

Énoncés 6

Exercices sur l'héritage et les paquetages

1 Exercice sur l'héritage : personnes

On définit comme suit une classe de personnes :

```
public class LaDate {
    private int jour;
    private int mois;
    private int annee;
    public LaDate (int j, int m, int a){
        this.jour=j;
        this.mois = m;
        this.annee = a;
    }
    public int getJour(){
        return jour;
    }
    public int getMois(){
        return mois;
    }
    public int getAnnee(){
        return annee;
    }
    public String toString() {
        return jour+"/"+mois+"/"+annee;
    }
}
class LaPersonne {
    protected LaDate naissance;
    protected String nom;
    public LaPersonne(String name, LaDate naiss) {
        this.nom = name;
        this.naissance = naiss;
    }
    public LaDate getNaissance(){
        return naissance;
    }
    public String getNom(){
        return nom;
    }
}
```

```
    }  
    public String toString() {  
        return nom+"_"+"naissance + ";  
    }  
}
```

Pour certaines personnes, on connaît en plus le lieu de naissance, mais pas pour toutes.
Ecrire une sous-classe des personnes dont on connaît le lieu de naissance. Redéfinir la méthode `toString` pour que le lieu de naissance soit dans la chaîne renvoyée.
Ecrivez un petit programme qui utilise la nouvelle sous-classe.

2 Exercice sur l'héritage : employés

Une entreprise a un certain nombre d'employés. Un employé est connu par son nom, son matricule (qui l'identifie de façon unique) et son indice salarier. Le salaire est calculé en multipliant cet indice par une certaine valeur qui peut changer en cas d'augmentation générale des salaires, mais qui est la même pour tous les employés.

Question 1

Écrivez la classe des employés avec les informations utiles et des méthodes pour obtenir les caractéristiques d'un employé et pour calculer son salaire, ainsi que l'inévitable méthode `toString`.

Question 2

Les commerciaux ont un salaire composé d'un fixe et d'un intéressement proportionnel à leurs ventes. Ecrivez une sous-classe des commerciaux qui contient l'information sur leurs ventes du dernier mois, une méthode pour mettre à jour cette information et redéfinissez la méthode de calcul de leurs salaires.

Question 3

Il s'agit d'écrire une classe appelée `Personnel` représentant le personnel d'une entreprise. Ce personnel comporte plusieurs employés dont certains sont des commerciaux et d'autres pas. Cette classe aura les caractéristiques suivantes :

- tous les employés seront enregistrés dans un unique attribut de type `ArrayList`.
- à la construction de l'objet, il n'y aura aucun employé enregistré.
- la classe contient deux méthodes pour ajouter et retirer un employé.
- la classe contient une méthode qui calcule le montant total des salaires du personnel (la somme des salaires des employés).
- la classe contient une méthode `toString`

1. Pour chaque méthode de la classe, donnez le type des paramètres et du résultat.
2. Écrivez la classe.

Question 4

Écrivez une nouvelle classe avec une méthode `main` qui contiennent les éléments suivants :

- la création de deux objets représentant des employés non commerciaux.
- la création de deux objets représentant des commerciaux.
- la création d'un objet instance de la classe `Personnel` qui contiennent les quatre employés créés.
- le calcul et l'affichage du salaire total du personnel.

3 Exercice sur les paquetages

On donne le code de 4 classes appartenant à deux paquetages différents. Chacune des classes contient 4 méthodes avec chacune un mode de visibilité différent (public, private, protected ou rien). Dans l'une des classes, il y a une méthode main avec un appel à chaque méthode de chaque classe.

Dîtes lesquels de ces appels sont incorrects et pourquoi. Vous pouvez utiliser les numéros de ligne pour les identifier.

```
package packageA;
public class Class1A{
    public void methPublique(){
        System.out.println("methPublique_de_Class1A");
    }
    private void methPrivate(){
        System.out.println("methPrivate_de_Class1A");
    }
    protected void methProtected(){
        System.out.println("methProtected_de_Class1A");
    }
    void methRienDuTout(){
        System.out.println("methRienDuTout_de_Class1A");
    }
}
```

```
package packageA;
public class Class2A{
    public void methPublique(){
        System.out.println("methPublique_de_Class2A");
    }
    private void methPrivate(){
        System.out.println("methPrivate_de_Class2A");
    }
    protected void methProtected(){
        System.out.println("methProtected_de_Class2A");
    }
    void methRienDuTout(){
        System.out.println("methRienDuTout_de_Class2A");
    }
}
```

```
1 package packageB;
2 import packageA.*;
3 public class Class1B extends Class1A{
4     public void methPublique(){
5         System.out.println("methPublique_de_Class1B");
```

```

6     }
7     private void methPrivate(){
8         System.out.println("methPrivate_de_Class1B");
9     }
10    protected void methProtected(){
11        System.out.println("methProtected_de_Class1B");
12    }
13    void methRienDuTout(){
14        System.out.println("methRienDuTout_de_Class1B");
15    }
16    public static void main(String[] args){
17        Class1A objet1A = new Class1A();
18        Class2A objet2A = new Class2A();
19        Class1B objet1B = new Class1B();
20        Class2B objet2B = new Class2B();
21        objet1A.methPublique();
22        objet1A.methPrivate();
23        objet1A.methProtected();
24        objet1A.methRienDuTout();
25        objet2A.methPublique();
26        objet2A.methPrivate();
27        objet2A.methProtected();
28        objet2A.methRienDuTout();
29        objet1B.methPublique();
30        objet1B.methPrivate();
31        objet1B.methProtected();
32        objet1B.methRienDuTout();
33        objet2B.methPublique();
34        objet2B.methPrivate();
35        objet2B.methProtected();
36        objet2B.methRienDuTout();
37    }
38 }

```

```

package packageB;
public class Class2B{
    public void methPublique(){
        System.out.println("methPublique_de_Class2B");
    }
    private void methPrivate(){
        System.out.println("methPrivate_de_Class2B");
    }
    protected void methProtected(){
        System.out.println("methProtected_de_Class2B");
    }
    void methRienDuTout(){
        System.out.println("methRienDuTout_de_Class2B");
    }
}

```
