

Programmation 1

Examen

Le sujet de l'examen est un exercice sur la gestion de documents auxquels sont affectés des mots-clés, et la gestion de cet ensemble de mots-clés. La première partie porte sur la gestion d'un ensemble de mots-clés, et la deuxième partie sur la gestion de documents et leurs mots-clés.

Partie 1

Un mot-clé est représenté par une classe contenant le mot-clé et son nombre d'occurrences dans les documents.

Question 1 (2 points)

Définir les constructeur et méthodes de la classe *Element*. La méthode *toString()* retourne la chaîne de caractères "(" nombre d'occurrences ", " le mot-clé ")". La méthode *CompareTo(Element e)* retourne la valeur obtenue par comparaison des attributs *motCle*.

Question 2 (2 points)

Définir le constructeur et la méthode *toString()* de la classe *Ensemble*. Le constructeur a un paramètre indiquant le nombre maximum de mots-clés gérés par la classe. La méthode *toString()* retourne une chaîne de caractères qui contient tous les éléments de l'ensemble séparés par des "," et entre "[" et "]" .

Question 3 (3 points)

Définir les méthodes *ajouter* et *oter*. La méthode *ajouter* ajoute un mot-clé à l'ensemble. Si le mot-clé est déjà présent dans l'ensemble, son nombre d'occurrences est augmenté de 1, et il n'est pas ajouté. Si le tableau est plein, on lève une exception. La méthode *oter* enlève un mot-clé de l'ensemble. Oter un élément revient à diminuer son nombre d'occurrences de 1 ; le mot-clé n'est réellement enlevé de l'ensemble que lorsque son nombre d'occurrences devient égal à 0.

Question 4 (2 points)

Définir la méthode *listerParMC* qui retourne un tableau contenant les mots-clés, le tableau étant dans l'ordre lexicographique. (rappel la classe *Arrays* possède la méthode de classe *sort(t)* qui ordonne le tableau *t* en utilisant la méthode *compareTo* des objets du tableau)

Partie 2

Un document est constitué uniquement d'un nom et d'une liste de mots-clés attachés à ce document.

Question 5 (2 points)

Définir le constructeur et les accesseurs de la classe *Document*.

Question 6 (2 points)

Définir la méthode *possede* de la classe *Document* qui retourne *true* si un document possède le mot-clé en paramètre et *false* sinon.

La classe *EnsembleDocs* est constituée du tableau des documents, et d'un ensemble (*ens*) de mots-clés, qui est l'ensemble des mots-clés, munis de leurs nombres d'occurrences dans les différents documents.

Question 7 (2 points)

Définir le constructeur de la classe *EnsembleDocs*. Les paramètres indiquent la taille du tableau *lesDocs*, et la taille de l'ensemble. Lorsqu'on crée un ensemble de documents, celui-ci est vide.

Question 8 (2 points)

Définir la méthode *ajoute* qui ajoute un document. Le document est ajouté dans le tableau de documents, s'il y a de la place, et les mots-clés sont ajoutés dans l'ensemble *ens*. Si l'une de ces deux opérations n'est pas possible, on lève une exception *RuntimeException*.

Question 9 (3 points)

Définir la méthode *lister*, qui affiche à la console la liste des documents possédant tous les mots-clés donnés en paramètre de la méthode.

```
public class Element implements Comparable{
    private String motCle;
    private int nOcc;
    public Element(String motCle){...}
    public String toString(){...}
    public int compareTo(Element e){...}
    public String getMotCle(){...}
    public void setMotCle(String motCle) {...}
    public int getNOcc(){...}
    public void setNOcc(int occ) {...}
}
```

```
public class Ensemble {
    private Element[] lesElements;
    private int nElements;

    public Ensemble(int n){...}
    public void ajouter(String motCle){...}
    public void oter(String motCle) {...}
    public String[] listerParMC(){...}
    public String toString(){...}
}
```

```
public class Document {
    private String [] motsCle;
    private String nom;

    public Document(String nom, String [] motsCle) {...}
    public String[] getMotsCle(){...}
    public String getNom(){...}
    public boolean possede(String motCle) {...}
}
```

```
public class EnsembleDocs {
    private Document[] lesDocs;
    private int nDocs;
    private Ensemble ens;

    public EnsembleDocs ( int tDocs, int tEns) {...}
    public void ajoute(Document d) {...}
    public void lister(String ... motsCle) {...}
}
```