

## Solution TD2 Programmation Orientée Objet

---

### Exercice 1

Soit la classe *Personne* suivante :

```
class Personne
{
    private String nom , prenom ;
    private int age;

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

    public void informations()
    {
        System.out.println ("Je suis : " + nom + " " + prenom );
        System.out.println ("Age de : " + age );
    }
}
```

1. Réaliser une classe *Etudiant*, dérivée de *Personne* permettant de manipuler des personnes définies par leurs *nom*, *prénom*, *age* et une *spécialité*. La classe *Etudiant* fait étendre la classe *Personne* en ajoutant les méthodes suivantes :
  - A. Un constructeur *Etudiant()* pour définir les informations et la spécialité d'un objet de type *Etudiant*.
  - B. Une méthode *informationsEtudiant()* pour afficher les informations et la spécialité d'un objet de type *Etudiant*.
2. Écrire une classe *Application* utilisant la classe *Etudiant* permettant **d'instancier** un étudiant, puis **afficher** les attributs de l'étudiant.

### Solution :

```
class Personne
{
    private String nom , prenom ;
    private int age;

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

    public void informations()
    {
```

```

        System.out.println ("Je suis : " + nom + " " + prenom );
        System.out.println ("Age de : " + age );
    }
}

class Etudiant extends Personne
{
    private String specialite;

    public Etudiant(String nom , String prenom, int age, String specialite)
    {
        super(nom,prenom,age); //Appel du constructeur de la super-classe Personne
        this.specialite = specialite;
    }

    public void informationsEtudiant()
    {
        super.informations();
        System.out.println ("Spécialité est : " + specialite );
    }
}

public class Exo1 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        Etudiant e = new Etudiant("Said", "BenYoucef", 20, "Informatique");
        e.informationsEtudiant();
    }
}

```

## Exercice 2

Soit la classe suivante :

```

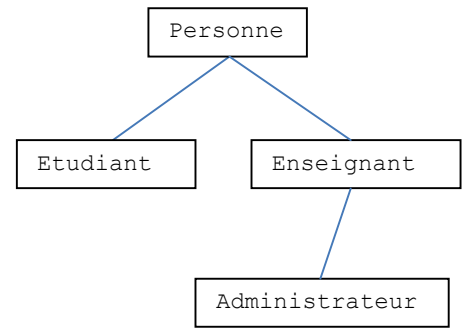
abstract class Personne
{
    private String nom;
    private String prenom;
    public Personne(String nom,String prenom)
    {
        .....
        .....
    }
    public String getNom()
    {
        .....
    }
    public String getPrenom()
    {
        .....
    }

    abstract public String getInformationAdditionnel();
    abstract public void afficherInformations();
}

```

1. Continuer le code manquant dans la classe Personne.

2. Suivant le schéma qui montre la relation d'héritage entre les 4 classes, **donner le code optimal** des classes Etudiant, Enseignant, Administrateur tel que :
- La classe Etudiant possède un attribut spécialité, un constructeur et les méthodes manquantes.
  - La classe Enseignant possède un attribut matière, un constructeur et les méthodes manquantes.
  - La classe Administrateur possède un attribut poste, un constructeur et les méthodes manquantes.



3. Ecrire une classe Application qui permet de créer 3 objets différents dans un seul tableau, puis afficher les informations de chaque objet.

### Solution :

```

abstract class Personne
{
    private String nom;
    private String prenom;
    public Personne(String nom,String prenom)
    {
        this.nom = nom;
        this.prenom = prenom;
    }
    public String getNom()
    {
        return nom;
    }
    public String getPrenom()
    {
        return prenom;
    }

    abstract public String getInformationAdditionnel();
    abstract public void afficherInformations();
}

class Etudiant extends Personne
{
    private String specialite;

    public Etudiant(String nom,String prenom, String specialite)
    {
        super(nom,prenom);
        this.specialite = specialite;
    }

    public String getInformationAdditionnel()
    {
        return specialite;
    }

    public void afficherInformations()
    {
        System.out.println("Nom : "+super.getNom()+", Prenom : "+super.getPrenom()+",
Specialite : "+specialite);
    }
}

class Enseignant extends Personne
{
    private String matiere;
  
```

```

public Enseignant(String nom,String prenom, String matiere)
{
    super(nom,prenom);
    this.matiere = matiere;
}

public String getInformationAdditionnel()
{
    return matiere;
}

public void afficherInformations()
{
    System.out.println("Nom : "+super.getNom()+", Prenom : "+super.getPrenom()+",
Specialite : "+matiere);
}
}

class Administrateur extends Enseignant
{
    private String poste;

    public Administrateur(String nom,String prenom, String matiere, String poste)
    {
        super(nom,prenom,matiere);
        this.poste = poste;
    }

    public String getInformationAdditionnel()
    {
        return poste;
    }

    public void afficherInformations()
    {
        super.afficherInformations();
        System.out.println("Poste : "+poste);
    }
}

public class Exo2
{
    public static void main(String[] args)
    {
        Personne[] personne = new Personne[3];
        personne[0] = new Etudiant("Amine","Bouraoui","INFO");
        personne[1] = new Enseignant("Ilyes","Saoudi","Math");
        personne[2] = new Administrateur("Ferhat","Chenoufi","Réseaux","Responsable
de Formation");
        personne[0].afficherInformations();
        personne[1].afficherInformations();
        personne[2].afficherInformations();
    }
}

```