

Cours 1 : Introduction au monde des objets

Le concept d'objet
Les liens entre objets
La notion de classe
Les relations entre classes
Introduction à l'approche orientée objets

Auteur : Christelle CHAUDET

Christelle CHAUDET

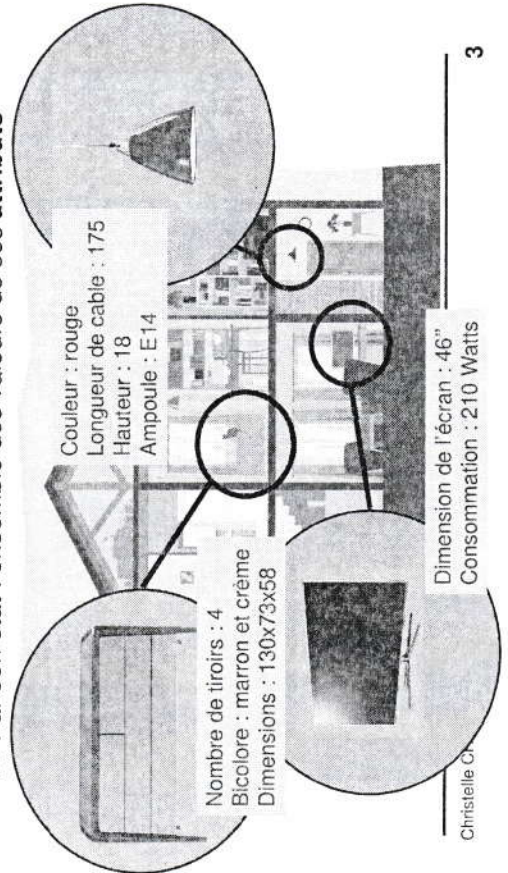
2

Le concept d'objet

- Un objet est une entité du monde réel ou du monde informatique
- Exemples d'objets :
 - abstractions du monde réel :
 - une voiture, une personne, un compte bancaire
 - un port, un bateau, une grue
 - entités du monde informatique :
 - une fenêtre, un menu, un bouton, une icône
 - un entier, une chaîne de caractères
 - une collection : tableau, liste ...

Le concept d'objet

- Comment décrire un objet ?
 - Par son état : ensemble des valeurs de ses attributs

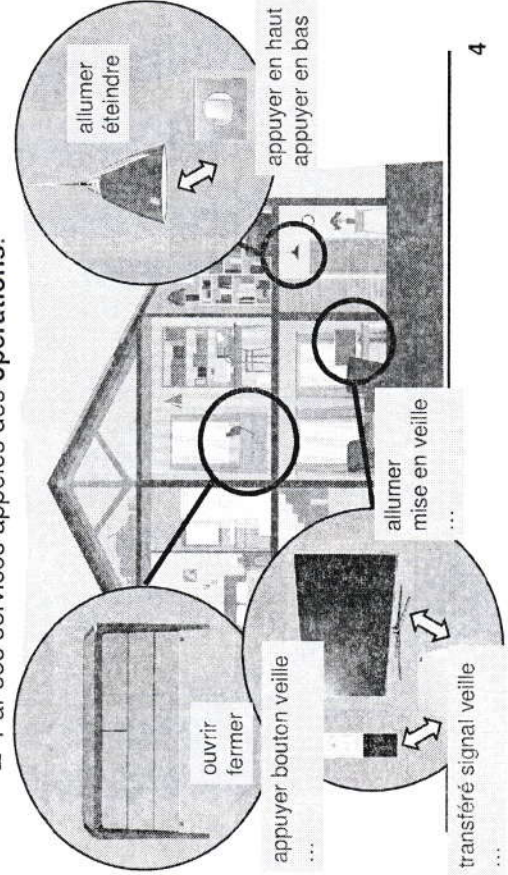


Christelle CH

3


Le concept d'objet

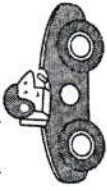
- Comment décrire un objet ?
 - Par ses services appelés des opérations.



4

Le concept d'objet

- Un objet possède des **caractéristiques** :
 - **Attributs** qui stockent de l'information (des données)
 - **Services** qu'il offre à celui qui l'utilise (ce qu'il sait faire)
- 
- Exemple : une voiture
 - **Attributs** :
 - sa marque, son modèle, sa couleur, son type de moteur, sa cylindrée, son numéro de série, son immatriculation ...
 - mais aussi : son kilométrage, sa vitesse instantanée ...
 - **Services (ou opérations)** :
 - démarrer, avancer, arrêter, reculer, éteindre ...
 - kilométrage ? vitesse instantanée ? ...



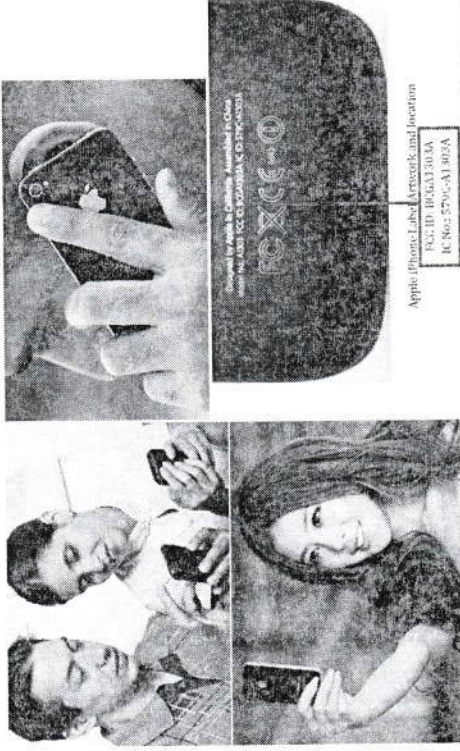
Le concept d'objet

- **Objet = Identité + Etat + Comportement**
 - Identité : propre à chaque objet
 - permet de le distinguer des autres

la voiture AB-123-CD

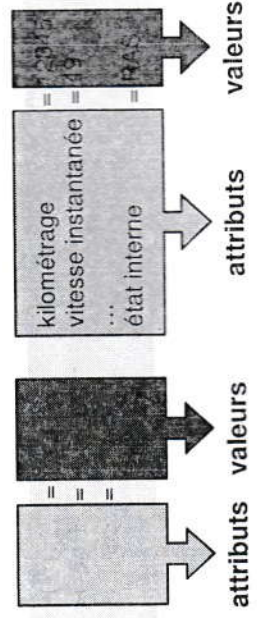
Le concept d'objet

- Chacun des objets est unique !



Le concept d'objet

- **Objet = Identité + Etat + Comportement**
 - Identité : propre à chaque objet
 - permet de le distinguer des autres
 - Etat :
 - l'objet stocke les valeurs de ses attributs (données)



Le concept d'objet

- **Objet = Identité + Etat + Comportement**
 - Identité : propre à chaque objet
 - permet de le distinguer des autres
- **Etat :**
 - l'objet stocke les valeurs de ses attributs (données)
 - l'état de l'objet est caractérisé par ce n-uplet de valeurs

marque = Renault	kilométrage	= 12345
modèle = Twingo	vitesse instantanée	= 49
couleur = rouge	...	
...	état interne	= RAS

État de l'objet

Christelle CHAUDET

9

Le concept d'objet

- **Objet = Identité + Etat + Comportement**
 - Identité : propre à chaque objet
 - permet de le distinguer des autres
- **Etat :**
 - l'objet stocke les valeurs de ses attributs (données)
 - l'état de l'objet est caractérisé par ce n-uplet de valeurs
- **Comportement :**
 - les opérations permettent :
 - de modifier cet état (**modificateurs**)
 - de le connaître (**accesseurs**)

	Etat	Accesseurs
...		
kilométrage = 12348		kilométrage ?
vitesse instantanée = 50		vitesse ?
		12348
		50

Christelle CHAU

11

Le concept d'objet

- **Objet = Identité + Etat + Comportement**
 - Identité : propre à chaque objet
 - permet de le distinguer des autres
- **Etat :**
 - l'objet stocke les valeurs de ses attributs (données)
 - l'état de l'objet est caractérisé par ce n-uplet de valeurs
- **Comportement :**
 - les opérations permettent :
 - de modifier cet état (**modificateurs**)
 - de modifier cet état (**modificateurs**)

Modificateurs démarrer, avancer, s'arrêter, reculer, éteindre ...

...	8	
...	7	
...	6	
kilométrage = 12345	5	50
vitesse instantanée = 49		
...		

État

Christelle CHAUDET

10

Le concept d'objet

- **Objet = Identité + Etat + Comportement**
 - Identité : propre à chaque objet
 - permet de le distinguer des autres
- **Etat :**
 - l'objet stocke les valeurs de ses attributs (données)
 - l'état de l'objet est caractérisé par ce n-uplet de valeurs
- **Comportement :**
 - les opérations permettent :
 - de modifier cet état (**modificateurs**)
 - de le connaître (**accesseurs**)
 - elles sont étroitement liées aux attributs
 - elles permettent à l'objet de réagir à des sollicitations extérieures, en agissant éventuellement sur d'autres objets

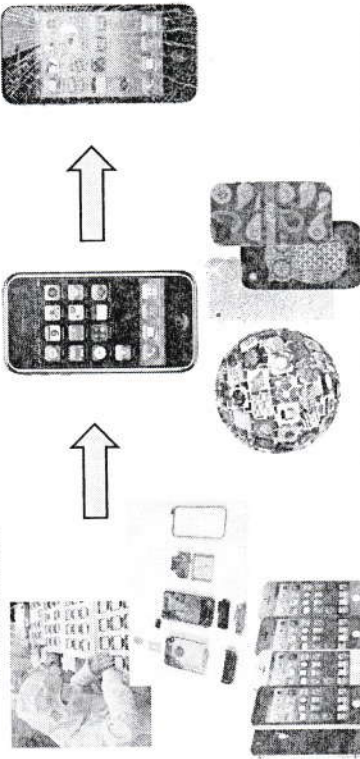
Christelle CHAUDET

12

Le concept d'objet

- Tout objet a une existence temporelle, plus ou moins longue :

- création,
- évolution,
- destruction.

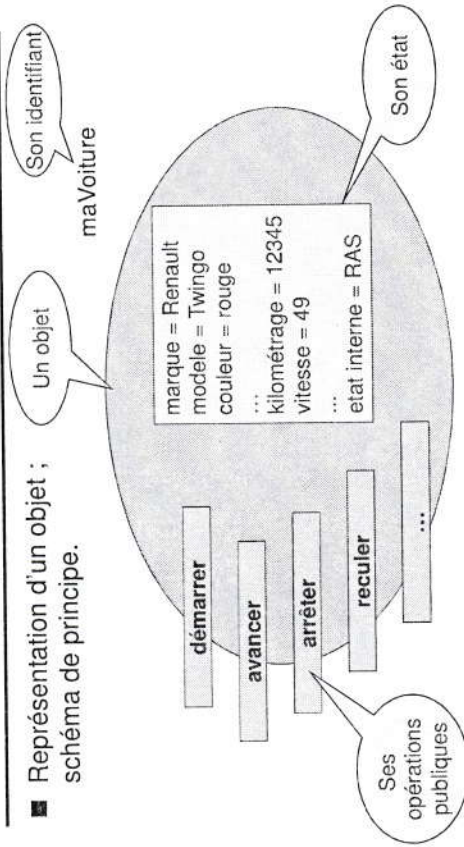


Christelle CHAUDET

13

Le concept d'objet

- Représentation d'un objet ; schéma de principe.



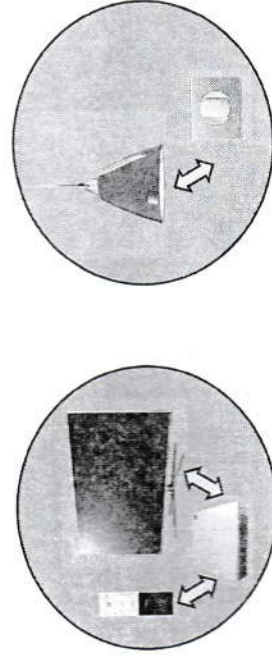
Christelle CHAUDET

14

Le concept d'objet

- En attendant le prochain cours voici un aperçu de l'objet « maVoiture » en notation UML :

maVoiture : Voiture
marque = "Renault"
modele = "Twingo"
couleur = "rouge"
...
kilométrage = 12345
vitesse = 49
...
etatInterne = "RAS"



Christelle CHAUDET

15

Les liens entre objets

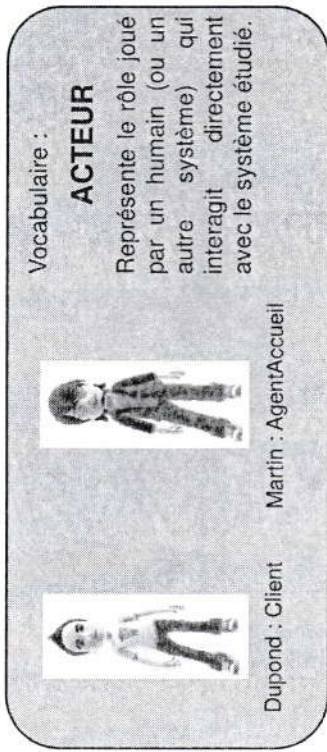
- Maintenant que nous avons une bonne notion de ce qu'est un objet, nous allons voir comment ces objets travaillent ensemble.

Christelle CHAUDET

16

Les liens entre objets

- Prenons par exemple, une personne (monsieur Dupond) qui va dans une agence de livraison pour transporter 3 tonnes de cailloux le 18 Décembre. L'agent d'accueil (monsieur Martin) va traiter sa demande à l'aide d'objets.



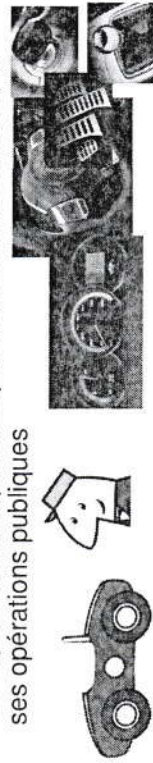
Christelle CHAUDET

17

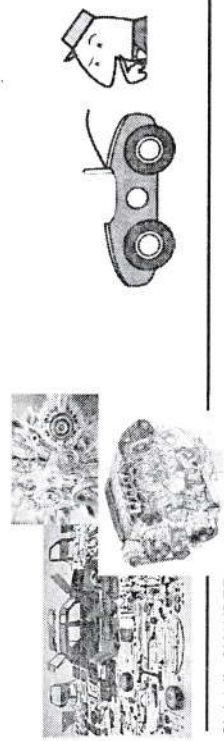
Les liens entre objets

- Remarques :

- Un objet n'est manipulable qu'à travers son **interface**, ie ses opérations publiques



- On n'a pas besoin de connaître le fonctionnement interne d'un objet pour pouvoir l'utiliser !

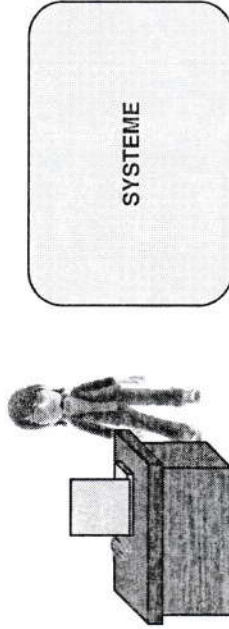


Christelle CHAUDET

19

Les liens entre objets

- Notre **acteur** Monsieur Martin
 - a pour rôle « agent d'accueil »,
 - communique avec son système de réservation grâce à un ordinateur :
 - entre les données grâce à un clavier et à une souris,
 - observe les réponses du système sur un écran.

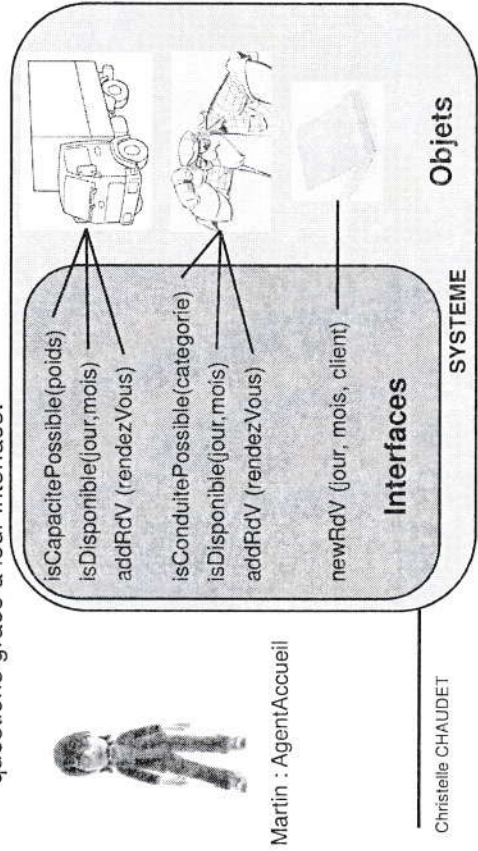


Christelle CHAUDET

18

Les liens entre objets

- Donc l'agent d'accueil (monsieur Martin) ne connaît pas le fonctionnement des objets mais peut poser certaines questions grâce à leur interface.



Christelle CHAUDET

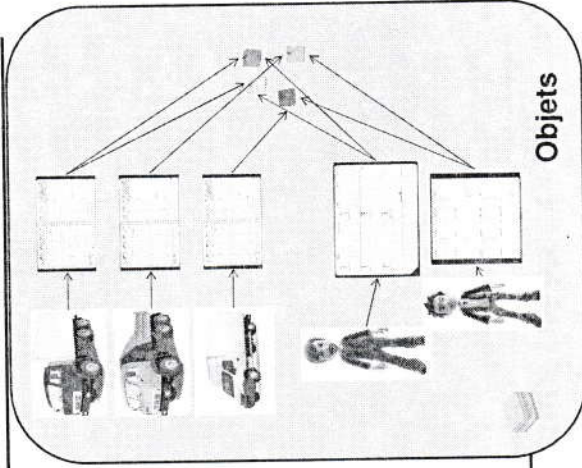
Les liens entre objets

Je veux louer un
véhicule pour transporter
3 tonnes de cailloux le
18 Décembre

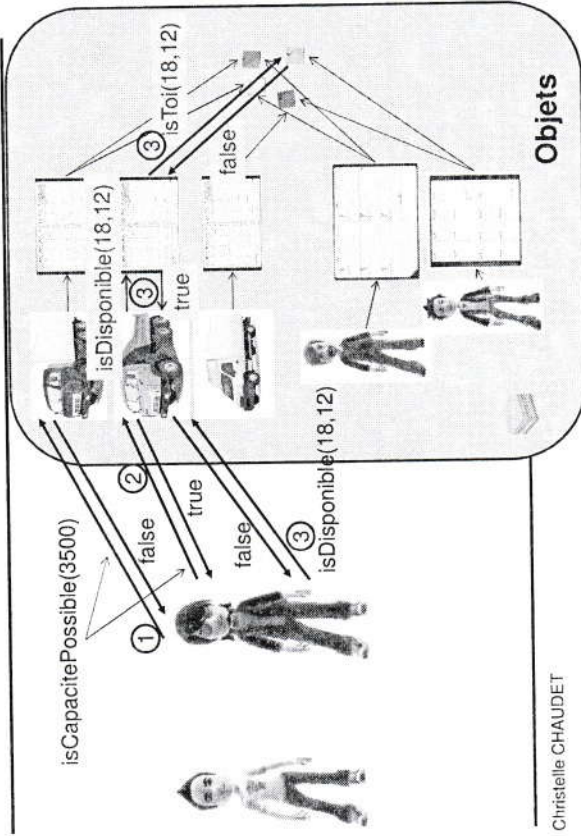


Dupond : Client Martin : AgentAccueil

Christelle CHAUDET

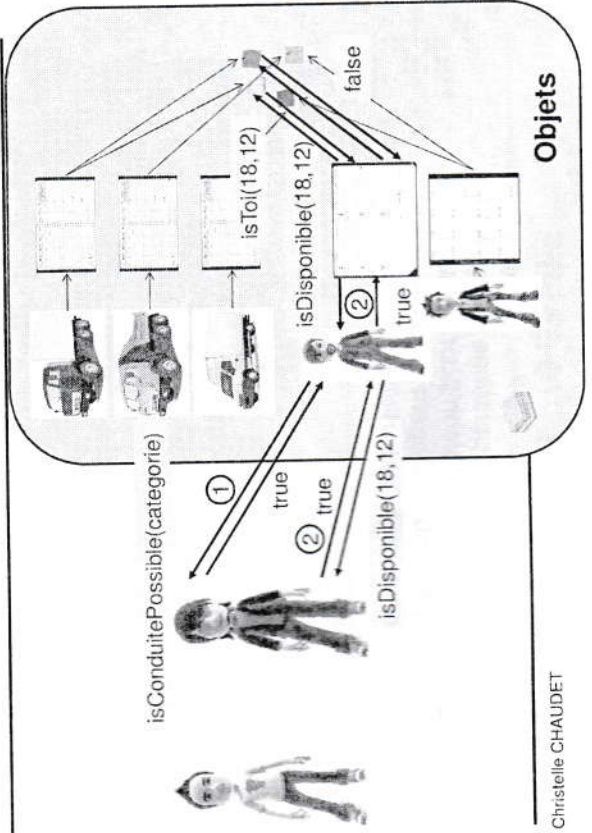


Les liens entre objets



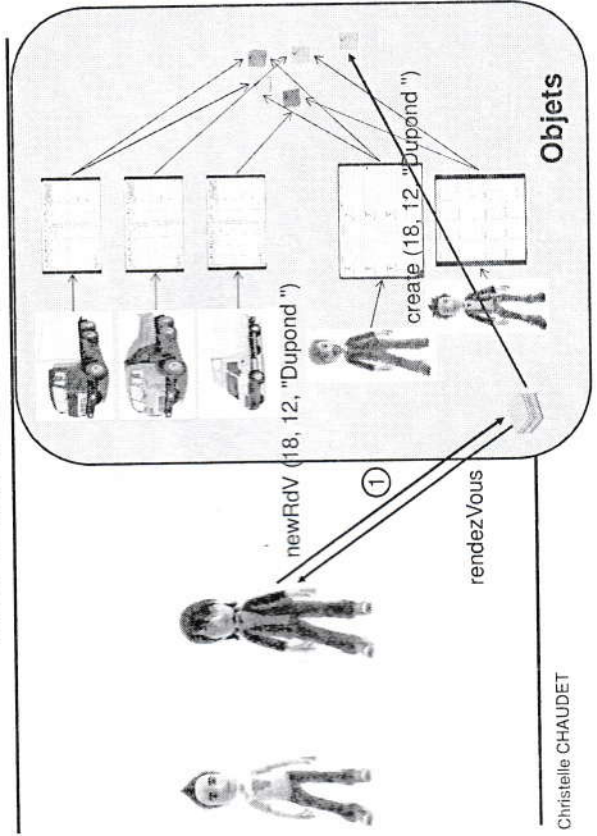
Christelle CHAUDET

Les liens entre objets



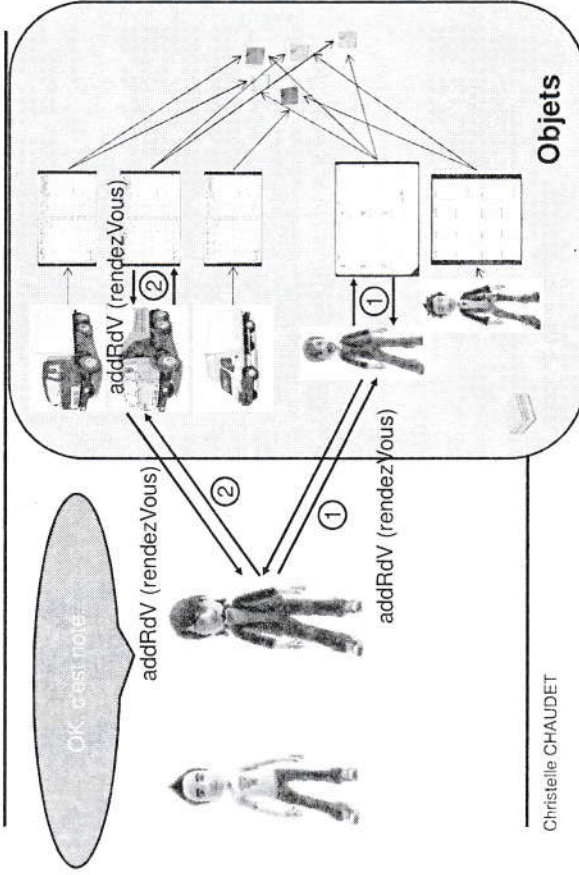
Christelle CHAUDET

Les liens entre objets



Christelle CHAUDET

Les liens entre objets



Christelle CHAUDET

26

Les liens entre objets

- Tout objet a une **existence temporelle**, plus ou moins longue :

- création : par un autre objet



- évolution : au fil des utilisations par d'autres objets



- destruction : par un autre objet



Christelle CHAUDET

Les liens entre objets

- Tout objet a une **existence temporelle**, plus ou moins longue :

- création : par un autre objet

- évolution : au fil des utilisations par d'autres objets

- destruction : par un autre objet.

- Tout objet est en **relation avec d'autres objets** :

- liens statiques : (relativement) stables dans le temps

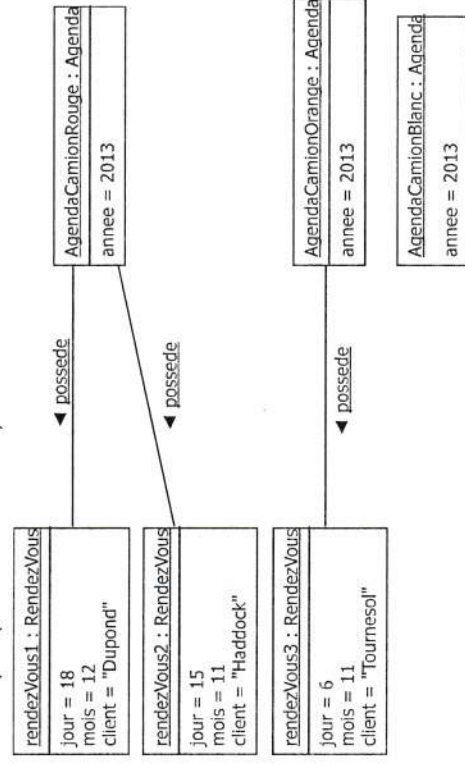


Christelle CHAUDET

27

Les liens entre objets

- Exemple (en notation UML) :

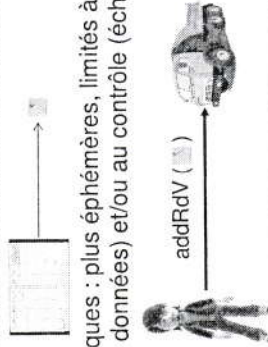


Christelle CHAUDET

28

Les liens entre objets

- Tout objet a une **existence temporelle**, plus ou moins longue :
 - création : par un autre objet
 - évolution : au fil des utilisations par d'autres objets
 - destruction : par un autre objet.
- Tout objet est en **relation avec d'autres objets** :
 - liens statiques : (relativement) stables dans le temps
 - liens dynamiques : plus éphémères, limités à la communication (échange de données) et/ou au contrôle (échange de stimuli).

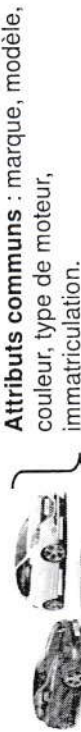


Christelle CHAUDET

29

La notion de classe

- **Classe** = **abstraction d'objets** ayant des propriétés (attributs, comportement et relations) communes
- **Type de données abstrait** qui :
 - définit les caractéristiques (attributs et opérations) et les relations communes à une famille d'objets



Attributs communs : marque, modèle, couleur, type de moteur, immatriculation.

- **Opérations communes** : démarrer, avancer, s'arrêter, reculer, éteindre.



Christelle CHAUDET

31

Les liens entre objets

- Tout objet a une **existence temporelle**, plus ou moins longue :
 - création : par un autre objet
 - évolution : au fil des utilisations par d'autres objets
 - destruction : par un autre objet.
- Tout objet est en **relation avec d'autres objets** :
 - liens statiques : (relativement) stables dans le temps
 - liens dynamiques : plus éphémères, limités à la communication (échange de données) et/ou au contrôle (échange de stimuli).
- Le modèle Orienté-Objet (OO) d'une application définit un ensemble d'objets coopérants, communicants et parfois concurrents, vivant indépendamment les uns des autres.
- Vocabulaire : les objets communiquent par **envoi de messages**.

Christelle CHAUDET

30

La notion de classe

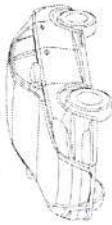
- **Classe** = **abstraction d'objets** ayant des propriétés (attributs, comportement et relations) communes
- **Type de données abstrait** qui :
 - définit les caractéristiques (attributs et opérations) et les relations communes à une famille d'objets
 - permet de créer (instancier) des objets possédant ces caractéristiques
- Classe = moule, patron qui permet de créer des objets

Christelle CHAUDET

32

La notion de classe

■ Classe



En notation UML :

Voiture
-marque: string
-modele: string
-couleur: string
-typeMoteur: string
-immatriculation: string
+demarrer()
+avancer()
+arreter()
+reculer()
+eteindre()

■ Objet



En notation UML :

voitureHaddock : Voiture
marque = "Renault"
modele = "Clio"
couleur = "bleue"
typeMoteur = "Diesel"
immatriculation = "AB-521-UH"

voitureTournesol : Voiture
marque = "peugeot"
modele = "208"
couleur = "blanche"
typeMoteur = "essence"
immatriculation = "RT-845-FG"

voitureCastafiore : Voiture
marque = "Citroen"
modele = "DS"
couleur = "rouge"
typeMoteur = "essence"
immatriculation = "UR-158-OJ"

Christelle CHAUDET

33

La notion de classe

- **Classe = abstraction d'objets** ayant des propriétés (attributs, comportement et relations) communes

■ Type de données abstrait qui :

- définit les caractéristiques (attributs et opérations) et les relations communes à une famille d'objets
- permet de créer (instancier) des objets possédant ces caractéristiques
- Classe = moule, patron qui permet de créer des objets
- Classe : n'a de sens qu'à la compilation

■ Objet = instance d'une classe

- Chaque objet est créé à partir de sa classe (**instanciation**)
- Objet : n'a d'existence qu'à l'exécution

Christelle CHAUDET

34

La notion de classe

■ Chaque classe :

- a un **nom**
- définit les **attributs** possédés par ses instances
- définit les **opérations** partagées par ses instances et qui permettent d'accéder aux attributs et de les modifier

■ Exemple (en notation UML) :

Voiture
marque: string
modele: string
couleur: string
typeMoteur: string
immatriculation: string
demarrer()
avancer()
arreter()
reculer()
eteindre()

nom de la classe

attributs

opérations

Voiture
marque: string
modele: string
couleur: string
typeMoteur: string
immatriculation: string
demarrer()
avancer()
arreter()
reculer()
eteindre()

voitureHaddock : Voiture
marque = "Renault"
modele = "Clio"
couleur = "bleue"
typeMoteur = "Diesel"
immatriculation = "AB-521-UH"

voitureCastafiore : Voiture
marque = "Citroen"
modele = "DS"
couleur = "rouge"
typeMoteur = "essence"
immatriculation = "UR-158-OJ"

voitureTournesol : Voiture
marque = "peugeot"
modele = "208"
couleur = "blanche"
typeMoteur = "essence"
immatriculation = "RT-845-FG"

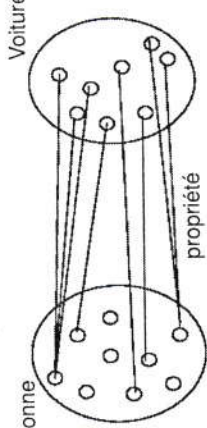
Christelle CHAUDET

35

Christelle CHAUDET

36

Les relations entre classes

- Deux points de vue :
 - Une relation met en correspondance des éléments d'ensembles.
- 
- Une relation permet la description d'un concept à l'aide d'autres concepts.
 - Une contrainte :
 - Une relation décrit un lien **stable** entre deux objets

Christelle CHAUDET

37

Les relations entre classes

- Association = abstraction de liens entre objets
 - Bidirectionnelle

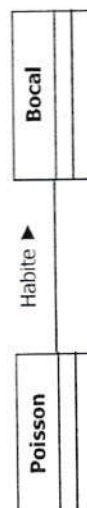


Christelle CHAUDET

38

Les relations entre classes

- Association = abstraction de liens entre objets
 - Bidirectionnelle
 - Possibilité de lui donner un **nom**
- Possibilité d'indiquer un sens de lecture

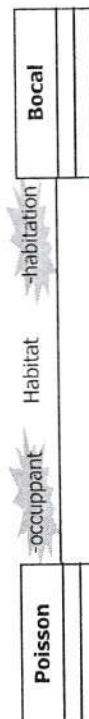


Christelle CHAUDET

47

Les relations entre classes

- Association = abstraction de liens entre objets
 - Bidirectionnelle
 - Possibilité de lui donner un **nom**
 - Possibilité d'indiquer les **rôles** joués par les extrémités

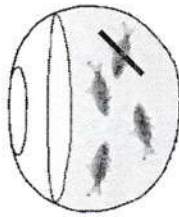


Christelle CHAUDET

48

Les relations entre classes

- Association = abstraction de liens entre objets
- Bidirectionnelle
- Possibilité de lui donner un nom
- Possibilité d'indiquer les rôles joués par les extrémités
- Possibilité d'indiquer les multiplicités des extrémités

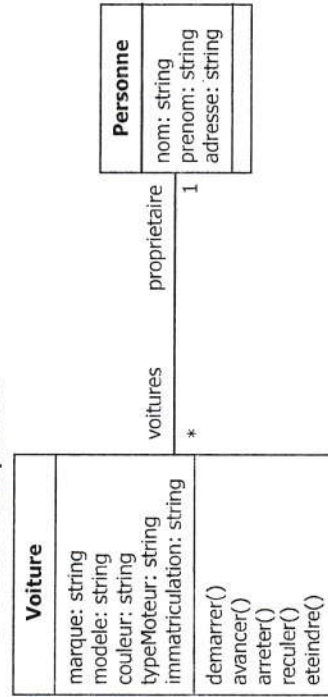


Christelle CHAUDET

49

Les relations entre classes

- Exemple d'association (en notation UML) :
- anonyme
- avec un nom de rôle
- avec les multiplicités

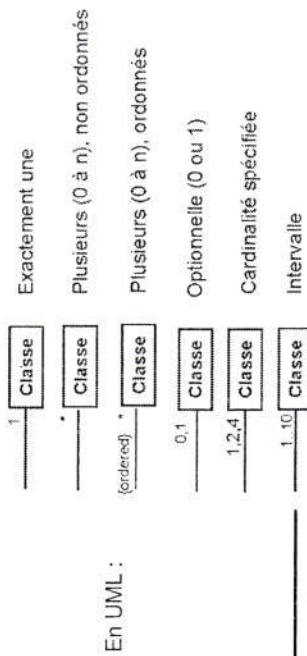


Christelle CHAUDET

43

Les relations entre classes

- Association = abstraction de liens entre objets
- Bidirectionnelle
- Possibilité de lui donner un nom
- Possibilité d'indiquer les rôles joués par les extrémités
- Possibilité d'indiquer les multiplicités des extrémités

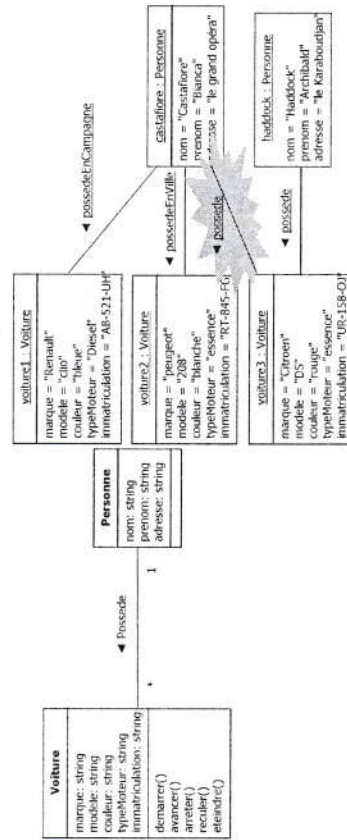


Christelle CHAUDET

42

Les relations entre classes

- Association entre Classes :
- lie 2 classes
- décrit un ensemble de liens
- est figée
- Lien entre Objets :
- lie 2 objets
- est une instance d'association
- peut être créé, modifié, détruit

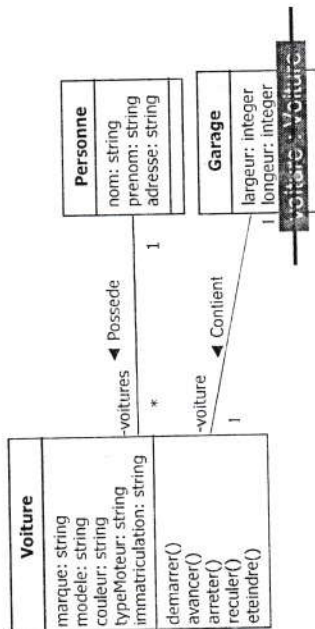


Christelle CHAUDET

44

Les relations entre classes

- Attention, si une notion existe en tant que classe, on ne peut pas l'utiliser comme attribut. Il faut créer un lien.

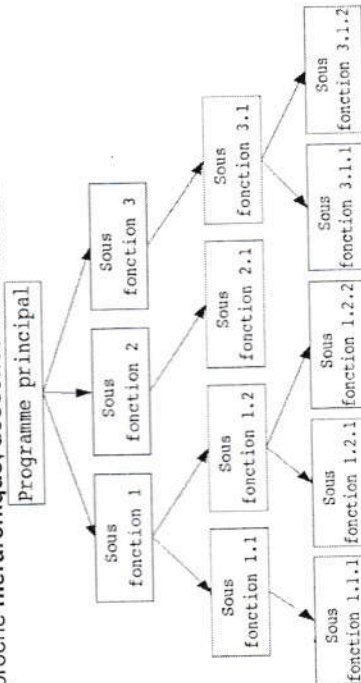


Les relations entre classes

- Dans le prochain cours nous expliquerons d'autres relations entre classes :
 - des formes particulières d'associations, permettant d'assembler des objets pour construire des objets plus complexes :
 - **Agrégation**
 - **Composition**
 - d'autres types de relations entre classes :
 - **Généralisation / spécialisation**

Introduction à l'approche orientée objets

- **Approche fonctionnelle (ou structurée) :**
 - Basée sur les **fonctions** à assurer et les **raffinements successifs**
 - Approche **hiérarchique, descendante et modulaire**

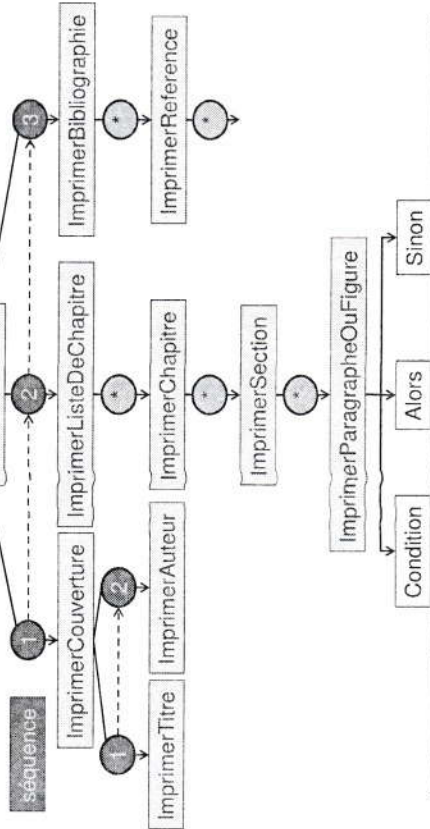


Introduction à l'approche orientée objets

- **Approche fonctionnelle (ou structurée) :**
 - Exemple : Imprimer un livre
 - Un livre est constitué d'un en-tête, d'une liste de chapitres et d'une bibliographie
 - Chaque chapitre est constitué d'une suite de sections, elles-mêmes composées de paragraphes et de figures
 - procédure `imprimerLivre` :
 - `imprimerEntête`;
 - `imprimerListeDesChapitres`;
 - `imprimerBibliographie`;
 - puis décomposition plus fine

Introduction à l'approche orientée objets

■ Approche fonctionnelle (ou structurée) :



Christelle CHAUDET

49

Introduction à l'approche orientée objets

■ Approche fonctionnelle (ou structurée) :

- Accès aux données :
 - soit **direct** (par le bas : une base de données, par ex.)
 - soit par **passage de paramètres** (par le haut, éventuellement depuis le programme principal)
- Les données peuvent être **éparpillées** dans toute la hiérarchie
- L'**architecture du système** est dictée par la **réponse au problème** (ie la fonction du système)
 - Un **changement de fonction** peut impacter toute la structure
 - Un **changement** dans la structure des données peut impacter de nombreux modules, difficiles à identifier

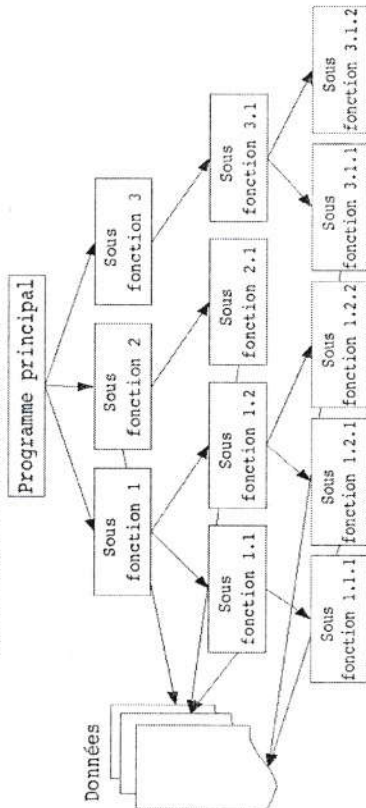
Christelle CHAUDET

51

Introduction à l'approche orientée objets

■ Approche fonctionnelle (ou structurée) :

- Dissocie représentation et traitement des données
 - Organisation des données **postérieure** à l'identification des fonctions et sous-fonctions



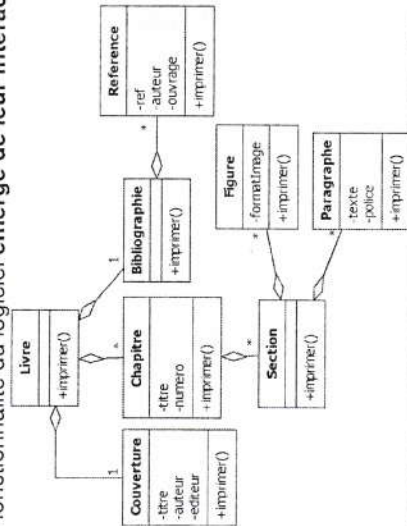
Christelle CHAUDET

50

Introduction à l'approche orientée objets

■ L'approche orientée objets :

- considère le logiciel comme une **collection d'objets dissociés**,
- la fonctionnalité du logiciel émerge de leur **interaction**.



Christelle CHAUDET

52

Introduction à l'approche orientée objets

- L'approche orientée objets :
 - Rapprocher les données et les traitements associés
 - au sein d'un unique objet
 - formant une entité cohérente, logique, tangible et stable vis-à-vis du problème traité
- Les objets du système sont plus stables que les fonctions qu'il réalise
- L'architecture du système est dictée par la structure du problème lui-même (et non plus par la fonction du système)
 - Une évolution des besoins se traduira le plus souvent par un simple changement des interactions entre objets
 - Si un changement dans les données est nécessaire, il n'impacte que les objets qui les encapsulent, bien identifiés

Christelle CHAUDET

53

Introduction à l'approche orientée objets

- Vocabulaire :
 - Encapsulation :
 - regroupement des informations d'état et de comportement en une entité unique
 - permet de se concentrer sur une classe et une seule
 - facilite les modifications à portée locale
- Interface :
 - vue externe d'un objet
 - elle définit les services offerts aux utilisateurs de l'objet
 - la dissociation interface/implantation favorise la réutilisation

Christelle CHAUDET

54

Introduction à l'approche orientée objets

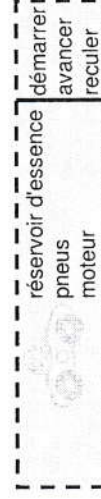
- Comment concevoir une solution selon l'approche orienté objet ?
 - Qu'est-ce qu'un bon objet ?
 - Quelle stratégie pour établir la liste d'objets ?

Christelle CHAUDET

55

Introduction à l'approche orientée objets

- Un bon objet doit contenir tout ce dont il a besoin pour accomplir sa tâche.



- Il ne doit pas cependant accomplir d'action qu'un autre objet doit réaliser.
- Il ne doit pas contenir d'informations dont il n'est pas directement responsable.

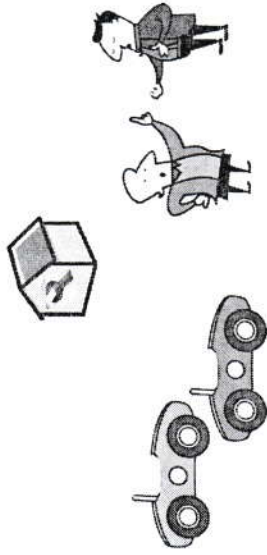


Christelle CHAUDET

56

Introduction à l'approche orientée objets

- Les objets représentent les éléments d'une solution que vous souhaitez mettre en place.
- Exemple : pour la gestion d'un garage, les objets seront :
 - les voitures,
 - les clients.



Christelle CHAUDET

57

Introduction à l'approche orientée objets

- Pour chaque objet que vous avez sélectionné, demandez-vous ce que vous devez connaître de lui.
- Exemple :
 - Projet de garage :
 - la date de la dernière réparation d'une voiture,
 - les types de réparations effectuées, ses accidents ...
 - Projet de parking :
 - taille des voitures.

Christelle CHAUDET

59

Introduction à l'approche orientée objets

- Stratégie pour établir la liste d'objets :
 - Ecrire tous les objets auxquels vous pensez (en rapport avec votre projet)
 - Supprimer ceux faisant double emploi
 - Quels objets
 - ont des fonctionnalités qui vont au-delà de vos besoins ?
 - ne vous sont pas nécessaires ?
 - Supprimez-les de votre liste
- Exemple :
 - Projet concernant la construction de voiture : objets en rapport avec toutes les pièces d'une voiture.
 - Projet concernant le contrôle du trafic d'un garage : un objet voiture ne contenant qu'une description du type, de la marque et de la couleur pourrait suffire

Christelle CHAUDET

58

Introduction à l'approche orientée objets

- Il vous faut définir l'usage de chaque objet et ce dont les objets ont besoin pour être indépendants
 - chaque objet doit être pourvu d'une méthode pour chacune des actions auxquelles vous le destinez
- Exemple : Que doit faire l'objet voiture ?
 - Méthode particulière pour sa création (nommée constructeur) permettant d'initialiser ses attributs
 - Méthodes permettant de
 - de démarrer,
 - d'avancer,
 - de s'arrêter,
 - ...

Christelle CHAUDET

60