

# MODÉLISATION DIAGRAMME DE CLASSES - ABC

1

Utiliser la notation UML pour représenter les diagrammes de classes simples et d'objets simples

# Plan

---

- **Introduction**
- **Modélisation Objet**
- **Diagrammes UML**
  - **Diagramme de classes**
  - **Diagramme d'objets**
  - **Exercice**

# Introduction

---

## Résumé

- UML est une notation, pas une méthode
- UML est un langage de **modélisation objet**
- UML convient pour toutes les méthodes objet
- UML est dans le domaine public

## Programmation Orientée Objet

- modéliser informatiquement des éléments d'une partie du monde réel en un ensemble d'entités informatiques (*objets*)

## Intérêt d'une méthode objet

- définir le problème à haut niveau sans rentrer dans les spécificités du langage
- définir un problème de façon graphique
- utiliser les services offertes par l'objet sans rentrer dans le détail de programmation (**Encapsulation**)
- Réutilisation du code

# Modélisation objet: l'objet ou l'instance

## Notion d'*Objet*

Une abstraction du monde réel c.-à-d. des données informatiques regroupant des caractéristiques du monde réel

## Exemple

une personne, une voiture, une maison, ...

## Caractérisation d'un objet

### ➤ **Identité** →

permet de le distinguer des autres objets

### ➤ **Etats** →

données caractérisant l'objet

### ➤ **Comportements** →

actions que l'objet est à même de réaliser

<u>flat-uno-17 : Voiture</u>
numeroDeSerie= 5323454 Poids= 1500 Immatriculation= 64 YFT 17 kilométrage= 23 000
Démarrer () Arrêter() Rouler()

# Modélisation objet: l'objet ou l'instance

---

- C'est une entité atomique qui possède :
  - une identité qui le caractérise de façon non ambiguë,
  - un état représenté par le contenu de ses attributs et les liens qu'il a avec les autres objets,
  - un comportement qui regroupe les compétences d'un objet et décrit ses actions et ses réactions (messages reçus , messages envoyés).
- Les objets communiquent entre eux par des messages.
- La persistance des objets est la capacité pour un objet de sauvegarder son état dans un système de stockage de l'information.

# Modélisation objet

---

Un objet a une vie :

- il naît,
- il vit,
- il meurt.

Il est symbolisé par :

Nom objet

ou

Nom objet : Classe

ou

: Classe

Le stéréotype de la classe peut surmonter le nom

<<StéréotypeClasse>>  
Nom objet : Classe

# Modélisation objet: la classe

---

## Notion de Classe

- Structure d'un objet, c.-à-d. une déclaration de l'ensemble des entités qui composeront l'objet
- Un objet est donc "issu" d'une classe, c'est le produit qui sort d'un moule

## Notation

un objet est une **instanciation** (***occurrence***) d'une classe

## Une classe est composée:

### ➤ attributs

données dont les valeurs représentent l'état de l'objet

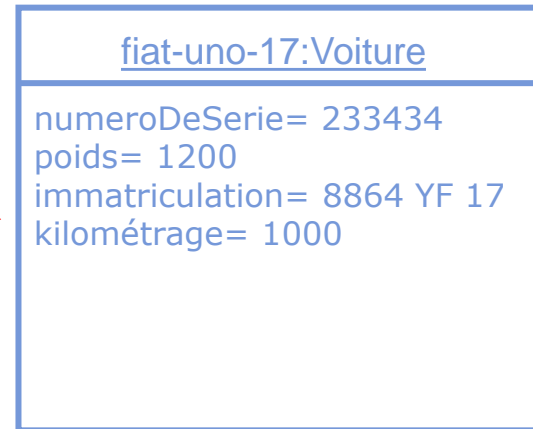
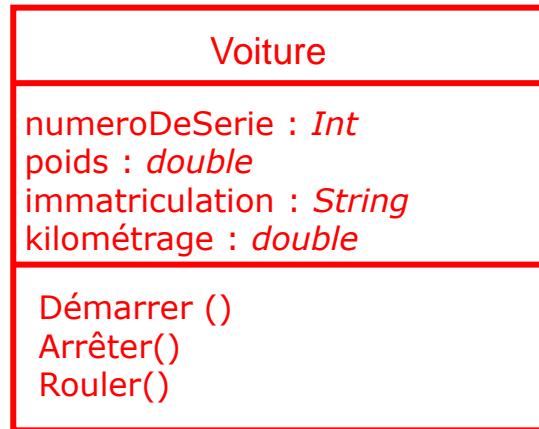
### ➤ méthodes/opérations

opérations applicables aux objets

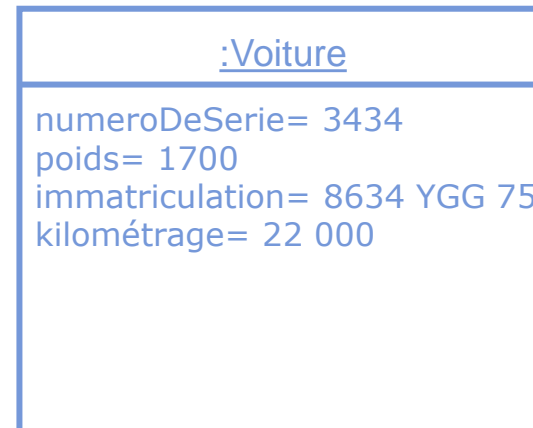
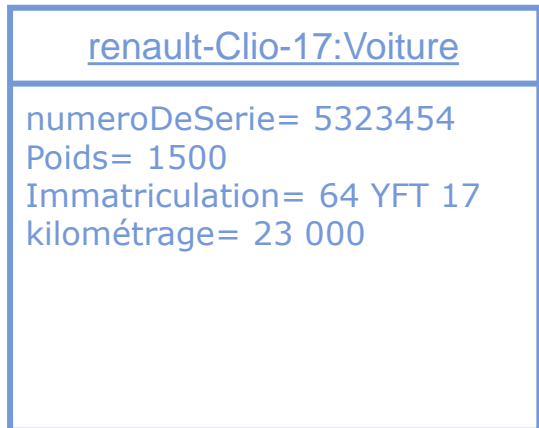
Nom_de_la_classe
attribut1 : Type attribut2 : Type ...
Methode1 () Methode2 () ...

# Modélisation objet

Classe



Object





# Modélisation objet

---

## Visibilité des attributs

définissent les droits d'accès aux données (pour la classe elle-même, d'une classe héritière, ou bien d'une classe quelconque)

### ➤ **Publique (+)**

les classes peuvent accéder aux données et méthodes d'une classe définie avec le niveau de visibilité *public*

➤ **Protégée (#)**: l'accès aux données est réservé aux fonctions des classes héritières

➤ **Privée (-)**: l'accès aux données est limité aux méthodes de la classe elle-même

Nom_de_la_classe
# Attribut1 : Type - Attribut2 : Type ...
+ méthode1 () Méthode2 () ...

# Attribut de classe et attribut d'instance

---

- Un attribut d'instance a une valeur différente d'un objet à l'autre.
- Pour atteindre un attribut d'instance :

**nomObjet.nomAttribut**

- Un attribut de classe a une valeur pour la classe; il est précédé par le mot clé « **static** ».
- Pour atteindre un attribut de classe :

**NomClasse.nomAttribut**

# Méthode de classe et méthode d'instance

---

- Une méthode d'instance représente un comportement propre à chaque instance.
- Une méthode de classe représente un comportement commun et ne dépendant donc pas de l'état de l'objet.
- Pour atteindre un méthode d'instance :  
**nomObjet.Methode();**
- Un méthode de classe est précédé par le mot clé « **static** ».
- Pour atteindre un méthode de classe :  
**NomClasse.Methode();**

# Classes et objets: Les catégories de méthode

---

- *Méthodes Constructeur*

*Elles permettent de construire un nouvel objet, une nouvelle instance de la classe*

- *Méthodes de classe*

*caractérisées en C# par le mot clé static*

- *Méthodes d'instance :*

*Elle s'adressent aux objets et non à la classe*

- *Modifieur : méthode qui met à jour une donnée privée (« set »)*
- *Accesseur : méthode retournant la valeur d'une donnée privée (« get »)*
- *Autres méthodes*

- *Méthodes Destructeur*

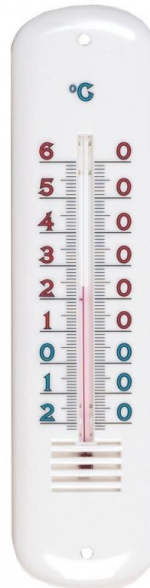
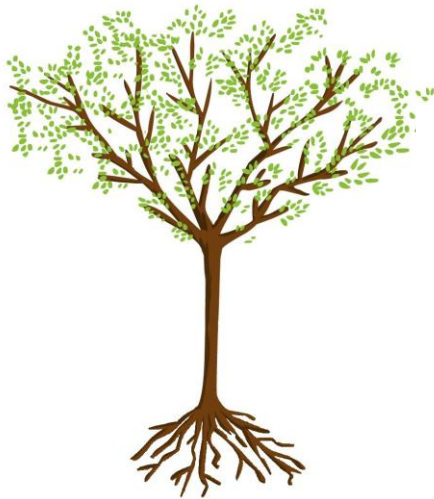
*Elles permettent de détruire les objets.*

- *Méthodes Itérateur*

*Elles permettent de visiter l'état d'un objet ou le contenu d'une structure de données*

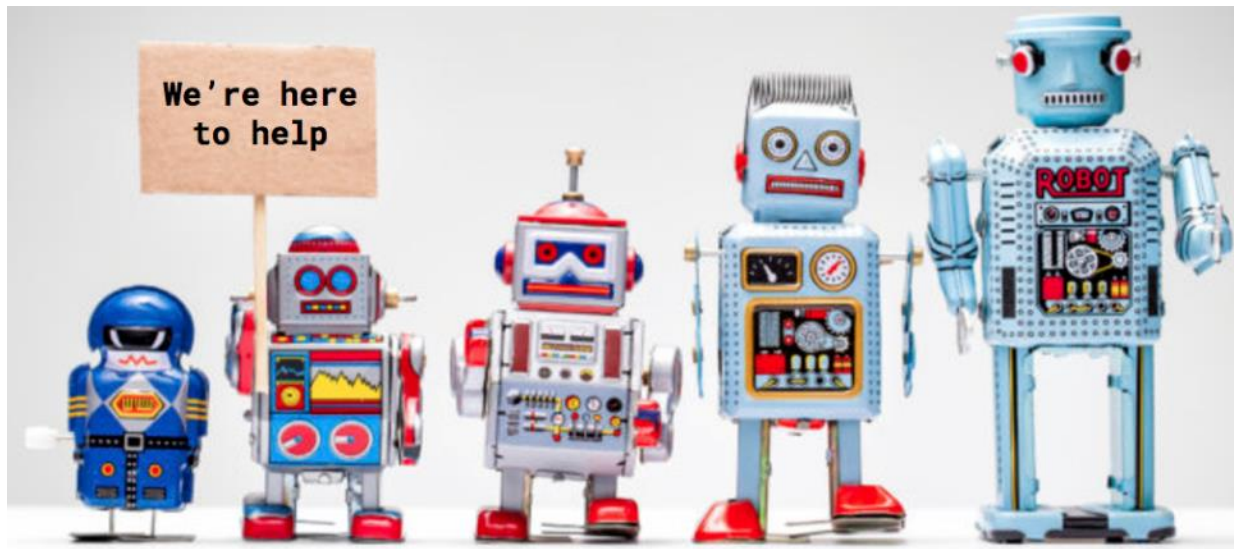
# EXERCICES

- 1. Essayez pour ces différents objets de citer des exemples d'états de l'objet, des exemples de comportements de l'objet.



# EXERCICES

2.Représentez sous la forme d'objet plusieurs robots. Si vous manquez d'imagination, en voici de beaux modèles:



3.Essayez de représenter une Classe qui servirait de moule aux objets que vous avez fait précédemment.

# EXERCICES

- 3. Vous devez modéliser une porte de garage que l'on peut ouvrir et fermer, mais également verrouiller et déverrouiller. La porte de garage peut être partiellement ouverte. Proposez d'abord 4 objets différents, représentant cette porte de garage dans un état différent, puis une Classe **PorteDeGarage**.



# EXERCICES

- 4. Proposez un objet de votre choix que vous allez modélisez