && \rightarrow Booleen \\
nbLivres &: Bibliothek && \rightarrow Entier \\
nbLivresAuteur &: Bibliothek \times Auteur && \rightarrow Entier
\end{aligned}$$

avec

- $bibliothekVide$ qui crée une nouvelle bibliothèque vide,
- $emprunter$ qui retire le livre de la bibliothèque à partir de son auteur et de son titre,
- $rendre$ qui remet le livre dans la bibliothèque à partir de son auteur et de son titre,
- $existeAuteur$ qui indique si oui ou non la bibliothèque contient au moins un livre de cet auteur,
- $existeTitre$ qui indique si oui ou non la bibliothèque contient un livre de cet auteur avec ce titre,
- $nbLivresAuteur$ qui indique le nombre de livres de cet auteur présents dans la bibliothèque.

Examen 2014

(1.3) Écrivez les deux pré-conditions (similaires à celles du type $Bibliotheque$) et les axiomes du type $Bibliothek$.

Quelques questions de cours :

(1.4) Quels sont les deux propriétés fondamentales que doit vérifier l'ensemble des axiomes ?

(1.5) Pour chacune de deux propriétés, donnez la question à laquelle il faut répondre affirmativement pour qu'elle soit vérifiée ?

(1.6) Complétez le tableau ci-dessous en sachant que $reviser$ est un Getter et $tuer$ une opération interne :

Type abstrait	Classe (en *Java*)
Opérations	?
?	Getters
?	Note reviser(Etudiant e, Temps t) ;
Opérations internes	?
$tuer : Baratheon \times Stark \rightarrow Spoil$	?

Exercice 2 : Files (6 points)

La file est un cas particulier de liste, dans laquelle les adjonctions se font uniquement en queue de liste, les consultations et les suppressions uniquement en tête. La liste des opérations du type `File[T]` est donnée ci-dessous :

```
Type File [T]
Opérations
 vide : ⟶ File[T]                          -- créer une file vide
 ajouter : File [T] X T ⟶ File [T]         -- ajouter un élément en queue
 estVide : File [T] ⟶ booléen              -- file vide?
 tete : File [T] ⟶ T                       -- élément en tête
 longueur : File [T] ⟶ entier              -- nombre d'éléments
 supprimer : File [T] ⟶ File [T]           -- enlever la tête
```

Comme on l'a fait pour chaque structure de données étudiée en cours, on implante ce type en Java en construisant une hiérarchie de classes. L'interface `File<T>` définit la liste des opérations. Certaines d'entre elles sont écrites dans la classe abstraite `AbstractFile<T>` qui implémente cette interface. On construit deux implantations différentes `CircularFile<T>` et `LinkedFile<T>`.

(2.1) Dessiner le diagramme de classes UML incluant les classes/interfaces citées.

(2.2) Ecrire le code complet de la classe abstraite `AbstractFile<T>`.

(2.3) Dans l'implantation contigüe (`CircularFile<T>`), les éléments de la file sont rangés dans un tableau géré de façon circulaire. Dans l'exemple, les éléments sont successivement 66, -3, 21, 9, 6 et 1 (66 est la tête de file). Le tableau est alloué avec une taille fixée par un paramètre du constructeur. Lorsqu'il est plein, un tableau plus grand d'un tiers est alloué, et les éléments sont recopiés. Ecrire le code complet de la classe `CircularFile<T>`.

9	6	1				66	−3	21
0	1	2				6	7	8

FIGURE 1 – Exemple

Exercice 3 : Arbres binaires de recherche (6 points)

(3.1) Quelles sont les 2 contraintes vérifiées en chaque noeud d'un arbre binaire de recherche ?

(3.2) Quelles opérations peut-on mettre en place pour équilibrer un arbre binaire de recherche ?

(3.3) Construisez un AVL en ajoutant, dans l'ordre, les éléments suivants : 18, 98, 51, 70, 62, 57, 20, 65, 22, 3, 19. Vous présenterez les différentes étapes de construction de l'arbre sous la forme de schémas, en précisant les opérations d'équilibrage utilisées.

(3.4) L'arbre, ainsi construit, est-il la représentation d'un tas ? Pourquoi ?

(3.5) Construisez un tas, sous la forme d'un arbre, en ajoutant dans l'ordre les éléments suivants : 18, 98, 51, 70, 62, 57, 20, 65, 22, 3, 19. Vous présenterez les différentes étapes de construction du tas sous la forme de schémas, en précisant les opérations utilisées à chaque fois.