

# Contrôle intermédiaire 1 - 2017

---

**Durée 1 h 30.**

## Programme

Tout le début du cours jusqu'aux paquetages (compris). Les supports de cours distribués pendant le cours sont autorisés (mais **les TPs et corrections de TPs ne sont pas autorisés**).

## Exercice

Thème : Gestion des lecteurs d'une bibliothèque.

Une classe `Bibliotheque` placée dans le paquetage `eg.ufe.gestionlivres` représente une bibliothèque dans laquelle des lecteurs s'inscrivent.

Un lecteur est représenté par une instance de la classe `Lecteur`.

Voici le code de la classe `Lecteur` :

```
package eg.ufe; // pas le même paquetage que Bibliotheque

public class Lecteur {
    private String nom;
    private int age;

    public Lecteur(String nom, int age) {
        this.nom = nom;
        this.age = age;
    }

    public String getNom() {
        return nom;
    }

    public int getAge() {
        return age;
    }
}
```

## Question 1

Écrire la classe `Bibliotheque` avec :

- Une variable d'instance nommée `nom`, de type `String` qui représente le nom de la bibliothèque.
- Un getter (accesseur) pour la variable d'instance `nom`.
- Un constructeur qui prend en paramètre le nom de la bibliothèque et le nombre **maximum** de lecteurs qui pourront s'inscrire à la bibliothèque.
- Une variable d'instance `nbLecteurs` dont la valeur est le nombre de lecteurs actuellement inscrits à la bibliothèque.

- Un getter pour pour la variable d'instance nbLecteurs.
- Une variable d'instance lecteurs de type « tableau de lecteurs » qui contient les lecteurs actuellement inscrits dans la bibliothèque.
- Une méthode afficherNomsLecteurs qui affiche les noms de tous les lecteurs de la bibliothèque.

```
package eg.ufe.gestionlivres;

import eg.ufe.Lecteur;

public class Bibliotheque {
    private String nom;
    private int nbLecteurs;
    private Lecteur[] lecteurs;

    public Bibliotheque(String nom, int nbMaxLecteurs) {
        this.nom = nom;
        this.lecteurs = new Lecteur[nbMaxLecteurs];
    }

    public int getNbLecteurs() {
        return this.nbLecteurs;
    }

    public void afficherNomsLecteurs() {
        for (int i = 0; i < nbLecteurs; i++) {
            Lecteur lecteur = lecteurs[i];
            System.out.println(lecteur.getNom());
        }
    }
}
```

### Question 2

Dans la classe Bibliotheque écrivez une méthode inscrireUnLecteur pour inscrire un lecteur dans la bibliothèque. Le lecteur est passé en paramètre de la méthode.

Le lecteur n'est pas inscrit si le nombre maximum de lecteurs est déjà atteint. En ce cas la méthode ne fait rien.

```
public void inscrireUnLecteur(Lecteur lecteur) {
    if (nbLecteurs < lecteurs.length) {
        lecteurs[nbLecteurs] = lecteur;
        nbLecteurs++;
    }
}
```

### Question 4

Dans la classe Bibliotheque écrivez une méthode inscrireDesLecteurs pour ajouter dans le tableau des lecteurs tous les lecteurs passés en paramètre dans un tableau de lecteurs.

On suppose que le tableau passé en paramètre de la méthode est totalement rempli par des lecteurs (aucune valeur null dans le tableau).

Les lecteurs sont inscrits jusqu'à ce qu'il n'y ait plus de place dans la bibliothèque (on ne dépasse pas le nombre maximum de lecteur de la bibliothèque).

```
public void inscrireDesLecteurs(Lecteur... lecteurs) {
    for (Lecteur lecteur : lecteurs) {
        this.inscrireUnLecteur(lecteur);
    }
}
```

### Question 5

Écrire une méthode `selectionLecteursParAge` qui retourne un tableau qui contient les lecteurs inscrits dans la bibliothèque et qui ont un âge supérieur à l'âge passé en paramètre (de type `int`).

Le tableau retourné ne doit avoir aucune valeur null et vous devrez donc lui donner la bonne longueur.

```
public Lecteur[] selectionLecteurParAge(int ageMin) {
    Lecteur[] resultat = new Lecteur[nbLecteurs];
    int n = 0;
    for (int i = 0; i < nbLecteurs; i++) {
        Lecteur lecteur = lecteurs[i];
        if (lecteur.getAge() > ageMin) {
            resultat[n] = lecteur;
            n++;
        }
    }
    if (n < nbLecteurs) {
        // Construit tableau à retourner
        Lecteur[] t = new Lecteur[n];
        for (int i = 0; i < n; i++) {
            t[i] = resultat[i];
        }
        resultat = t;
    }
    return resultat;
}
```

### Question 4

Écrire une classe `TestBibliotheque` qui est dans le paquetage `eg.ufe.test` qui teste la classe `Bibliotheque` avec une méthode `main` qui fait ceci :

- Créer une bibliothèque qui peut avoir jusqu'à 100 lecteurs.
- Ajouter à la bibliothèque un lecteur.
- Ajouter 2 autres lecteurs contenus dans un tableau, en utilisant la méthode `inscrireDesLecteurs`.
- Afficher les noms des lecteurs qui ont plus de 30 ans.

```
package eg.ufe.test;

import eg.ufe.Lecteur;
import eg.ufe.gestionlivres.Bibliotheque;

public class TestBibliotheque {
    public static void main(String[] args) {
        Bibliotheque bibli = new Bibliotheque("Naguib Mahfouz", 100);
        bibli.inscrireUnLecteur(new Lecteur("Mohamed", 35));
        Lecteur[] lecteurs = {
            new Lecteur("Sarah", 28),
            new Lecteur("Salma", 40)
        };
        bibli.inscrireDesLecteurs(lecteurs);
        System.out.println("Tous les lecteurs");
        bibli.afficherNomsLecteurs();
        System.out.println("Les lecteurs qui ont plus de 30 ans");
        Lecteur[] selection = bibli.selectionLecteurParAge(30);
        for (Lecteur lecteur : selection) {
            System.out.println(lecteur.getNom());
        }
    }
}
```