DEVOIR SURVEILLE

Matière	**:** Programmation Orientée Objets	**Groupe**	**:** II3
Enseignante :	**Mlle I.Sghaier**	🕐	**:** 1H
🕮	**:** Non autorisés	**Nb. Pages:**	2

Exercice1: (10 pts: 5+5)

1 ère partie:

1. Créer une classe « employe» correspondante au schéma UML suivant. La méthode decrisToi() affiche à l'écran le nom et le prénom de l'employé

Class employe
- String nom - String prenom - Double Salaire - int NCI -
+ Employe() + Employe(int n) + Employe(String nom,String prenom) + Employe(String nom,String prenom,int n, double s) + decrisToi() + String getNom() + String getPrenom() + Double getSalaire() + Int getNCI()

2. Créer une classe equipe qui utilise la classe Employe. La fonction main de Equipe comprendra la création d'un tableau de quatre employés et l'affichage de leurs noms

3. Peut-on modifier la variable d'instance nom depuis l'extérieur de la classe ? et depuis l'intérieur ?

2ème Partie :

1. Créer une classe technicien qui hérite de la classe employe. Celle-ci comprendra une variable d'instance supplémentaire : specialite.
    a. Ecrire des constructeurs : sans arguments et à 1 seul argument « String spécialité ». Vous utiliserez les constructeurs de la classe mère.
    b. Redéfinisser la méthode decrisToi(), en faisant appel à la méthode de la classe mère. (décris toi va afficher à l'écran : nom, prénom, spécialité).

2. En changeant les modificateurs d'accès des données membres de la classe mère, de private vers protected peut-on accéder à ces variables de l'extérieur de la classe Employe? Quel danger cela présente t-il en termes d'encapsulation ?
3. Expliquer le mécanisme de construction d'une classe dérivée
4. Ecrire une classe, qui teste la classe technicien.

Exercice 2: (10 pts : 6+4)

1. Ecrire une classe appelée « FonctionMath » qui contient les méthodes **statiques** suivantes :
   - Factoriel : c'est une méthode qui permet de calculer et d'afficher le factoriel d'un entier passé lors de l'exécution.
   - RacineCarré : C'est une méthode qui permet de calculer et d'afficher la racine carrée d'un nombre n passé lors de l'exécution
   - BinaryCode : C'est une méthode qui permet de lire un entier sur la ligne de la commande et afficher sa correspondance en binaire
   - HexCode : C'est une méthode qui permet de lire un entier sur la ligne de la commande et afficher sa correspondance en Hexadécimal.

NB : Pour les trois dernières méthodes utiliser les méthodes suivantes de la classe **Integer** : *toBinaryString et toHexString*.

2. Ecrire une classe «TestFonctionMath» qui utilise ces fonctions mathématiques.