



Java

Licence professionnelle CISII, 2009-2010

Cours 5 : l'héritage Solutions

Cours inspiré des cours de Richard Grin, Antoine Tabbone et Hazel Everett

Corrigé

■ La classe Animal :

```
class Animal
{
    protected int nbPattes;
    Animal(int nb){//constructeur
        setNbPattes(nb);
    }
    //accesseur
    public int getNbPattes(){return nbPattes;}
    //modifieur
    public void setNbPattes(int nb){nbPattes = nb;}
    public String toString() {
        return "Ceci est un animal doté de " + getNbPattes() + "
pattes";
    }
    public void affiche(){
        System.out.println(toString());
    }
}
```

Corrigé (2)

■ Les classes Autruche et Lapin :

```
class Autruche extends Animal
{
    Autruche(int nb){//constructeur
        super(nb);
    }
    public String toString() {
        return "Ceci est une autruche dotée de " + getNbPattes()
        + " pattes";
    }
}
class Lapin extends Animal
{
    Lapin(int nb){//constructeur
        super(nb);
    }
    public String toString() {
        return "Ceci est un lapin doté de " + getNbPattes() + "
        pattes";
    }
}
```

Corrigé (3)

■ La classe Main :

```
class Main
{
    public static void main(String args[])
    {
        Autruche kiki = new Autruche(2);
        Lapin bunny = new Lapin(4);
        kiki.affiche();
        bunny.affiche();
    }
}
```



Cours5-TD5

■ Exercice 1

- Créez la classe Vehicule qui contient les données
 - `private boolean moteur`, et
 - `private int vitesseMax`, et
- les méthodes suivantes :
 - constructeur,
 - `String toString()`, qui renvoie les caractéristiques du véhicule,
 - `void Vmax()` qui affiche la vitesse maximale du véhicule.
- Créez ensuite une classe Voiture_Composee dont les membres sont
 - `private Vehicule ve`,
 - `private int nombreDePortes`,
 - et `Voiture_Derivee` (dérivant de la classe Vehicule) avec la donnée membre
 - `private int nombreDePortes`.
- Pour les deux classes, définissez leur constructeur et leur méthode `String toString()`. Comparez les deux implantations. Comment accède-t-on aux différentes données membres et aux différentes méthodes des classes Voiture Composee et Voiture Derivee ? Affichez la vitesse maximale de la voiture dans les deux cas.

```

public class Vehicule {
    private boolean moteur;
    private int vitesseMax;

    public Vehicule() {
        moteur = false;
        vitesseMax = 0;
    }

    public Vehicule(boolean m, int v){
        moteur = m;
        vitesseMax = v;
    }

    public String toString(){
        String S="\nvehicule ";
        if (moteur) S=S+"a moteur, ";
        else
            S=S+"sans moteur, ";
        S=S+"dont la vitesse maximale est
de "+vitesseMax+"km/h";
        return S;
    }
}

```

```

public void Vmax(){
    System.out.println("\nvitesse max :
"+vitesseMax+" km/h\n");
}

public boolean getMoteur () {
    return moteur;
}

public int getVitesseMax() {
    return vitesseMax;
}

public static void main(String[] args){
    Vehicule v1=new Vehicule(true,121);
    System.out.println(v1);
    v1.Vmax();
}
}

```

```
public class Voiture_Composee {
    private Vehicule v;
    private int nombreDePortes;

    public Voiture_Composee()
    {
        v = new Vehicule();
        nombreDePortes = 2;
    }

    public Voiture_Composee(boolean m,
        int vit, int n){
        v = new Vehicule(m,vit);
        nombreDePortes = n;
    }

    public String toString(){
        String S;
        S = v.toString();
        S = S+" C'est une voiture de " +
            nombreDePortes + " portes.\n";
        return S;
    }
}
```

```
public static void main(String[] args)
{
    Voiture_Composee v1 = new
        Voiture_Composee(true,152,4);
    System.out.println(v1);
    System.out.println(v1.nombreDePortes);
    System.out.println(v1.v.getVitesseMax());
    System.out.println(v1.v.getMoteur());
    v1.v.Vmax();
}
}
```

```

public class Voiture_Derivee extends
    Vehicule {
    private int nombreDePortes;

    public Voiture_Derivee() {
        super();
        nombreDePortes = 2;
    }

    public Voiture_Derivee(boolean m,int
        v,int n) {
        super(m,v);
        nombreDePortes = n;
    }

    public String toString(){
        String S;
        S=super.toString();
        S=S+" C'est une voiture de " +
            nombreDePortes + " portes.\n";
        return S;
    }
}

```

```

public static void main(String[] args)
{
    Voiture_Derivee v1 = new
        Voiture_Derivee(true,150,2);
    System.out.println(v1);
    System.out.println(v1.getMoteur());
    System.out.println(v1.getVitesseMax());
    System.out.println(v1.nombreDePortes);
    v1.Vmax();
}
}

```



```
public class Main {
    public static void main(String[] args) {

        System.out.println("\nVoiture composee \n=====
        ===== \n");
        Voiture_Composee v1 = new Voiture_Composee(true, 152, 4);
        System.out.println(v1);
        // v1.v.Vmax(); il faut une méthode getVehicule() dans
        Voiture_Composee

        System.out.println("\nVoiture Derivee\n=====
        =====\n");
        Voiture_Derivee v2 = new Voiture_Derivee(true, 150, 2);
        System.out.println(v2);
        v2.Vmax();
    }
}
```



Cours5-TD5

■ Exercice 2

- Créez une classe Liquide contenant seulement une méthode imprimer() qui affiche : "je suis un liquide"
- Créez 2 classes dérivées de la classe Liquide, les classes Cafe et Lait, dont les méthodes imprimer() affichent respectivement "je suis un Café", "je suis du Lait"
- Enfin créez une classe Tasse ayant un attribut l de la classe Liquide et une méthode AjouterLiquide (Liquide li) et une méthode imprimer()
- Testez l'utilisation : créer un tableau de Tasses qui contient différents liquides

```

public class Liquide {
    public void imprimer(){
        System.out.println("je suis un
        liquide");}
}

class Cafe extends Liquide {
    public void imprimer() {
        System.out.println("je suis du
        cafe");}
}

class Lait extends Liquide{
    public void imprimer(){
        System.out.println("je suis du
        lait");}
}

class Tasse {
    private Liquide l;
    public void AjouterLiquide (Liquide l)
        {this.l = l;}
    public void imprimer() {
        l.imprimer();}
}

```

```

public class Main {
    public static void main(String[] args) {

        Tasse latasse;
        latasse =new Tasse();
        latasse.AjouterLiquide(new Liquide());
        latasse.imprimer();

        latasse=new Tasse();
        latasse.AjouterLiquide(new Cafe());
        latasse.imprimer();

        Tasse[] lestasses;
        lestasses = new Tasse[3];
        for (int i=0;i<3;i++)
            lestasses[i]= new Tasse();

        lestasses[0].AjouterLiquide(new Cafe());
        lestasses[1].AjouterLiquide(new Lait());
        lestasses[2].AjouterLiquide(new Liquide());

        for (int i=0;i<3;i++)
            lestasses[i].imprimer();
    }
}

```

je suis un liquide

je suis du cafe

je suis du cafe

je suis du lait

je suis un liquide