

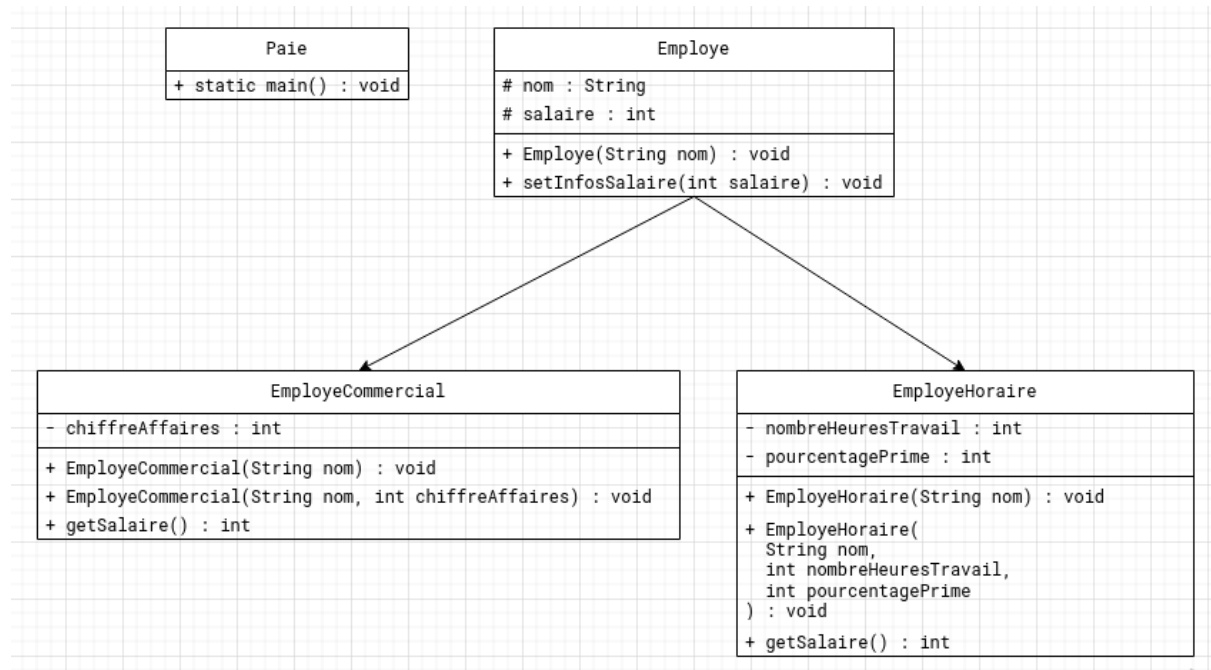
Sommaire

Définition.....	3
Architecture UML.....	4
Documentation.....	5
Classe "Employe".....	5
Attributs.....	5
Méthodes.....	5
Classe "EmployeCommercial".....	6
Attributs.....	6
Méthodes.....	6
Classe "EmployeHoraire".....	7
Attributs.....	7
Méthodes.....	7
Classe "Paie".....	8
Méthodes.....	8
Code source.....	9
Résultat du programme.....	12

Définition

En Java, l'héritage permet de lier les classes enfants aux classes parents. L'avantage avec ce système, c'est qu'on peut utiliser les attributs et les méthodes parents à partir des classes enfants, ce qui permet d'éviter les répétitions au niveau du code.

Architecture UML



Les classes "**EmployeCommercial**" et "**EmployeHoraire**" héritent de la classe "**Employe**". La classe "**Paie**" sert de point d'entrée pour le programme; elle affiche le salaire de chaque employé, quel que soit leur type.

"-" signifie "private".

"#" signifie "protected".

"+" signifie "public".

Documentation

Classe "Employe"

Attributs

Nom	Description
String nom	Nom de l'employé.
int salaire	Salaire de l'employé. Par défaut, il est défini sur 1540, qui est le SMIC actuel.

Méthodes

Nom	Description
void Employe(String nom)	Constructeur avec un seul paramètre : - String nom : Nom de l'employé
void setInfosSalaire(int salaire)	Définir le salaire de l'employé.

Classe "EmployeCommercial"

Attributs

Nom	Description
int chiffreAffaires	Le chiffre d'affaires qui sera utilisé pour faire des calculs.
EmployeCommercial(String nom)	Un constructeur avec un seul paramètre : <ul style="list-style-type: none">- String nom : Nom de l'employé.

Méthodes

Nom	Description
void EmployeCommercial(String nom)	Constructeur avec un seul paramètre : <ul style="list-style-type: none">- String nom : Nom de l'employé
void EmployeCommercial(String nom, int chiffreAffaires)	Un constructeur avec un 2 paramètres : <ul style="list-style-type: none">- String nom : Nom de l'employé.- int chiffreAffaires: Chiffre d'affaires.
int getSalaire()	Retourne le salaire de l'employé commercial. (*)

** L'employé commercial est payé avec une somme fixe à laquelle 1% du chiffre d'affaires va être additionné dans la semaine.*

Classe "EmployeHoraire"

Attributs

Nom	Description
int chiffreAffaires	Le chiffre d'affaires qui sera utilisé pour faire des calculs.

Méthodes

Nom	Description
void EmployeHoraire(String nom)	Constructeur avec un seul paramètre : <ul style="list-style-type: none">- String nom : Nom de l'employé
void EmployeHoraire(String nom, int nombreHeuresTravail, int pourcentagePrime)	Un constructeur avec un seul paramètre : <ul style="list-style-type: none">- String nom : Nom de l'employé.- int nombreHeuresTravail : Nombre d'heures de travail par jour.- int pourcentagePrime : Pourcentage du prime.
int getSalaire()	Retourne le salaire de l'employé horaire. (**)

**** :** L'employé horaire qui est payé suivant le nombre d'heures qu'il a travaillé dans la semaine. Il est payé à un certain tarif horaire et son heure supplémentaire (> 35 heures) est payée en fonction du pourcentage du prime qui a été donné en paramètre.

Classe "Paie"

Méthodes

<i>Nom</i>	<i>Description</i>
static void main(String[] args)	Point d'entrée pour le programme.

Code source

```
public class EmployeCommercial extends Employe {
    private int chiffreAffaires;

    /**
     * @param nom Nom de l'employé
     */
    public EmployeCommercial(String nom) {
        super(nom);
    }

    /**
     * @param nom Nom de l'employé
     * @param nombreHeuresTravail Nombre d'heures de travail par jour.
     * @param pourcentagePrime Pourcentage du prime.
     * @param chiffreAffaires Chiffre d'affaires.
     */
    public EmployeCommercial(String nom, int chiffreAffaires) {
        super(nom);

        this.chiffreAffaires = chiffreAffaires;
    }

    /**
     * Retourne le salaire de l'employé commercial.
     * INFO: L'employé commercial est payé avec une somme fixe à laquelle 1% du chiffre
     * d'affaires va être additionné dans la semaine.
     */
    public int getSalaire() {
        return super.getSalaire() + (1 / 100) * this.chiffreAffaires;
    }

    @Override
    public String toString() {
        return String.format("%s gagne %d €", this.nom, this.getSalaire());
    }
}
```



```

6 public class EmployeHoraire extends Employe {
7     private int nombreHeuresTravail = 35;
8     private int pourcentagePrime = 30;
9
10    /**
11     * @param nom Nom de l'employé
12     */
13    public EmployeHoraire(String nom) {
14        super(nom);
15    }
16
17    /**
18     * @param nom Nom de l'employé
19     * @param nombreHeuresTravail Nombre d'heures de travail par jour.
20     * @param pourcentagePrime Pourcentage du prime.
21     * @param chiffreAffaires Chiffre d'affaires.
22     */
23    public EmployeHoraire(String nom, int nombreHeuresTravail, int pourcentagePrime) {
24        super(nom);
25
26        this.nombreHeuresTravail = nombreHeuresTravail;
27        this.pourcentagePrime = pourcentagePrime;
28    }
29
30    /**
31     * Retourne le salaire de l'employé horaire.
32     *
33     * INFO: L'employé horaire qui est payé suivant le nombre d'heures qu'il a travaillé
34     * dans la semaine. Il est payé à un certain tarif horaire et son heure supplémentaire (> 35 heures)
35     * sont payés en fonction du pourcentage du prime qui a été donné en paramètre.
36     */
37    public int getSalaire() {
38        if (this.nombreHeuresTravail > 35) {
39            int prime = super.salaire * (this.pourcentagePrime / 100);
40
41            return super.salaire + prime;
42        } else {
43            return super.salaire;
44        }
45    }
46
47    @Override
48    public String toString() {
49        return String.format("%s gagne %d €", this.nom, this.getSalaire());
50    }
51 }

```

I

```

6 public class Employe {
7     protected String nom;
8     protected int salaire = 1540; // SMIC
9
10    /**
11     * @param nom Nom de l'employé
12     */
13    public Employe(String nom) {
14        this.nom = nom;
15    }
16
17    /**
18     * Définir le salaire de l'employé.
19     */
20    public void setInfosSalaire(int salaire) {
21        this.salaire = salaire;
22    }
23
24    @Override
25    public String toString() {
26        return String.format("%s gagne %d €", this.nom, this.salaire);
27    }
28 }

```

```

1 package java_tp2;
2
3 public class Paie {
4     /**
5      * Point d'entrée du programme.
6      */
7     public static void main(String[] args) {
8         int CHIFFRE_AFFAIRES = 250_000;
9
10        EmployeCommercial ec1 = new EmployeCommercial("Durand");
11        EmployeCommercial ec2 = new EmployeCommercial("Samuel", CHIFFRE_AFFAIRES);
12        EmployeCommercial ec3 = new EmployeCommercial("Aziz", CHIFFRE_AFFAIRES);
13
14        // ec1.setInfosSalaire(1540); // (SMIC)
15        ec2.setInfosSalaire(2200);
16        ec3.setInfosSalaire(1900);
17
18        EmployeHoraire eh1 = new EmployeHoraire("Victor");
19        EmployeHoraire eh2 = new EmployeHoraire("Jade", CHIFFRE_AFFAIRES, 38, 30);
20        EmployeHoraire eh3 = new EmployeHoraire("Hakim", CHIFFRE_AFFAIRES, 39, 50);
21
22        // eh1.setInfosSalaire(1540); // (SMIC)
23        eh2.setInfosSalaire(2100);
24        eh3.setInfosSalaire(1700);
25
26        System.out.println(ec1); // 1540 €
27        System.out.println(ec2); // 2240 €
28        System.out.println(ec3); // 1900 €
29
30        System.out.println(eh1); // 1540 €
31        System.out.println(eh2); // 2100 €
32        System.out.println(eh3); // 1700 €
33    }
34 }

```

Résultat du programme

Durand gagne 1540 €

Samuel gagne 2200 €

Aziz gagne 1900 €

Victor gagne 1540 €

Jade gagne 2100 €

Hakim gagne 1700 €