Université Abou Bakr Belkaïd – Tlemcen - Faculté des Sciences - Département d'Informatique

2017- 2018 **2ème année Informatique**

Test - POO

Documents, Smartphones, connexion, … interdits **Durée : 1h30'**

Dans le département d'Informatique, on trouve deux catégories de personnes : des enseignants et des étudiants. Chaque personne est caractérisée par son **nom** et son **âge**. L'attribut **nbPersonnes** comptabilise le nombre de Personne du département.

On définit les méthodes public suivantes de la classe **Personne** :

- Le constructeur **Personne** (String nom, int age)
- **String toString** () : retourne une chaîne de caractères correspondant aux caractéristiques d'une personne.
- **afficherPersonne** () : pourrait afficher les caractéristiques d'une personne, seulement, il faut commencer par afficher la catégorie de la personne ("Enseignant : …" ou "Etudiant : …") qui n'est pas encore connue.
- **static void nbPersonnes** () : affiche le nombre total de personnes et le nombre de personnes par catégorie (Enseignant et Etudiant).

Un **Enseignant** est une Personne, enseignant une **matière** (POO, Algo, SI, etc.). L'attribut **nbEnseignants** compte le nombre d'enseignants créés. Une méthode static **nbEnseignants()** retourne sa valeur.
.On définit les méthodes public suivantes pour Enseignant :

- Le constructeur **Enseignant** (String nom, int age, String matière)
- String **toString** () : fournit une chaîne contenant les caractéristiques d'un enseignant (ou d'un Etudiant).
- **afficherPersonne** () : écrit "Enseignant : " suivi des caractéristiques d'un Enseignant.

Un **Etudiant** est une Personne préparant un **diplôme** (Master Info., Licence Info., etc.). L'attribut **nbEtudiants** compte le nombre d'étudiants créés. Une méthode static **nbEtudiants ()** retourne sa valeur.
.On définit les méthodes public suivantes pour Etudiant :

- Le constructeur **Etudiant** (String nom, int age, String diplôme)
- String **toString** () : fournit une chaîne contenant les caractéristiques d'un Etudiant.
- **afficherPersonne** () : écrit "Etudiant : " suivi des caractéristiques d'un Etudiant.

- La classe **PPPersonne** crée un Enseignant et deux étudiants :

```
Personne reda = new Enseignant ("Reda", 25, "POO");
Personne e1 = new Etudiant ("Brahim", 19, "licence info");
Personne e2 = new Etudiant ("Karim", 20, "Master info");
```

- Afficher le nombre de personnes, d'enseignants et d'étudiants créés.
- Remplir un tableau de personnes par ces objets

```
Personne[] personnes = {reda, e1, e2};
```

- Afficher les personnes du tableau en utilisant la méthode afficherPersonne()
- Modifier la matière de reda : "POO" par "Algo."
- Afficher de nouveau les personnes du tableau

NB. : Tous les attributs seront déclarés private.

```
Nombre d’employés : 3
Nombre d’enseignants : 1
Nombre d’étudiants : 2
 ----------------------------
Enseignant : Reda 25
 spécialité : POO
Etudiant : Brahim 19
 Diplome en cours : licence info
Etudiant : Karim 20
 Diplome en cours : Master info


Après modification:
 ----------------------------
Enseignant : Reda 25
 spécialité : Algo.
Etudiant : Brahim 19
 Diplome en cours : licence info
Etudiant : Karim 20
 Diplome en cours : Master info
```

```
abstract class Personne { // classe abstraite, non instanciable  (06 points)        0.5
        protected String nom;
        protected int age;
        protected static int nbPersonnes = 0; // nombre de Personne                  1
        // constructeur de Personne
        Personne (String nom, int age) {
                this.nom = nom;
                this.age = age;
                nbPersonnes++;                                                      1
        }
        // fournir les caractéristiques d'une Personne sous forme  d'un objet de la classe String
        public String toString() {
                return nom + " " + age;                                             0.5
        }
        // méthode abstraite (à redéfinir dans les classes dérivées)
        abstract void afficherPersonne ();
        static void nbPersonnes () {                                                1
                System.out.println ( "\nNombre d'employés : " + nbPersonnes +
                                "\nNombre d'enseignants : " + Enseignant.nbEnseignants() +   2
                                "\nNombre d'étudiants : " + Etudiant.nbEtudiants()   );
        }
} // Personne
```

```
class Etudiant extends Personne { // héritage de classe  (04 points)              0.5
    private String diplomeEnCours;
    private static int nbEtudiants = 0; // nombre d'Etudiant                       0.5
    Etudiant (String nom, int age,    String diplomeEnCours) {
            super (nom, age); // appel du constructeur Personne                    1
            this.diplomeEnCours = diplomeEnCours;
            nbEtudiants++;
    }
    public String toString () {
            return super.toString() + "\n Diplome en cours : "    + diplomeEnCours;   1
    }
    void afficherPersonne () {
            System.out.println ("Etudiant : " + toString());                       0.5
    }
    public String diplomeEnCours () {
       return diplomeEnCours;
    }
    static int nbEtudiants () { return nbEtudiants; }                               0.5
}
```

```
class Enseignant extends Personne { // héritage de classe (05 points)
    private String matière;
    private static int nbEnseignants = 0; // nombre d'Enseignant
    Enseignant (String nom, int age,    String matière) {
        super (nom, age); // appel du constructeur Personne
        this.matière = matière;
        nbEnseignants++;
    }
    void setMatiere(String matière){
        this.matière=matière;
    }
    public String toString () {
        return super.toString() + "\n matière : " + matière;
    }
    void afficherPersonne () {
        System.out.println ("Enseignant : " + toString());
    }
    static int nbEnseignants () { return nbEnseignants; }
} // Enseignant
```

0.5

0.5

1

1

1

0.5

0.5

```
class PPPersonne { // Programme Principal Personne  (05 points)
   public static void main(String[] args) {                                    0.5

      Personne reda = new Enseignant ("Reda", 25, "POO");
      Personne e1 = new Etudiant ("Brahim", 19, "licence info");
      Personne e2 = new Etudiant ("Karim", 20, "Master info");

      Personne.nbPersonnes(); // méthode static                                 1
      System.out.println ("-----------------------------");
      Personne[] personnes={reda, e1,e2};
      for (Personne p:personnes)                                               1
         p.afficherPersonne();

      System.out.println ("\n\nAprès modification :");
      System.out.println ("-----------------------------");
      ((Enseignant) reda).setMatiere("Algo.");                                 2
      for (Personne p:personnes)                                               0.5
         p.afficherPersonne();



   } // main
                                                  }
```