

# Salaire

Le directeur d'une entreprise de produits chimiques souhaite gérer les salaires et primes de ses employés au moyen d'un programme Java. Un employé est caractérisé par son nom, son prénom, son âge et sa date d'entrée en service dans l'entreprise.

Codez une classe abstraite `Employe` dotée des attributs nécessaires, d'une méthode abstraite `calculerSalaire` (ce calcul dépendra en effet du type de l'employé) et d'une méthode `getNom` retournant une chaîne de caractère obtenue en concaténant la chaîne de caractères "L'employé " avec le prénom et le nom.

Dotez également votre classe d'un constructeur prenant en paramètre l'ensemble des attributs nécessaires.

Calcul du salaire Le calcul du salaire mensuel dépend du type de l'employé. On distingue les types d'employés suivants :

- Ceux affectés à la Vente. Leur salaire mensuel est le 20 % du chiffre d'affaire qu'ils réalisent mensuellement, plus 400 €.
- Ceux affectés à la Représentation. Leur salaire mensuel est également le 20 % du chiffre d'affaire qu'ils réalisent mensuellement, plus 800 €.
- Ceux affectés à la Production. Leur salaire vaut le nombre d'unités produites mensuellement multipliées par 5.
- Ceux affectés à la Manutention. Leur salaire vaut leur nombre d'heures de travail mensuel multipliées par 65 €.

Faire une classe `Entreprise` permettant:

- `void ajouterEmploye(Employe)` qui ajoute un employé à la collection.
- `void calculerSalaires()` qui affiche le salaire de chacun des employés de la collection.
- `double salaireMoyen()` qui affiche le salaire moyen des employés de la collection.

## Correction

Une parmi tant d'autre

```
/* *****  
 * La classe Employe  
 * *****/  
abstract class Employe {
```

```

private String nom;
private String prenom;
private int age;

private String date;

public Employe(String prenom, String nom, int age, String date) {

    this.nom = nom;
    this.prenom = prenom;

    this.age = age;
    this.date = date;

}

public abstract double calculerSalaire();

public String getTitre()
{
    return "L'employé " ;
}

public String getNom() {
    return getTitre() + prenom + " " + nom;
}
}

/* *****
* La classe Commercial (regroupe Vendeur et Représentant)
* *****/
abstract class Commercial extends Employe {

private double chiffreAffaire;

public Commercial(String prenom, String nom, int age, String date,

```

```

        double chiffreAffaire) {
super(prenom, nom, age, date);
this.chiffreAffaire = chiffreAffaire;

}

public double getChiffreAffaire()
{
    return chiffreAffaire;
}

}

/* *****
* La classe Vendeur
* *****/
class Vendeur extends Commercial {

private final static double POURCENT_VENDEUR = 0.2;
private final static int BONUS_VENDEUR = 400;

public Vendeur(String prenom, String nom, int age, String date,

        double chiffreAffaire) {
super(prenom, nom, age, date, chiffreAffaire);
}

public double calculerSalaire() {
return (POURCENT_VENDEUR * getChiffreAffaire()) + BONUS_VENDEUR;
}

public String getTitre()
{
    return "Le vendeur ";
}
}

```

```

}

/* *****
* La classe Représentant
* *****/
class Représentant extends Commercial {

    private final static double POURCENT_REPRESENTANT = 0.2;
    private final static int BONUS_REPRESENTANT = 800;

    public Représentant(String prenom, String nom, int age, String date, double
chiffreAffaire) {

        super(prenom, nom, age, date, chiffreAffaire);
    }

    public double calculerSalaire() {

        return (POURCENT_REPRESENTANT * getChiffreAffaire()) + BONUS_REPRESENTANT;

    }

    public String getTitre()
    {
        return "Le représentant ";
    }
}

/* *****
* La classe Technicien (Production)
* *****/
class Technicien extends Employe {

    private final static double FACTEUR_UNITE = 5.0;
    private int unites;

    public Technicien(String prenom, String nom, int age, String date, int unites) {

```

```

        super(prenom, nom, age, date);
        this.unites = unites;

    }

    public double calculerSalaire() {
        return FACTEUR_UNITE * unites;
    }

    public String getTitre()
    {
        return "Le technicien ";
    }
}

/* *****
 * La classe Manutentionnaire
 * *****/
class Manutentionnaire extends Employe {

    private final static double SALAIRE_HORAIRE = 65.0;
    private int heures;

    public Manutentionnaire(String prenom, String nom, int age, String date,
        int heures) {
        super(prenom, nom, age, date);
        this.heures = heures;
    }

    public double calculerSalaire() {
        return SALAIRE_HORAIRE * heures;
    }

    public String getTitre()
    {

```

```

        return "Le manut. " ;

    }
}

/* *****
 * L'interface d'employés à risque
 * *****/
interface ARisque {

    int PRIME = 200;
}

/* *****
 * Une première sous-classe d'employé à risque
 * *****/

class TechnARisque extends Technicien implements ARisque {

    public TechnARisque(String prenom, String nom, int age, String date, int unites) {

        super(prenom, nom, age, date, unites);
    }

    public double calculerSalaire() {

        return super.calculerSalaire() + PRIME;
    }

}

/* *****
 * Une autre sous-classe d'employé à risque
 * *****/
class ManutARisque extends Manutentionnaire implements ARisque {

    public ManutARisque(String prenom, String nom, int age, String date, int heures) {

        super(prenom, nom, age, date, heures);
    }
}

```

```

public double calculerSalaire() {

    return super.calculerSalaire() + PRIME;
}

}
/* *****
* La classe Personnel
* *****/
class Personnel {
    private Employe[] staff;

    private int nbreEmploye;
    private final static int MAXEMPLOYE = 200;

    public Personnel() {
        staff = new Employe[ MAXEMPLOYE];

        nbreEmploye = 0;
    }

    public void ajouterEmploye(Employe e) {

        ++nbreEmploye;
        if (nbreEmploye <= MAXEMPLOYE) {

            staff[nbreEmploye - 1] = e;
        } else {

            System.out.println("Pas plus de " + MAXEMPLOYE + " employés");

        }
    }

    public double salaireMoyen() {

        double somme = 0.0;
        for (int i = 0; i < nbreEmploye; i++) {

```

```

        somme += staff[i].calculerSalaire();
    }

    return somme / nbreEmploye;
}

public void afficherSalaires() {

    for (int i = 0; i < nbreEmploye; i++) {

        System.out.println(staff[i].getNom() + " gagne "

            + staff[i].calculerSalaire() + " francs.");

    }
}

}

class Salaires {

    public static void main(String[] args) {

        Personnel p = new Personnel();
        p.ajouterEmploye(new Vendeur("Pierre", "Business", 45, "1995", 30000));

        p.ajouterEmploye(new Representant("Léon", "Vendtout", 25, "2001", 20000));

        p.ajouterEmploye(new Technicien("Yves", "Bosseur", 28, "1998", 1000));

        p.ajouterEmploye(new Manutentionnaire("Jeanne", "Stocketout", 32, "1998", 45));

        p.ajouterEmploye(new TechnARisque("Jean", "Flippe", 28, "2000", 1000));

        p.ajouterEmploye(new ManutARisque("Al", "Abordage", 30, "2001", 45));

        p.afficherSalaires();
        System.out.println("Le salaire moyen dans l'entreprise est de "

            + p.salaireMoyen() + " francs.");
    }
}

```

}

}

---

Revision #4

Created 19 November 2019 11:08:35 by Admin

Updated 20 November 2019 07:11:40 by Admin