



PROGRAMMATION ORIENTÉE OBJET

CHAP 2: OBJET, CLASSE, ABSTRACTION ET ENCAPSULATION

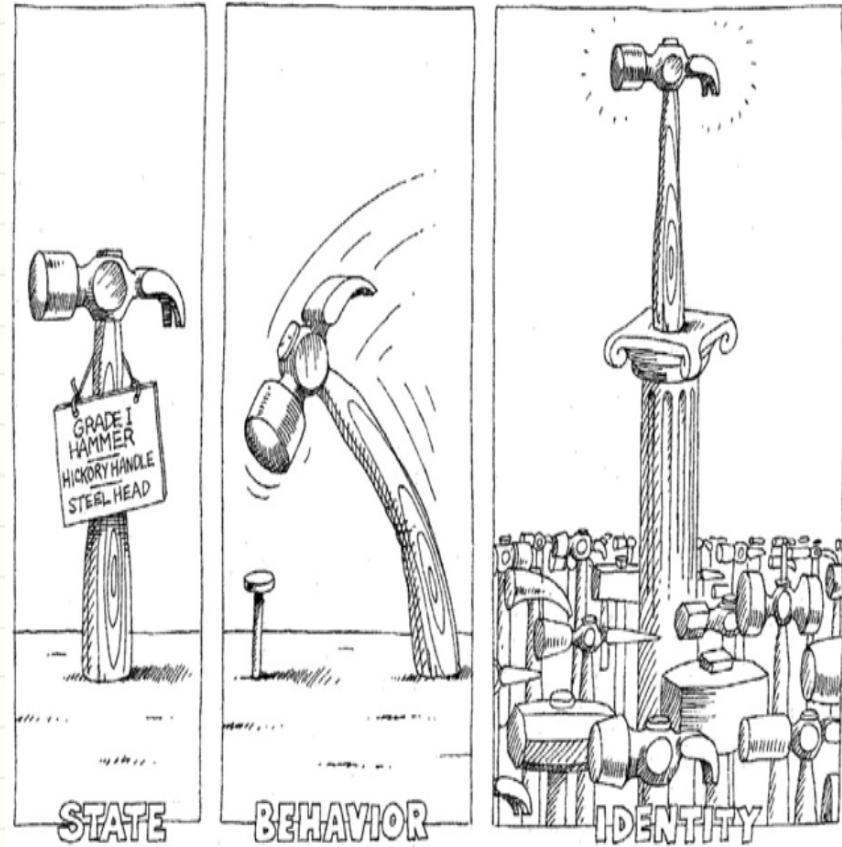
Le principe d'encapsulation

Objet = Attributs (données) + Méthodes (fonctions)

- Le principe d'**encapsulation** consiste à regrouper, dans un même élément informatique, les aspects statiques et dynamiques (les données et les fonctions) spécifiques à une entité.
- Cet élément informatique est appelé : « **objet** »:
 - Les données ou structures de données définies dans un objet sont appelées **les attributs** de l'objet;
 - Les fonctions (de manipulation) définies dans un objet sont appelées **les méthodes** de l'objet.
- L'encapsulation permet de définir deux niveaux de perception :
 - **Le niveau externe** : perception de l'objet depuis l'extérieur. Il constitue l'interface de l'objet formée par ses éléments publics.
 - **Le niveau interne** : perception de l'objet depuis l'intérieur. Ce niveau représente le corps de l'objet formé par ses éléments privés.

Un objet

- **Chaque objet** représente un objet ou un concept de la vie réelle.
 - a **une identité** unique et invariante
 - a **un état** caractérisé par la valeur de ses **attributs**
 - propose des services sous la forme de **méthodes**. Son **comportement** reflète la manière dont il agit ou réagit par changements de son état ou suite à des messages reçus.
 - exécute un service lorsque l'objet reçoit un message



Source: Object-Oriented Analysis and Design with Applications. Grady Booch et al.

Un objet

Un objet = une identité + un état + un comportement



Voiture: Princesse Jaune

Marque: Renault

Modèle: Clio Sport

Couleur: Jaune

Vitesse: 90

Demarrer();

Accelerer();

Ralentir();

Tourner();

Arreter();

Une identité: un nom qui permet de distinguer un objet d'un autre.

Un état: ensemble de valeurs qui caractérisent l'objet à un instant t.

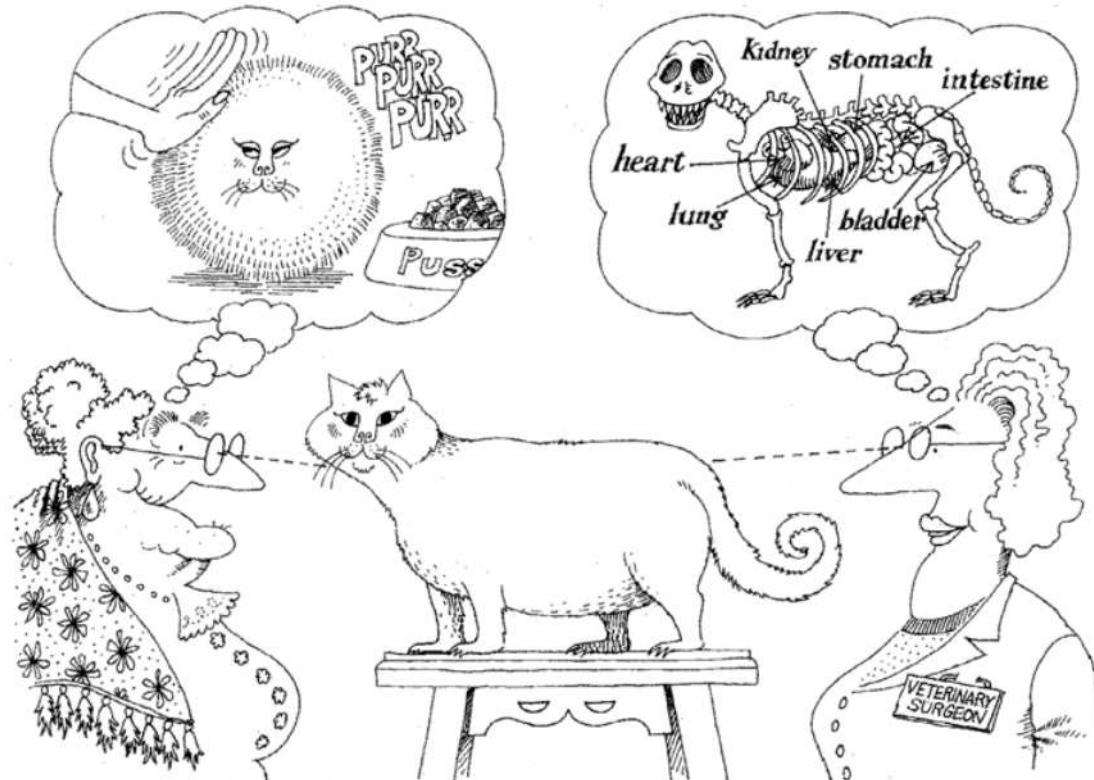
Un comportement: ensemble de services ou d'opérations que peut rendre un objet ou qui modifient son état.



Le principe d'abstraction

- Le monde réel est constitué de très nombreux objets en interaction. Ces objets constituent souvent des entités qui sont trop **complexes** pour être comprises du premier coup dans leur intégralité.
- Pour réduire cette complexité ou du moins pour la maîtriser, l'être humain a appris à **regrouper** les éléments qui se ressemblent et à distinguer des structures de plus haut niveau d'**abstraction**, débarrassées de **détails** inutiles.

Le principe d'abstraction



Abstraction focuses on the essential characteristics of some object, relative to the perspective of the viewer.

Source: *Object-Oriented Analysis and Design with Applications*. Grady Booch et al.

Le principe d'abstraction

- Une abstraction fait ressortir les caractéristiques d'une structure qui la distinguent de tous les autres types de structures du domaine et donc procure des frontières conceptuelles rigoureusement définies par rapport au point de vue de l'observateur.
- L'abstraction est une ignorance sélective.

**L'objectif de l'abstraction n'est pas d'être vague,
mais de créer un nouveau niveau sémantique dans
lequel il est possible d'être très précis.**

Edsger Dijkstra

Une classe



Voiture: Princesse
Jaune

Marque: Renault
Modele: Clio Sport
Couleur: Jaune
Vitesse: 90

Demarrer();
Accelerer();
Ralentir();
Tourner();
Arreter();

*Un objet est donc une entité ayant un état, un comportement et une identité. La **structure** et le **comportement** d'objets similaires sont définis au niveau de leur **classe** commune.*

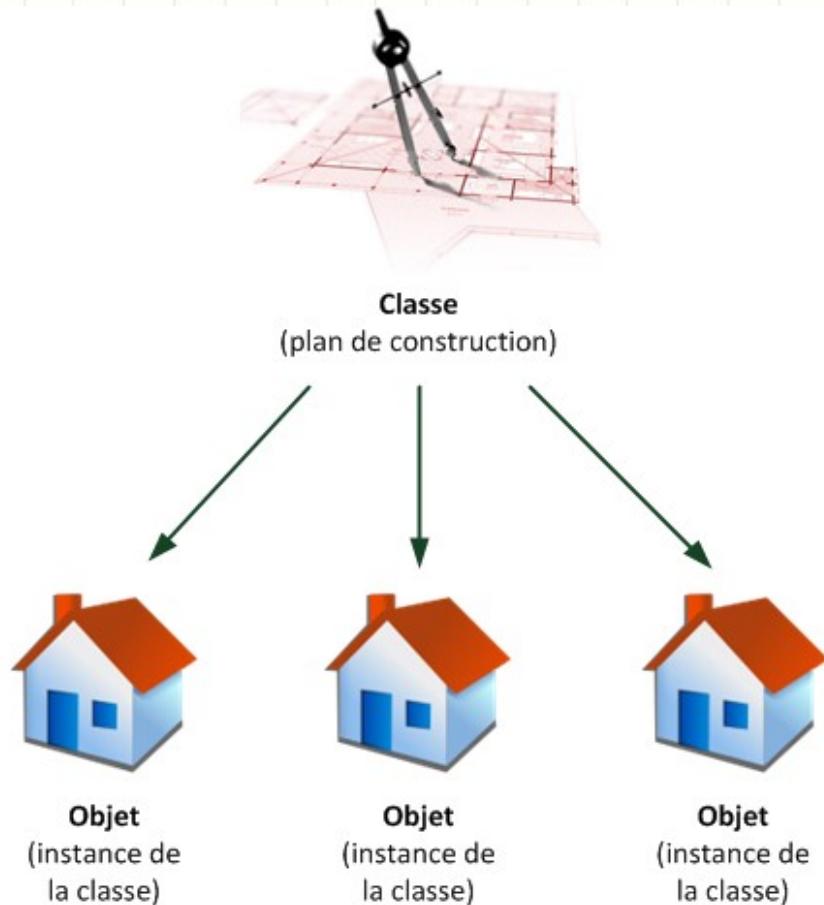


Voiture: Reine Rouge

Marque: Volkswagen
Modele: Polo VII
Couleur: Rouge
Vitesse: 110

Demarrer();
Accelerer();
Ralentir();
Tourner();
Arreter();

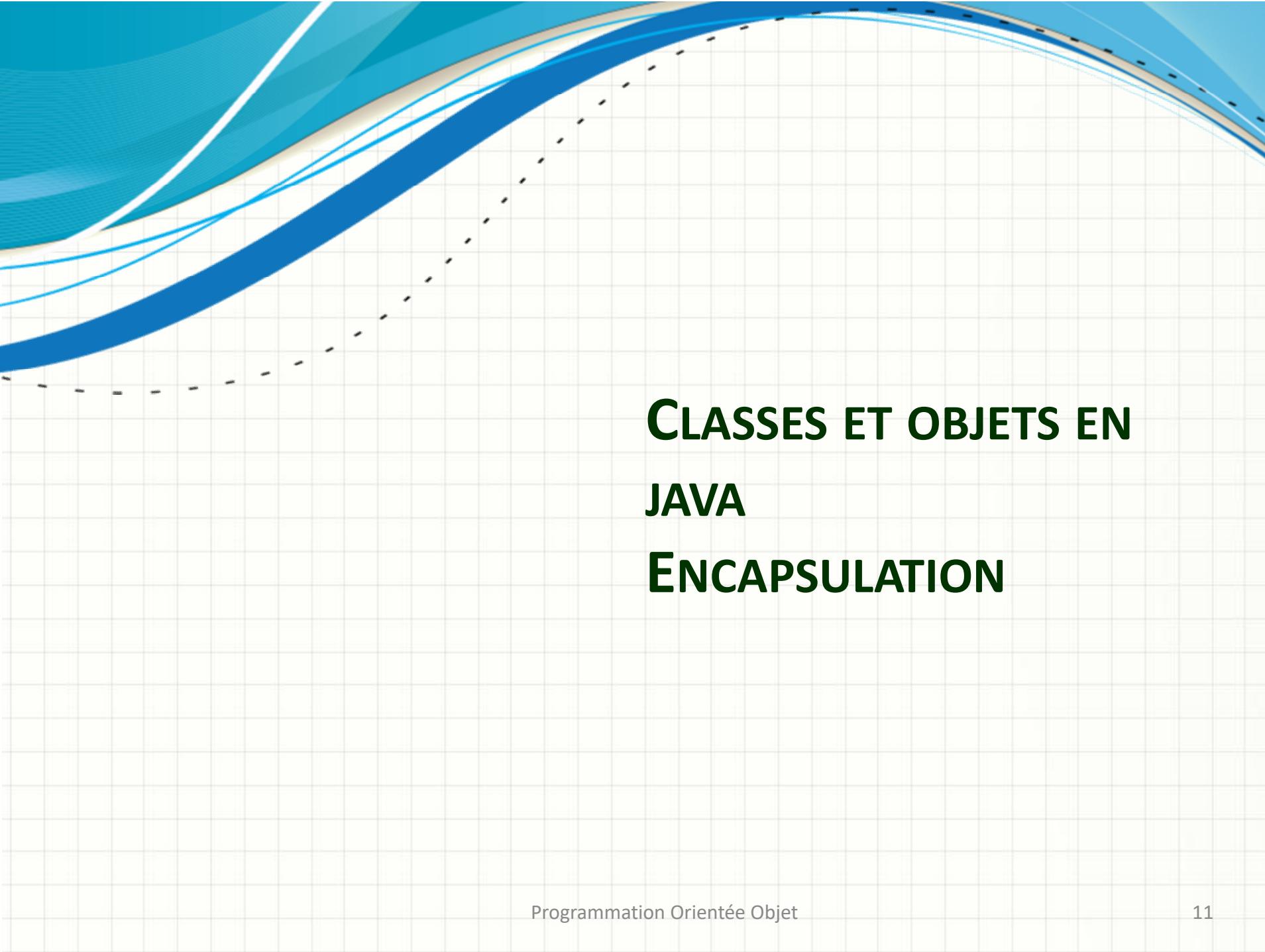
Une classe



- Pour créer un objet, il faut d'abord créer une classe.
- Au même titre que pour construire une maison, il faut déjà dessiner les plans. Créer une classe c'est en quelque sorte dessiner le plan de l'objet.
- On dit qu'un objet est une instance d'une classe.

Une classe: définition

- La classe décrit le domaine de définition d'un ensemble d'objets. Chaque objet appartient à une classe. Les généralités sont contenues dans la classe et les particularités sont contenues dans les objets. Les objets informatiques sont construits à partir de la classe, par un processus appelé **instanciation**. De ce fait, tout objet est une instance de classe.
- Une classe est l'**abstraction** d'un ensemble d'objets qui possèdent une **structure identique** (Liste des attributs) et **un même comportement** (liste des fonctions).



CLASSES ET OBJETS EN JAVA ENCAPSULATION

Identification de la classe

```
public class Voiture {  
  
    private String marque;  
    private int annee;  
    private double kilometrage;  
    private double prix;  
  
    public void configurer(String m, int a, double km, double px){  
        marque = m;  
        annee = a;  
        kilometrage = km;  
        prix = px;  
    }  
  
    public void rouler(double distance){  
        kilometrage += distance;  
    }  
  
    public void afficher(){  
        System.out.println("Marque = "+ marque);  
        System.out.println("Année = "+ annee);  
        System.out.println("Kilometrage = "+ kilometrage+" km");  
        System.out.println("Prix = "+ prix+" Kdnt");  
    }  
  
    public static void main(String[] args){  
        Voiture v1 = new Voiture();  
        v1.configurer("Ford", 2008, 103000, 32);  
        v1.afficher();  
    }  
}
```

Attributs privés

Méthodes publiques

La fonction principale main()

Instanciation de la classe Voiture

Appel de méthodes

Classes et objets en Java

- le mot-clé « **private** » précise que les attributs *marque*, *annee*, *kilometrage* et *prix* ne seront pas accessibles à l’extérieur de la classe, c’est-à-dire en dehors de ses propres méthodes.
- Le mot clé « **public** » fait que les méthodes de la classe Voiture sont accessibles depuis un programme quelconque.
- Cela correspond à l’encapsulation des données qui n’est pas obligatoire en Java. Mais, elle est fortement recommandée.
- Le mot clé « **static** » précise que la méthode *main()* de la classe Voiture n’est pas liée à une instance particulière (objet) de la classe.

Classes et objets en Java

- Considérons cette séquence de programme:

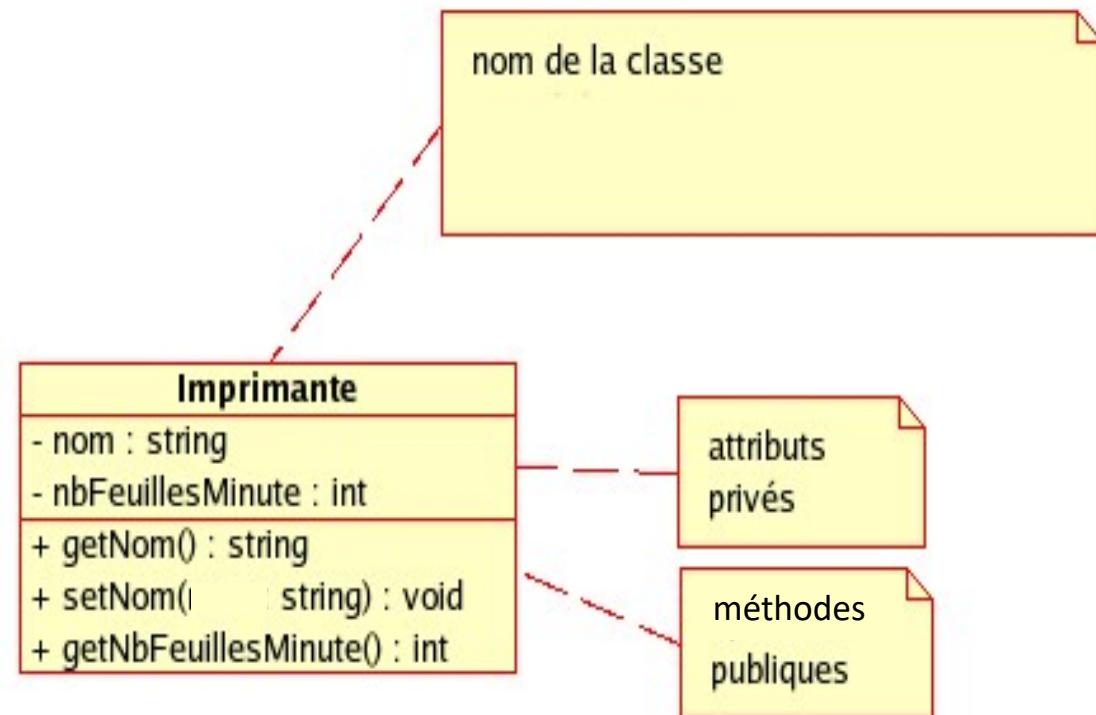
```
public static void main(String[] args) {
    Voiture v1 = new Voiture();
    Voiture v2 = new Voiture();

    v1.configurer("Ford", 2008, 103000, 19);
    v2.configurer("Toyota", 2010, 85000, 26);
    v2.afficher();
    v2.rouler(60);
    System.out.println("la distance totale parcourue par v2 est de "+
v2.kilometrage+" km");
    v2.prix += 2;
    v2.afficher();
}
```

Voiture.java

Marque = Toyota
Annee = 2010
Kilometrage = 85000.0 km
Prix = 26.0 Kdnt
la distance totale parcourue par v2 est de 85060.0 km
Marque = Toyota
Annee = 2010
Kilometrage = 85060.0 km
Prix = 28.0 Kdnt

Représentation UML d'une classe



Exercice

- Implémenter la classe Rectangle définie comme suit:

Rectangle
- longueur: int
- largeur: int
+ initialiser(int, int): void
+ primetre(): int
+ aire(): int
+ afficher(): void

- Créer un objet r1 de longueur 14 et de largeur 5.
- Afficher la valeur de ses attributs
- Afficher son aire et son périmètre.

Classes et objets en Java

- Considérons maintenant cette 2^{ème} classe Moto

```
public class Moto {  
    private String modele;  
    private String couleur;  
    private double kilometrage;  
  
    public void parametrer(String m, String cl, double km) {  
        modele = m;  
        couleur = cl;  
        kilometrage = km;  
    }  
  
    public void afficher(){  
        System.out.println("Modele = "+ modele);  
        System.out.println("Couleur = "+ couleur);  
        System.out.println("Kilometrage = "+ kilometrage+ " km");  
    }  
}
```



Encapsulation

```
public static void main(String[] args) {
    Voiture v2 = new Voiture();
    v2.configurer("Toyota", 2010, 85000, 26);
    v2.rouler(100);
    v2.prix += 2;
    System.out.println("la distance totale parcourue par v2 est de "+
v2.kilometrage+ " km");
    System.out.println("Le prix de v2 est de "+v2.prix+" kdnt");

    Moto m1 = new Moto();
    m1.parametrer("Yamaha MT-09", "noire", 1200);
    m1.afficher();
    System.out.println("La couleur de m1 est "+m1.couleur);
    m1.kilometrage += 30;
}
```

Voiture.java

Error:(37, 54) java: couleur has private access in Moto
Error:(38, 11) java: kilometrage has private access in Moto

L'appel aux attributs privés de m1 en dehors des méthodes de la classe Moto transgresse le principe d'encapsulation. Des erreurs sont signalées.

Encapsulation

- Un fichier .java ne peut contenir plus qu'une classe publique.
- Le fichier porte obligatoirement le nom de cette classe .java
- Il est également recommandé de consacrer une classe pour la fonction main(). Cette classe est dite exécutable.

Voiture.java

Moto.java

MainClass.java

Encapsulation

- La classe MainClass aurait donc cette forme:

```
public class MainClass {  
  
    public static void main(String[] args) {  
        Voiture v2 = new Voiture();  
        v2.configurer("Toyota", 2010, 85000, 26);  
        v2.rouler(100);  
        v2.afficher();  
        Moto m1 = new Moto();  
        m1.parametrer("Yamaha MT-09", "noire", 1200);  
        m1.afficher();  
    }  
}
```

MainClass.java

- Noter que tous les appels se font à travers les méthodes publiques des classes Voiture et Moto.

Encapsulation

```
public class Voiture {  
    private String marque;  
    private int annee;  
    private double kilometrage;  
    private double prix;  
  
    public void configurer(String m, int a, double km, double px) {  
        marque = m;  
        annee = a;  
        kilometrage = km;  
        prix = px;  
    }  
  
    public void rouler(double distance) {  
        kilometrage += distance;  
    }  
  
    public void afficher(){  
        System.out.println("Marque = "+ marque);  
        System.out.println("Annee = "+ annee);  
        System.out.println("Kilometrage = "+ kilometrage+" km");  
        System.out.println("Prix = "+ prix+" Kdnt");  
    }  
}
```

Voiture.java