

Compte rendu TD 2 SLAM4

Pour rappel, ce TD peut comporter des éléments au début que nous n'aurons vu que plus tard dans le TD.

Dans un premier temps, nous allons modéliser cette situation à l'aide des classes sous forme de package comme ci-dessous :

```
Employee.java  Paie.java  Entreprise.java
1 package fr.biraveen.employe;
```

Ensuite, nous allons créer une classe Employé comme ci-dessous :

```
Employee.java  Paie.java  Entreprise.java  Entrepr
1 package fr.biraveen.employe;
2
3 public class Employe extends Entreprise{
4
5     protected String nomEmp;
6     protected int infosalaire;
7     protected int nbrHeures;
8     protected double CA;
9     protected String nom;
10
```

Une fois que ceci a été fait, nous allons créer un modificateur qui se nommera "setInfosSalaire" comme ci-dessous :

```
public void setInfosalaire(int infosalaire) {
    this.infosalaire=infosalaire;
}
```

Par la suite, la classe doit obligatoirement avoir un constructeur comme ci-dessous :

```
public Employe(String nomEmp, int infosalaire, int nbrHeures, double CA,String nom,int nbrEmploye, int capital,String mission) {
    super(nom, nbrEmploye, capital, mission);
    this.nomEmp= nomEmp;
    this.infosalaire = infosalaire;
    this.CA=CA;
    this.nbrHeures=nbrHeures;
}
```

Ci-dessous il ya un mini programme que nous avons fait pour le calcul du salaire dans la méthode "getInfoSalaire" :

```
public int getinfoSalaire() {
    int a =20;
    this.nbrHeures =7;
    infosalaire=(a*nbrHeures)*30;
    System.out.println( "L'employé " + nomEmp + " gagne " + infosalaire + " €.");
    return infosalaire;
}
```

Résultat :

```
public static void main(String[] args) {  
    // TODO Auto-generated method stub  
    Employe employe1 = new Employe ("Jules",0,7,3000, null,0,0, null);// 1/ le nom, 2/ infosala:  
    //employe1.setInfosalaire(1500);  
    employe1.setNom("APEC");  
    StringBuilder employe3 = new StringBuilder("Jules âgé de 21 ans vit en région parisienne");  
  
    System.out.println(employe1.getinfoSalaire());  
}
```

Console ✕

```
<terminated> Paie [Java Application] C:\P  
L'employé Jules gagne 4200 €.  
4200
```

Une fois que ceci a été fait, nous allons créer une nouvelle classe qui se nommera **"Paie"** comme ci-dessous :

Paie.java ✕

Cette nouvelle classe contiendra un main comme ci-dessous :

```
Employee.java Paie.java Entreprise.java EntrepriseSecrete.java EntrepriseSansProfit.java  
6  
7 public class Paie {  
8  
9     public static void main(String[] args) {  
10        // TODO Auto-generated method stub  
11        Employe employe1 = new Employe ("Jules",0,7,3000, null,0,0, null);// 1/ le nom, 2/ infosalai  
12        //employe1.setInfosalaire(1500);  
13        employe1.setNom("APEC");  
14        StringBuilder employe3 = new StringBuilder("Jules âgé de 21 ans vit en région parisienne");  
15  
16        System.out.println(employe1.getinfoSalaire());  
}
```

Maintenant le but de cette partie est de faire un tableau qui contient l'ensemble des noms d'employés d'une entreprise (**La création de cette future classe se fera plus tard**)

Tout d'abord, commençons par créer une nouvelle classe qui se nommera **"Entreprise"** comme ci-dessous :

Entreprise.java ✕

Cette nouvelle classe contiendra plusieurs variables comme ci-dessous :

```
Employee.java Paie.java Entreprise.java ✕  
1 package fr.biraveen.employe;  
2  
3 public class Entreprise {  
4     protected String nom;  
5     protected int nbrEmploye;  
6     protected double capital;  
7     protected String mission;  
}
```

Maintenant, nous allons faire un tableau qui comportera tous les noms d'employés comme ci-dessous :

```
List<String> Employee=new ArrayList<String>();  
Employee.add("Jules");  
Employee.add("Spike");  
Employee.add("Lea");  
Employee.add("Robine");
```

En plus de cela nous allons faire "For" pour parcourir tout le tableau comme ci-dessous :

```
for(String i : Employee);  
System.out.println(Employee);
```

Voici le résultat :



Console ✖
<terminated> Paie [Java Application] C:\P
[Jules, Spike, Lea, Robine]

Il faut savoir qu'on doit avoir possibilité d'ajouter ou d'enlever un salarié voici les deux formats :

`Employee.add("Jules");` → Pour **ajouter** un mot, on met le nom de la classe suivi du mot clé "add" et si il s'agit d'une chaîne de caractère mettre le mot entre guillemets. Mais **attention** lors de la création du tableau il faut préciser quel sera le type des termes que nous allons ajouter (Voir au-dessus).

`Employee.remove(0);` → Pareil que celui d'au dessus mais dans ce contexte, on **enlève** un salarié de l'entreprise avec le mot clé "remove" et en indiquant l'indice par exemple :

[Jules, Spike, Lea, Robine]
0 1 2 3

Maintenant, en appliquant cette syntaxe avec l'indice 0:

[Spike, Lea, Robine] On peut voir que le nom "Jules" a été supprimé.

Maintenant, nous allons écrire une méthode "ToString" qui affichera le nom de l'entreprise, le nom de l'employé et aussi nous afficherons la description de de chaque employé avec une nouvelle syntaxe qui se nomme : **StringBuilder**.

```
@Override
public String toString() {
    return "Le salarie " + nomEmp + " gagne un salaire de " + infosalaire + " € et dans l'entreprise " + nom + ".";
}
```

Il s'agit de la méthode **ToString**. Maintenant pour afficher la description on procède de cette manière:

```
employee1.setNom("APEC");
StringBuilder employee3 = new StringBuilder("Jules agé de 21 ans vit en région parisienne");
```

Maintenant, voici la syntaxe pour l'afficher :

```
System.out.println (employee1.toString() + employee3);
```

Le résultat lors de l'exécution :

```
Console
<terminated> Paie [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (9 oct. 2020 à 16:44:21 – 16:44:22)
Le salarie Jules est dans l'entreprise APEC.Jules agé de 21 ans vit en région parisienne
```

Maintenant, affichons quelque employés dans 2 entreprises tout en faisant afficher les noms d'entreprises et leurs employés comme ci-dessous :

```
Employee employee1 = new Employee ("Jules",0,7,3000, null,0,0, null);// 1/ le nom, 2/ infosalaire
//employee1.setInfosalaire(1500);
employee1.setNom("APEC");
StringBuilder employee3= new StringBuilder("Jules agé de 21 ans vit en région parisienne");

//System.out.println(employee1.getNom());

Employee employee2 = new Employee ("Spike",0,5,4000, null, 0, 0, null);// 1/ le nom, 2/ infosalaire
employee2.setInfosalaire(2000);
employee2.setNom("Derichebourg");
StringBuilder employee4 = new StringBuilder("Spike agé de 29 ans et vit en banlieue");

System.out.println (employee1.toString() + employee3);
System.out.println (employee2.toString() + employee4);
```

Le résultat lors de l'exécution:

```
Console
<terminated> Paie [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (9 oct. 2020 à 16:53:26 – 16:53:27)
Le salarie Jules est dans l'entreprise APEC.Jules agé de 21 ans vit en région parisienne
Le salarie Spike est dans l'entreprise Derichebourg.Spike agé de 29 ans et vit en banlieue
```

Maintenant, nous allons une nouvelle classe qui se nommera "**EntrepriseSecrete**" dans laquelle il héritera de la classe "Entreprise" comme ci-dessous:


```

1 package fr.biraveen.employe;
2
3 public class EntrepriseSecrete extends Entreprise {
4
5     public EntrepriseSecrete(String nom,int nbrEmploye, int capital,String mission) {
6         super(nom, nbrEmploye, capital, mission);
7     }
8
9     public String mission() {
10        return mission;
11    }
12 }

```

Dans cette classe nous devons la méthode mission va lancé l'exception **SecretMissionException**. Comme ci-dessous :

```

public String mission() {
    return mission;
}

```

Maintenant, nous allons une nouvelle classe qui se nommera **"EntrepriseSansProfit"** dans laquelle il héritera de la classe **"Entreprise"** comme ci-dessous:

```

1 package fr.biraveen.employe;
2
3 public class EntrepriseSansProfit extends Entreprise {
4
5     public EntrepriseSansProfit(String nom,int nbrEmploye, int capital,String mission) {
6         super(nom, nbrEmploye, capital, mission);
7     }
8
9     public int capital(int capital) {
10        return capital;
11    }
12
13    public String toString() {
14        return "Entreprise Sans Profit";
15    }
16 }

```

Maintenant testons la méthode pour que ce dernier affiche les entreprises avec les missions, et avec les capitaux comme ci-dessous :

```

Entreprise entreprise1=new Entreprise ("APEC",50,100000,null);
entreprise1.setCapital(100000);
StringBuilder entreprise3= new StringBuilder("Apec est spécialisé dans le service de recrutement.");

Entreprise entreprise2=new Entreprise ("Derichebourg",30,0,null);
entreprise2.setCapital(20000);
StringBuilder entreprise4= new StringBuilder("Derichebourg est spécialisé dans le service de nettoyage.");

```

Quand on exécute :

```

System.out.println(entreprise1.toString() + entreprise3);
System.out.println(entreprise2.toString() + entreprise4);

```

```

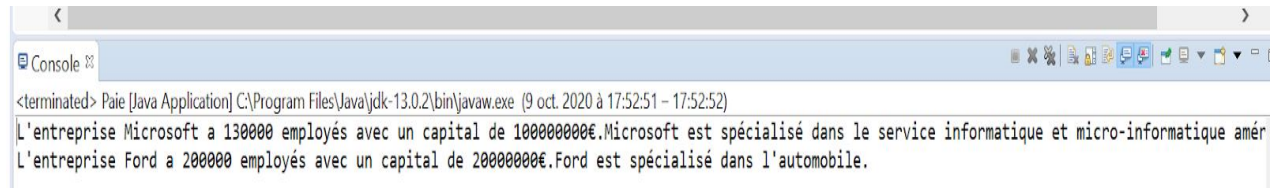
<terminated> Paie [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (9 oct. 2020 à 17:47:49 - 17:47:52)
L'entreprise APEC a 50 employés avec un capital de 100000€.Apec est spécialisé dans le service de recrutement.
L'entreprise Derichebourg a 30 employés avec un capital de 20000€.Derichebourg est spécialisé dans le service de nettoyage.

```

```
Entreprise entreprise5=new Entreprise ("Microsoft",130000,0,null);
entreprise5.setCapital(100000000);
StringBuilder entreprise6= new StringBuilder("Microsoft est spécialisé dans le service informatique et micro-informatique américaine.");

Entreprise entreprise7=new Entreprise ("Ford",200000,0,null);
entreprise7.setCapital(200000000);
StringBuilder entreprise8= new StringBuilder("Ford est spécialisé dans l'automobile.");
```

Résultat:



```
<terminated> Paie [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (9 oct. 2020 à 17:52:51 - 17:52:52)
L'entreprise Microsoft a 130000 employés avec un capital de 100000000€.Microsoft est spécialisé dans le service informatique et micro-informatique amér
L'entreprise Ford a 200000 employés avec un capital de 200000000€.Ford est spécialisé dans l'automobile.
```