


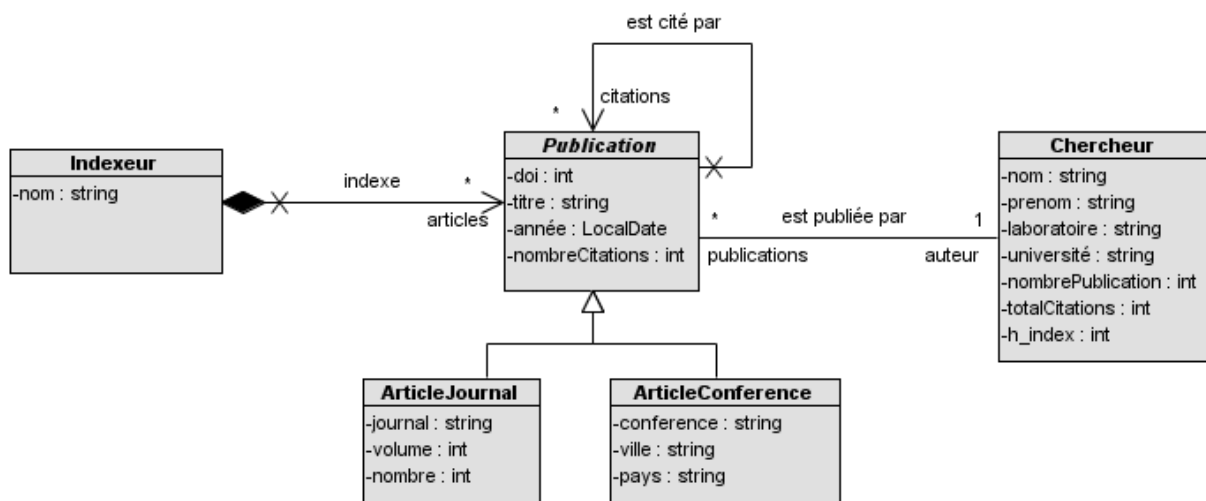
| | |
|--|---------------------|
|  FACULTE DES SCIENCES BEN M'SICK UNIVERSITÉ HASSAN II DE CASABLANCA | Nom : |
| | Prénom : |
| | Salle : |
| | Code apogée : |

Examen Java Session de Printemps 2024
SMI S6
Pr. Abdessamad Belangour

Nous souhaitons réaliser une petite application très simplifiée de gestion des publications scientifiques d'un chercheur. Ainsi, lorsqu'un chercheur arrive à des résultats scientifiques importants, il rédige un article et le publie soit dans une conférence, soit dans un journal scientifique (revue scientifique). Pour simplifier nous allons considérer qu'une publication ne dispose que d'un seul auteur.

Pour qu'une publication soit le plus visible possible, des sites bibliographiques spéciaux (appelés Indexeurs) répertorient, organisent et indexent ces publications provenant de diverses conférences et journaux à travers le monde. Ainsi chaque chercheur est identifié et son travail est diffusé et évalué à travers un ensemble d'indicateurs (nombre de publications, nombre de citations, H-index et autres). Un chercheur à un h-index de 10, par exemple, si au moins 10 de ses publications ont été cités 10 fois.

Le diagramme de classe de cette application est comme suit :



Compléter le code qui manque :

```

public class Indexeur {
    private String nom;
    private Map<Integer, Publication> articles;
    public Indexeur(String nom) {
        this.nom = .....;
        this.articles = .....;
    }
    public Indexeur(String nom, Map<Integer, Publication> articles) { // founi
    public Publication recupererPublication(int doi) {
        .....;
    }
}
  
```

```

public boolean ajouterPublication(Publication p) {
    .....
    .....
    .....
    .....
    .....
}
@Override
public String toString() {
    return "Nom Indexeur=" + nom + ", Nombre articles=" + .....;
}
}

```

```

// on suppose que les getters et setters sont fournis à part ceux qui sont demandés
public abstract class Publication implements Comparable<Publication>{
    private int doi; // document object identifier
    private String titre;
    private LocalDate année;
    private Chercheur auteur;
    private int nombreCitations;
    private Map<Integer,Publication> citations;
    // nouvelle publication non encore citée
    public Publication(int doi, String titre, LocalDate année, Chercheur auteur) {
        this.doi = doi;
        this.titre = titre;
        this.année = année;
        this.auteur = auteur;
        this.nombreCitations = ..... ;
        this.citations=.....;
    }
    // nouvelle publication citée
    public Publication(int doi, String titre, LocalDate année, Chercheur auteur, Map<Integer,
Publication> citations) {
        this.doi = doi;
        this.titre = titre;
        this.année = année;
        this.auteur = auteur;
        this.nombreCitations = ..... ;
        this.citations = ..... ;
    }
    public void setCitations(Map<Integer, Publication> citations) {
        ..... ;
        ..... ;
    }
    public boolean ajouterCitation(Publication pub){
        .....
        .....
        .....
        .....
        .....
    }
}

```



```

public class ArticleJournal extends Publication {
    private String journal;
    private int volume;
    private int nombre;
    // tout est fourni pour cette classe (Constructeurs, getters et setters, toString())
}

```

```

// on suppose que les getters et setters sont fournis
public class Chercheur {
    private String nom;
    private String prenom;
    private String laboratoire;
    private String université;
    private int nombrePublications;
    private int totalCitations;
    private int h_index;
    private Set<Publication> publications;
    // nouveau chercheur sans publications
    public Chercheur(String nom, String prenom, String laboratoire, String université) { // fourni}
    // nouveau chercheur avec publications
    public Chercheur(String nom, String prenom, String laboratoire, String université,
        Set<Publication> publications) {
        .....
        .....
        .....
        .....
        .....
        .....
        .....
    }
    public void ajouterPublication(Publication nouvellePublication) {
        .....
        .....
        .....
        .....
    }
    public void ajouterArticleJournal(String journal, int volume, int nombre, int doi, String titre,
        LocalDate année, Chercheur auteur, Map<Integer, Publication> citations) {
        .....
        .....
        .....
        .....
    }
    //donner le code de cette méthode en utilisant les streams
    public void ajouterArticleConference(String conference, String ville, String pays, int doi, String
        titre, LocalDate année, Chercheur auteur, Map<Integer, Publication> citations){

```

[illegible]