esprit
Ecole Supérieure Privée d'Ingénierie et de Technologies

APP 2: Programmation orientée objet(JAVA)

Gestion d'un Laboratoire de Recherches (2)

Objectifs :

- ✓ Manipuler les instances de classes
- ✓ Utiliser les meilleures pratiques « Convention de Nommage »
- ✓ Faire la Surcharge des constructeurs
- ✓ Utiliser le mot clé« this », « static »,….
- ✓ Etablir des relations entre les classes
- ✓ Manipuler les tableaux

Etude de cas : Gestion d'un Laboratoire de Recherches (Même énoncé)

Dans un monde en constante progression technologique, le laboratoire de Recherche et Développement d' **«Intuitiv Tech»** est un véritable outil de veille révolutionnaire, permettant d'offrir aux clients des solutions toujours avant-gardistes et plus que jamais innovantes. Ce laboratoire offre une maîtrise parfaite des Nouvelles Technologies et renforce son positionnement.

Dans le cadre de gestion du laboratoire **«Intuitiv Tech»**, on se propose de développer un programme, en utilisant le langage JAVA, pour gérer les différentes données de ce laboratoire ; créer des bureaux pour diverses spécialités et créer aussi des chercheurs, les affecter chercheurs aux différents bureaux,…

Afin de créer un programme pour gérer le laboratoire **«Intuitiv Tech»**, nous allons considérer les points suivants :

- ✓ Un laboratoire qui est caractérisé par son nom, sa spécialité.
- ✓ Un laboratoire est composé par plusieurs de bureaux (50 bureaux maximum)
- ✓ Un laboratoire dispose aussi d'une adresse
- ✓ Une adresse est composée par un gouvernorat, une ville et un code postal
- ✓ Un chercheur est caractérisé par son nom, son poste et le numéro de son ordinateur.
- ✓ Un bureau est identifié par son code, son nom et il contient un ensemble de chercheurs,
- ✓ Chaque bureau peut contenir jusqu'à 5 chercheurs.

Travail à faire : (voir support du cours)

Question 1 :

Dessiner le diagramme de classe de ce problème en se basant sur le projet détaillé ci-dessus.

Question 2 :

Ajouter une autre méthode **comparer** qui permet de comparer entre 2 chercheurs cad tester l'égalité entre le nom, le numOrdinateur et le poste.

public boolean comparer(Chercheur c1, Chercheur c2)

Question 3 :

Créer dans la classe **Bureau** :

1. une méthode qui permet d'ajouter un chercheur au bureau

public boolean ajouterChercheur(Chercheur chercheur)

NB : il ne faut pas dépasser le nombre maximal de chercheurs par bureaux

2. une méthode qui vérifie l'appartenance d'un chercheur au bureau.

public int trouverChercheur(Chercheur chercheur)

3. une méthode pour supprimer un chercheur de son bureau.

boolean supprimerChercheur(Chercheur chercheur)

4. La méthode qui retourne le bureau ayant un nombre supérieur de chercheurs

public static Bureau compare (Bureau bureau1, Bureau bureau2)

Question 4 :

Dans la classe principale « Test »:

1. Créer un laboratoire *« lab1»* en lui affectant l'adresse créée *«ad1»*
2. Créer deux bureaux *«bureau11»* et *« bureau12 »*,
3. Affecter les deux bureaux au laboratoire *« laboratoire1»*.
4. Affecter les deux chercheurs *«ch1»* et *«ch2»* au bureau *«bureau11»*
5. Affecter le chercheur *«ch3»* au deuxième bureau *«bureau12»*.
6. Afficher les différentes données d'un bureau *«bureau11»*, ainsi les chercheurs qui y appartiennent.
7. Afficher le bureau ayant un nombre supérieur de chercheurs.
8. Réécrire la méthode ***comparer(Chercheur c1, Chercheur c2)*** de la classe Chercheur en utilisant la méthode **equals,** (faire le même traitement pour la classe Adresse).

   NB : Utiliser la méthode **equals** de la classe **Object**

Question 5 :

Remplacer, au niveau des attributs de la classe «Adresse», le mot clé «**public**» par « **private**», qu'est ce vous remarquez ?