

DÉPARTEMENT MATHÉMATIQUE ET INFORMATIQUE

Filière : Génie du Logiciel et des
Systèmes Informatiques Distribués

Programmation
Orientée Objet
Java



**HÉRITAGE,
REDÉFINITION,
POLYMORPHISME**

RÉALISÉ PAR
Achraf TAFFAH

ENCADRANT
M. Abdelmajid BOUSSELHAM

2022/2023

Table des matières:

Introduction Générale	3
L'énoncé des exercices	4
Exercice 1 :	4
Exercice 2 :	4
Conception et modélisation.....	5
Exercice 1	5
Exercice 2	6
Présentation des programmes finals.....	7
Exercice 1	7
Exercice 2	10
Conclusion	13

Introduction Générale

La programmation orientée objet (POO) est un paradigme informatique consistant à définir et à faire interagir des objets grâce à différentes technologies, notamment les langages de programmation (Python, Java, C++, Ruby, Visual Basic, .NET, Simula...).

L'objectif de ces exercices est d'appliquer les notions de l'héritage, la redéfinition et le polymorphisme avec le langage de programmation orientée objet Java. Bien sûr l'intérêt de ces notions consiste à la réutilisation de code ce qui va nous aider à produire un code ouvert au niveau de développement.

L'énoncé des exercices

Exercice 1 :

On souhaite créer une application JAVA pour la gestion des livres et des adhérents d'une bibliothèque.

1. Créez une classe *Personne* avec les attributs privés : nom, prénom, email, tel, et âge. Ajoutez le constructeur avec paramètres pour initialiser les différents attributs et la méthode **afficher()** pour afficher ces attributs.
2. Créez une deuxième classe *Adhèrent* qui hérite de la classe *Personne* et qui contient l'attribut numAdherent et redéfinit la méthode **afficher()**.
3. Créez une troisième classe *Auteur* qui hérite de la classe *Personne*, qui contient l'attribut numAuteur et redéfinit la méthode **afficher()**.
4. Créez la classe *Livre* qui contient un attribut ISBN (entier) et un *auteur*. Ajoutez également la méthode **afficher()** qui affiche le ISBN, le titre et les informations de l'auteur.
5. Créez une application qui contient une méthode **main()** pour tester les différentes classes, dans laquelle :
 - A. Déclarez et instanciez un adhérent ;
 - B. Déclarez et instanciez un livre qui est écrit par un auteur ;
 - C. Affichez les informations de l'adhérent et du livre.

Exercice 2 :

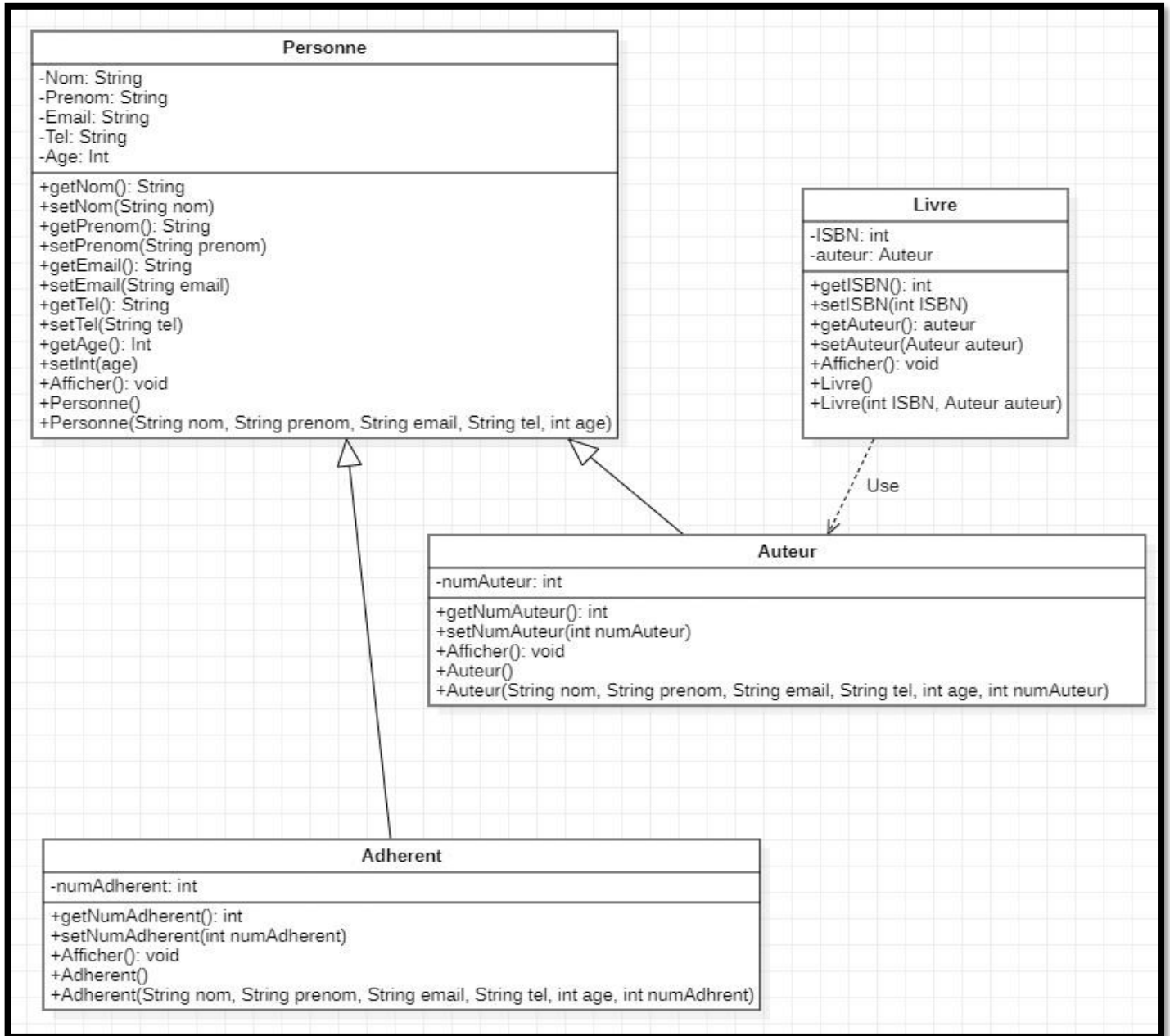
On souhaite créer une application en java qui permet de gérer les salaires des ingénieurs et des managers d'une entreprise de développement informatique.

1. Créez la classe abstraite *Employé* avec les attributs nom, prénom, email, téléphone, et salaire. Ajoutez les constructeurs avec et son paramètres, puis la méthode abstraite **calculerSalaire()** qui retourne le salaire d'un employé.
2. Créez la classe *Ingénieur* avec l'attribut spécialité. Redéfinissez la méthode **calculerSalaire()** sachant qu'on prévoit une augmentation de 15% par rapport à son salaire normal.

Conception et modélisation

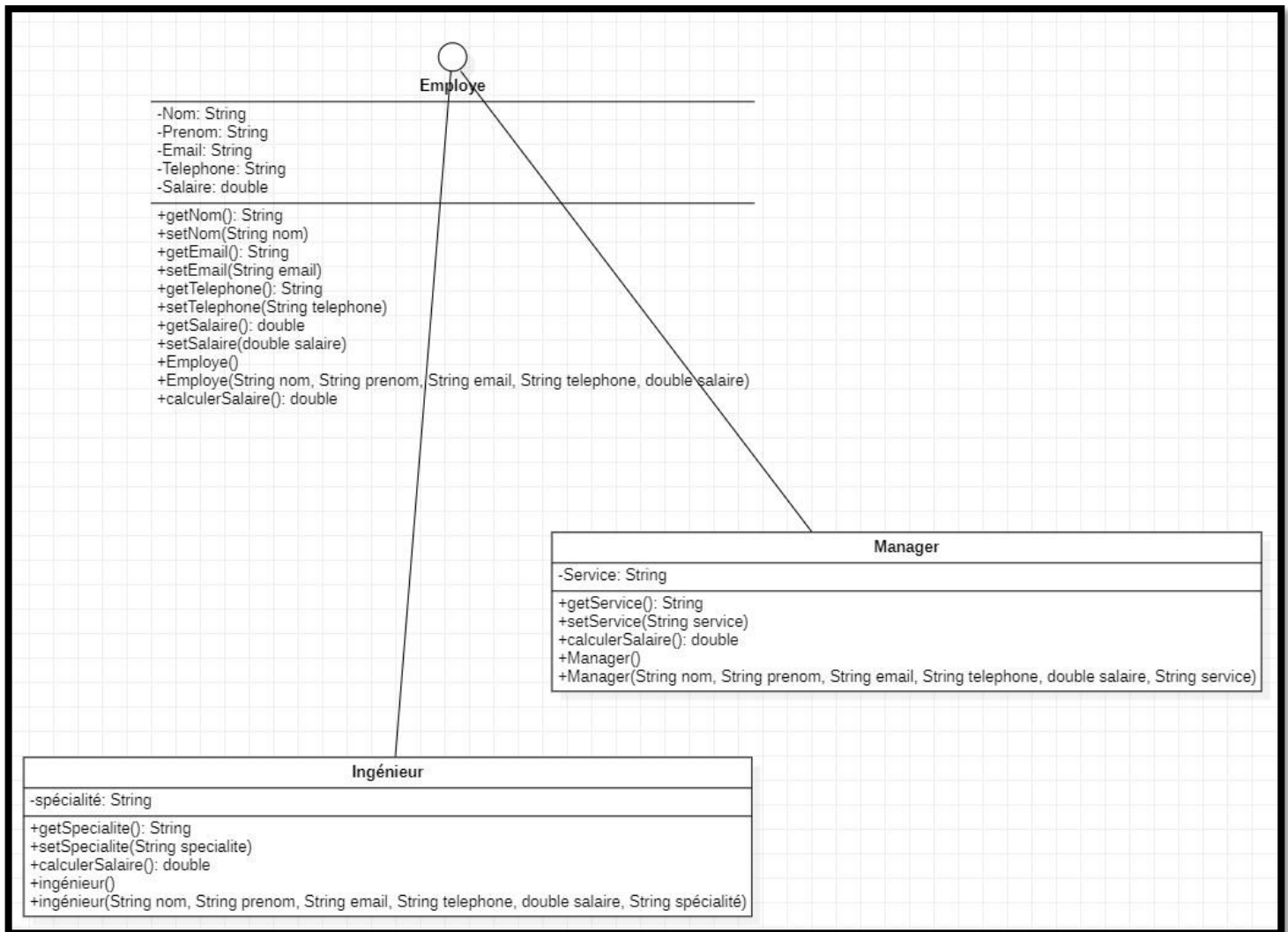
Exercice 1

Ce diagramme de classe représente les classes *Auteur* et *Adhèrent* qui héritent de la classe de base *Personne*, aussi ce dernier représente la classe *Livre* qui utilise la classe *Auteur*.



Exercice 2

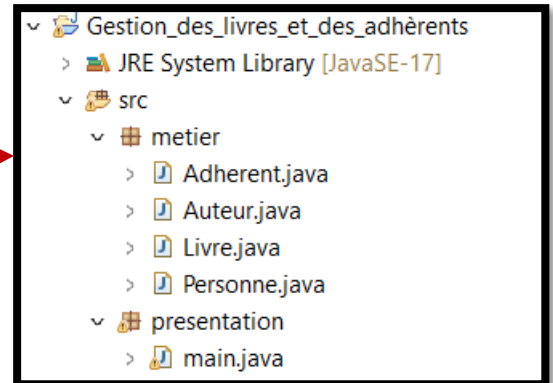
Ce diagramme de classe représente les classes Manager et Ingénieur qui implémentés l'interface Employé.



Présentation des programmes finals

Exercice 1

J'ai créé le package métier qui contient la classe de bases *Personne* et les classes *Adhérent* et *Auteur* qui héritent la classe de base, puis j'ai créé la classe *Livre* qui utilise la classe *Auteur*, Enfin j'ai créé le package présentation qui contient la classe *main*.



J'ai créé une classe *Personne* avec les attributs privés : nom, prénom, email, tel, et âge. Puis j'ai créé des constructeurs et la méthode **afficher()** pour afficher ces attributs.

```
1 package metier;
2 public class Personne {
3     private String nom,email,tel;
4     private int age;
5     public Personne() {
6         this.nom = ""; this.email = ""; this.tel = ""; this.age = 0;
7     }
8     public Personne(String nom, String email, String tel, int age) {
9         this.nom = nom; this.email = email; this.tel = tel; this.age = age;
10    }
11    public String getNom() {
12        return nom;
13    }
14    public void setNom(String nom) {
15        this.nom = nom;
16    }
17    public String getEmail() {
18        return email;
19    }
20    public void setEmail(String email) {
21        this.email = email;
22    }
23    public String getTel() {
24        return tel;
25    }
26    public void setTel(String tel) {
27        this.tel = tel;
28    }
29    public int getAge() {
30        return age;
31    }
32    public void setAge(int age) {
33        this.age = age;
34    }
35    public void Afficher() {
36        System.out.println("Personne [nom=" + nom + ", email=" + email + ", tel=" + tel + ", age=" + age + "];");
37    }
38 }
```

J'ai créé une deuxième classe *Adherent* qui hérite de la classe *Personne* et qui contient l'attribut *numAdherent*, puis j'ai redéfini la méthode **affichée()**.

```
1 package metier;
2
3 public class Adherent extends Personne {
4     private int numAdherent;
5
6     public Adherent() {
7         super();
8         this.numAdherent = 0;
9     }
10
11     public Adherent(int numAdherent) {
12         super();
13         this.numAdherent = numAdherent;
14     }
15
16     public int getNumAdherent() {
17         return numAdherent;
18     }
19
20     public void setNumAdherent(int numAdherent) {
21         this.numAdherent = numAdherent;
22     }
23
24     public void Afficher() {
25         System.out.println(" numAdherent=" + numAdherent);
26         super.Afficher();
27     }
28 }
29
```

J'ai créé une troisième classe *Auteur* qui hérite de la classe *Personne*, qui contient l'attribut *numAuteur* et redéfinit la méthode **afficher()**.

```
1 package metier;
2
3 public class Auteur extends Personne{
4     private int numAuteur;
5
6     public Auteur() {
7         super();
8         this.numAuteur = 0;
9     }
10
11     public Auteur(int numAuteur) {
12         super();
13         this.numAuteur = numAuteur;
14     }
15
16     public int getNumAuteur() {
17         return numAuteur;
18     }
19
20     public void setNumAuteur(int numAuteur) {
21         this.numAuteur = numAuteur;
22     }
23
24     public void Afficher() {
25         System.out.println(" numAuteur=" + numAuteur);
26         super.Afficher();
27     }
28 }
29
```


J'ai créé la classe *Livre* qui contient un attribut ISBN (entier) et un auteur. J'ai ajouté la méthode *afficher()* qui affiche le ISBN, le titre et les informations de l'auteur.

```
1 package me.Cloze ;
2
3 public class Livre {
4     private int ISBN;
5     private Auteur auteur;
6
7     public Livre() {
8         ISBN = 0;
9         this.auteur = new Auteur();
10    }
11    public Livre(int isbn, Auteur auteur) {
12        ISBN = isbn;
13        this.auteur = auteur;
14    }
15
16    public int getISBN() {
17        return ISBN;
18    }
19
20    public void setISBN(int isbn) {
21        ISBN = isbn;
22    }
23
24    public Auteur getAuteur() {
25        return auteur;
26    }
27
28    public void setAuteur(Auteur auteur) {
29        this.auteur = auteur;
30    }
31
32    public void affiche() {
33        System.out.println(" ISBN=" + ISBN);
34        auteur.Afficher();
35    }
36 }
37
```

J'ai créé une application qui contient une méthode *main()* pour tester les différentes classes.

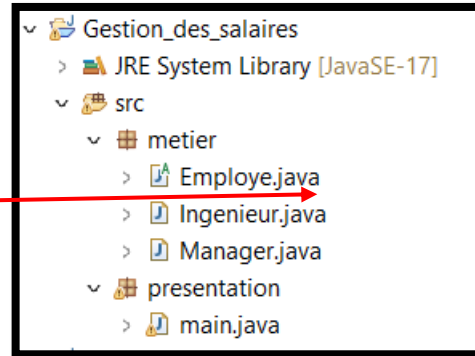
```
1 package presentation;
2
3 import metier.Adherent;
4 import metier.Auteur;
5 import metier.Livre;
6
7 public class main {
8     public static void main(String args[]) {
9         Auteur auteur = new Auteur("Achraf taffah", "taffahachraf184@gmail.com", "0684134782", 22, 78946521);
10        Adherent adherent = new Adherent("Ahmed erraoui", "ahmed@gmail.com", "05114477", 21, 87542165);
11        Livre livre = new Livre(854, auteur);
12
13        auteur.Afficher();
14        System.out.println("");
15        adherent.Afficher();
16        System.out.println("");
17        livre.affiche();
18    }
19 }
20
```

```
Problems  Javadoc  Declaration  Console x
<terminated> Main [Java Application] C:\Users\R1304\Downloads\eclipse-java-2022-09-R-win32-x86_64\eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.4.v20220903-1038\jre\b
[ nom=Achraf taffah, email=taffahachraf184@gmail.com, tel=0684134782, age=22 numAuteur=78946521 --> Auteur]
[ nom=Ahmed erraoui, email=ahmed@gmail.com, tel=05114477, age=21 numAdherent=87542165 --> Adherent]
[[ nom=Achraf taffah, email=taffahachraf184@gmail.com, tel=0684134782, age=22 numAuteur=78946521 --> Auteur] ISBN=854] -->Livre
```

Le résultat après la compilation et l'exécution du programme principal.

Exercice 2

J'ai créé le package métier qui contient l'interface « Employé » et les classes *Ingenieur* et *Manager* qui implémentent l'interface ladite, puis j'ai créé le package présentation qui contient la classe *main*.



J'ai créé la classe abstraite *Employe* avec les attributs *nom*, *prenom*, *email*, *telephone*, et *salaire*. J'ai ajouté les constructeurs avec et sans paramètres, puis la méthode abstraite *calculerSalaire()* qui retourne le salaire d'un employé.

```
1 package metier;
2 abstract public class Employe {
3     private String nom, prenom, email, telephone; private double salaire;
4     public Employe() {
5         this.nom = ""; this.prenom = ""; this.email = ""; this.telephone = ""; this.salaire = 0;
6     }
7     public Employe(String nom, String prenom, String email, String telephone, double salaire) {
8         this.nom = nom; this.prenom = prenom; this.email = email; this.telephone = telephone; this.salaire = salaire;
9     }
10    public String getNom() {
11        return nom;
12    }
13    public void setNom(String nom) {
14        this.nom = nom;
15    }
16    public String getPrenom() {
17        return prenom;
18    }
19    public void setPrenom(String prenom) {
20        this.prenom = prenom;
21    }
22    public String getEmail() {
23        return email;
24    }
25    public void setEmail(String email) {
26        this.email = email;
27    }
28    public String getTelephone() {
29        return telephone;
30    }
31    public void setTelephone(String telephone) {
32        this.telephone = telephone;
33    }
34    public double getSalaire() {
35        return salaire;
36    }
37    public void setSalaire(double salaire) {
38        this.salaire = salaire;
39    }
40    public abstract double calculerSalaire();
41    public abstract void afficherEmploye();
42 }
43
```

J'ai créé la classe Ingénieur avec l'attribut spécialité. Puis j'ai redéfinis la méthode calculerSalaire() sachant qu'on prévoit une augmentation de 15% par rapport à son salaire normal.

```
1 package metier;
2
3 public class Ingénieur extends Employe{
4     private String specialite;
5
6     public Ingénieur() {
7         super();
8         this.specialite = "";
9     }
10
11     public Ingénieur(String nom, String prenom, String email, String telephone, double salaire, String specialite) {
12         super(nom, prenom, email, telephone, salaire);
13         this.specialite = specialite;
14     }
15
16     public String getSpecialite() {
17         return specialite;
18     }
19
20     public void setSpecialite(String specialite) {
21         this.specialite = specialite;
22     }
23
24     public double calculerSalaire() {
25         return getSalaire()*1.15;
26     }
27 }
28
```

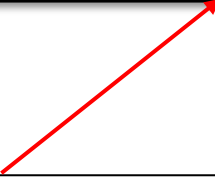
J'ai créé la classe Manager avec l'attribut service. Puis j'ai redéfinis la méthode calculerSalaire() sachant qu'on prévoit une augmentation de 20% par rapport à son salaire normal.

```
1 package metier;
2
3 public class Manager extends Employe{
4
5     private String service;
6
7     public Manager() {
8         super();
9         this.service = "";
10    }
11
12    public Manager(String nom, String prenom, String email, String telephone, double salaire, String service) {
13        super(nom, prenom, email, telephone, salaire);
14        this.service = service;
15    }
16
17    public String getService() {
18        return service;
19    }
20
21    public void setService(String service) {
22        this.service = service;
23    }
24
25    public double calculerSalaire() {
26        return getSalaire()*1.2;
27    }
28
29 }
30
```

```
1 package presentation;
2
3 import metier.Ingénieur;
4
5
6 public class main {
7     public static void main(String args[]) {
8         Ingénieur ingénieur = new Ingénieur("Achraf", "TAFFAH", "taffahachraf184@gmail.com", "0684134782", 12000.0, "Informatique");
9         Manager manager = new Manager("Anouar", "Bouchlouch", "anouar@gmail.com", "0545778899", 25000.00, "Marketing digital");
10
11         ingénieur.afficherEmploye();
12         manager.afficherEmploye();
13     }
14 }
15
```

J'ai créé une application qui contient une méthode main() pour tester les différentes classes.

```
nom : Achraf prenom : TAFFAH email : taffahachraf184@gmail.com telephone : 0684134782 salaire :13799.999999999998  
nom : Anouar prenom : Bouchlouch email : anouar@gmail.com telephone : 0545778899 salaire :30000.0
```



Le résultat après la compilation
et l'exécution du programme
principal.

Conclusion

Grâce à ces exercices j'ai appliqué tout ce que j'ai étudié ont ce qui concerne l'abstraction, l'héritage et le polymorphisme avec le langage de programmation Java, j'ai bien compris l'intérêt de l'interface, ce dernier vient pour accompli le trou de l'absence de la notion de l'héritage multiple, ce dernier a beaucoup d'inconvénients.