

①

```

public class Employe {
    private int idEmp;
    private String nom;
    private double salaire;
    public Employe (int idEmp, String nom, double salaire)
    {
        this.idEmp = idEmp;
        this.nom = nom;
        this.salaire = salaire;
    }
    public double getSalaire () {
        return salaire;
    }
    public String toString ()
    {
        String s = "Id:" + idEmp + ", " + "Nom:" + nom +
            ", " + "Salaire:" + salaire;
        return s;
    }
}

```

②

```

public class Infirmier extends Employe {
    private int nbHeures;
    private double coutHeure;
    public Infirmier (int id, String nom, double salaire, int nbH,
        double coutH)
    {
        super (id, nom, salaire);
        this.nbHeures = nbH;
        this.coutHeure = coutH;
    }
}

```

```

public double getSalaire ()
{
    salInf = nbHeures * coutHeure;
    return salInf;
}
public String toString ()

```

```

public class Medecin extends Employe {
    private Fonction f;
    public Medecin (int id, String nom, double salaire, Fonction f)
    {
        super (id, nom, salaire);
        this this.f = f;
    }

```

```

! public double getSalaire ()
{
    super.getSalaire
    salmed = salaire + f.getPrime();
    return salmed;
}

```

(2)

```

public class Fonction {
    private int idf;
    private String nom;
    private double prime;

```

The way to succeed

```

    public Fonction (int id, String nom, double prime) {
        this.idf = id; this.nom = nom; this.prime = prime;
    }

```

```

public double getPrime() { return prime; }
public String getNom() { return nom; }
public void setNom(String n) { this.nom = n; }

```

```

public class Test {
    public static void main(String args[]) {
        Infirmier inf1 = new Infirmier(1, "Ali", 1000);
        Infirmier inf2 = new Infirmier(2, "Salah", 1150);
        private List<Infirmier> listInf = new ArrayList<Infirmier>();

```

~~(public void)~~

```
listInf.add(inf1);
```

```
listInf.add(inf2);
```

```

public void afficherEtat() {
    for (int i = 0; i < listInf.size(); i++)
        System.out.println(listInf.get(i).toString());
}

```

3

```

Medecin med1 = new Medecin(10, "Mohamed", 4000, f.setNom("Assistant"));

```

```

Medecin med2 = new Medecin(11, "AbdALLAH", 4300, f.setNom("chef"));

```

```

private List<Medecin> listMed = new ArrayList<Medecin>();

```

```
listMed.add(med1);
```

```
listMed.add(med2);
```

il faut rajouter toString dans la classe Infirmier

```

for( Infirmier i: listInf ) {
    system.out.println(i);
}

```

(4)

Soit la classe Entreprise et la classe Personne structurées avec les prop. nom et salaire. Ces deux var. doivent être déclarés private. Donnez la classe Entreprise d'une méthode qui permet d'augmenter le salaire d'une personne, selon un coeff lorsque son salaire est < 1000 .

```

public class Personne {
    private String nom; private Entreprise e;
    private double salaire;
    public Personne(String n, double sal) {
        this.nom = n;
        this.salaire = sal; }
    public double getSalaire() {
        return salaire; }
    public void setSalaire(double sal) {
        this.salaire = sal; }
}

```

```

public class Entreprise {
    private double coeff;
    private List<Personne> l = new
    ArrayList<Personne>();
    public void augmenterSal(double coeff)
    {
        for (Personne p: l)
        {
            if (p.getSalaire() < 1000)
            {
                p.setSalaire(p.getSalaire() * coeff);
            }
        }
        return 0;
    }
}

```