

TP MÉDIATHÈQUE : GESTION DES VIDÉOS À L'AIDE DE DICTIONNAIRE

La médiathèque de Beaucouzé propose déjà une location de vidéo(DVD) à ses adhérents. Elle souhaite créer une nouvelle application Android afin de gérer ce service de location par Tablettes. Ces tablettes seront présentes DANS la médiathèque.

Dans un premier temps, vous êtes chargé de concevoir les classes METIER.

La classe METIER **vidéo** est déjà créée et présentée en **annexe A**.

Pour gérer le tri et le rangement des vidéos, un code a été créé suivant la nomenclature suivante :

- 2 caractères pour les initiales du réalisateur
- 2 chiffres pour l'année de sortie du film
- 1 tiret et 1 caractère pour le nombre d'exemplaires du film

À vous de créer la classe **Médiathèque** et de gérer son dictionnaire.

PARTIE 1 : Travail à faire	
1	Consulter le cours C5 sur les Dictionnaires(HashMap) et le comprendre Étude de la classe vidéo <i>en Annexe</i>
2	La médiathèque souhaite lister toutes ces vidéos. a) Quel diagramme de classes proposez-vous ? b) Quel dictionnaire proposez-vous et pourquoi ? Créer un projet sous AndroidStudio nommé TPDicoVotrenom Récupérer les fichiers Video.java et Mediatheque.java et les mettre au bon endroit dans votre projet : \\StudioProjects\\TPDictionnaire\\app\\src\\main\\java\\com\\example\\frederiquederobien\\tpdictionnaire Récupérer le fichier MediathequeTest.java et le mettre au bon endroit dans votre projet : \\StudioProjects\\TPDictionnaire\\app\\src\\androidTest\\java\\com\\example\\frederiquederobien\\tpdictio nnaire
3	Écrire les différentes méthodes de la classe Médiathèque afin de gérer le dictionnaire (HashMap) : Écrire le constructeur de la classe Médiathèque
4	Écrire l'algorithme de la méthode <i>ajouterVideo (uneVideo : Video)</i> de la classe Médiathèque.
5	Écrire l'algorithme de la méthode <i>supprimeVideo (uneCle : TypeClé)</i> de la classe Médiathèque.
6	Écrire l'algorithme de la méthode <i>rechercheVideo(uneCle : TypeClé)</i> qui retourne la vidéo correspondante à la clé passée en paramètre.
7	Écrire l'algorithme de la méthode <i>consulterTitres()</i> de la classe Médiathèque qui retourne la liste des clés dans une nouvelle collection (nommée lesTitresVideos).
8	Écrire l'algorithme de la méthode <i>ConsulterVidéos()</i> de la classe Médiathèque qui retourne la liste des valeurs uniquement (ici l'objet Vidéo que vous pouvez afficher sous la forme d'un chaîne grâce à la méthode toString() déjà existante)
9	les tester grâce à la classe de Test nommée MediathequeTest

La médiathèque souhaite aussi que ses adhérents puissent consulter les livres identifiés par un numéro ISBN (ex: 978-2-09-255566-8).

Attention, les livres regroupent plusieurs sortes (BD, Revue, Manga etc...).

<u>PARTIE 2 : Travail à faire</u>	
10	Quel nouveau diagramme de classes proposez-vous ?
11	Quel dictionnaire ou collection proposez-vous et pourquoi ?
12	Compléter le projet (TPDictionnaire) en gérant les livres présents à la médiathèque afin de pouvoir consulter sur une tablette.
13	Compléter la classe de Test nommée MediathequeTest afin de vérifier vos différentes méthodes

Annexe A – classe Vidéo

```
import java.util.GregorianCalendar;
import java.util.Calendar;

public class Video implements Comparable {
    private String titre, realisateur;
    private int annee;

    public Video(){
        this("Inconnu", "Inconnu", 0);
    }

    public Video (String titre, String realisateur){
        this(titre, realisateur, (new
GregorianCalendar()).get(Calendar.YEAR));
    }
    public Video (String titre, String realisateur, String
annee){
        this(titre, realisateur, Integer.parseInt(annee));
    }
    public Video (String titre, String realisateur, int
annee){
        if (titre == null || realisateur == null)
            throw new NullPointerException();
        this.titre = titre;
        this.realisateur = realisateur;
        this.annee = annee;
    }

    public String getTitre(){
        return this.titre;
    }
    public String getRealisateur(){
        return this.realisateur;
    }
}
```

```
public int getAnnee() {
    return this.annee;
}

public void setTitre(String titre) {
    this.titre = titre;
}
public void setRealisateur(String realisateur) {
    this.realisateur = realisateur;
}
public void setAnnee(int annee) {
    this.annee = annee;
}

@Override
public int hashCode() {
    return annee * titre.hashCode() + realisateur.hashCode();
}

public boolean equals(Object o) {
    if(!(o instanceof Video))
        return false;
    Video v = (Video)o;
    return(this.hashCode() == v.hashCode());
}

public int compareTo(Object o) {
    if(!(o instanceof Video))
        throw new ClassCastException();
    Video v = (Video)o;
    int comparaison;
    if((comparaison = titre.compareTo(v.getTitre())) != 0)
        return comparaison;
    else if((comparaison =
realisateur.compareTo(v.getRealisateur())) != 0)
        return comparaison;
    else
        return (new Integer(annee)).compareTo(new
Integer(v.getAnnee()));
}

@Override
public String toString() {
    StringBuffer res = new StringBuffer("[");
    res.append(titre);
    res.append(", ");
    res.append(realisateur);
    res.append(", ");
    res.append(annee);
    return res.append("]\n").toString();
}
}
```