

Ecole Supérieure Privée Technologies & Ingénierie

Type d'épreuve	: <input type="checkbox"/> Devoir	<input checked="" type="checkbox"/> Examen SESSION PRINCIPALE
Enseignant	: A. Arfaoui / Ben Chiba Mohamed Amine	
Matière	: JAVA	
Année Universitaire	: 2023-2024	Semestre : 1
Classe	: TIC-1	
Documents	: <input type="checkbox"/> Autorisés	<input checked="" type="checkbox"/> Non autorisés
Date	: 22/07/2024	Durée : 1h30mn
Nombre de pages	: 03	
Barème	:	

QCM :

Cochez-la ou les bonnes réponses

1. Que signifie le mot clef "static" associé à un attribut ?
 - a. Que la valeur de cet attribut est constante
 - b. Que cet attribut n'est visible que dans la classe où il est défini
 - c. Que cet attribut sera toujours passé par valeur
 - d. Que cet attribut a une valeur unique pour toutes les instances de la classe
2. Au sujet du mot clef "abstract", quelle assertion est fausse ?
 - a. Une classe abstraite ne peut être instanciée
 - b. Une méthode abstraite n'a pas d'implémentation
 - c. Une classe abstraite n'a pas forcément de classe fille
 - d. Une classe abstraite doit contenir au moins une méthode abstraite
3. Peut-on utiliser plusieurs blocs catch après un bloc try ?
 - a. Vrai
 - b. Faux
4. Quelle instruction faut-il utiliser pour créer un nouveau tableau d'entiers de type int de 5 cases ?
 - a. `Array t = new Array ("int", 5);`
 - b. `int t = new int[5];`
 - c. `int[] t = new int[5];`
 - d. `int[] t = new int[4];`
 - e. `Array t = new int[5];`

5. Un attribut statique est aussi appelé :
 - a. Variable d'instance
 - b. Variable de classe
 - c. Variable d'interface
 - d. Variable locale
6. Quels keywords sont utilisés pour spécifier la visibilité des propriétés et des méthodes ?
 - a. final
 - b. private
 - c. abstract
 - d. protected
 - e. public

Exercice 1 :

1. Pourquoi Java n'utilise-t-il pas l'héritage multiples ?
2. Donner quatre concepts fondamentaux de la programmation orientée objet.

Exercice 2 :

Remarque : Pour chacune des classes, ne définir les accesseurs et les mutateurs que dans le cas de besoin.

On désire créer une application qui permet de gérer les documents dans une bibliothèque. Deux types de documents sont disponibles : les livres et les rapports.

- Un **Emprunteur** est caractérisé par :
 - Un identifiant (chaîne de caractère)
 - Un nom (chaîne de caractères),
 - Un prénom (chaîne de caractères)
- Un **Document** est caractérisé par :
 - Un code (chaîne de caractère)
 - Un titre (chaîne de caractères)
 - Le nombre de copies disponibles (entier)
- Un **Livre** est un document qui est caractérisé par :
 - Un Numéro d'édition (entier)

- Une Date de la publication (chaîne de caractères)
- Un **Rapport** est un document caractérisé par :
 - Un président de jury (chaîne de caractères)
 - Un rapporteur (chaîne de caractères)
- Une **Bibliothèque** est caractérisée par :
 - L'ensemble des documents
 - Les emprunts

Une bibliothèque permet de gérer l'ensemble des emprunts c'est-à-dire qu'elle permet d'associer un emprunteur à un document. Un emprunteur ne peut emprunter qu'un seul document. Toutefois, un document peut être emprunté à plusieurs emprunteurs (autant de fois qu'il existe des copies disponibles).

1. Ecrire les classes **Bibliothèque**, **Emprunteur**, **Document**, **Livre** et **Rapport** en respectant les règles d'encapsulation.
2. Dans chacune de ces classes, définir les constructeurs adéquats.
3. Redéfinir la méthode "toString" dans les classes "**Document**", "**Livre**", "**Rapport**" et "**Emprunteur**".
4. Implémenter les méthodes suivantes, dans la classe "Bibliothèque":
 - a. La méthode **ajouterDocument** qui permet d'ajouter un nouveau document à la bibliothèque.
 - b. La méthode "**boolean rechercher** (Document D) " qui permet de rechercher un document dans une bibliothèque.
 - c. La méthode "**boolean emprunter_livre**(Emprunteur E, Document D)" qui ne permet d'emprunter un document D à un emprunteur E que s'il existe des copies disponibles du document. Cette méthode doit retourner vrai si l'emprunt a eu lieu et faux sinon (**Vous pouvez ajouter une classe en cas de besoins**)
 - d. La méthode "**afficherDocuments**" qui permet d'afficher tous les documents de la bibliothèque.
 - e. La méthode "**afficherLesEmprunts**" qui permet d'afficher tous les emprunts.
 - f. La méthode "**EmprunteursTries** " qui permet d'afficher la liste de tous les emprunteurs triés par l'ordre alphabétique de leurs noms.
5. Ajouter dans la classe "Test " un programme principal qui permet de :
 - a. Créer deux emprunteurs E1 et E2.
 - b. Créer deux documents : un livre disponible en 2 copies et un rapport disponible en une seule copie. Ajouter les deux documents à une bibliothèque.

- c. Afficher tous les documents de la bibliothèque.
- d. Emprunter à E1 une des copies du livre et à E2 la seule copie du rapport.
- e. Afficher l'état des emprunts.
- f. Trier et afficher les emprunteurs.