



Les objets et les classes

Pierre-Alain Muller

ENSISA

pa.muller@uha.fr

03.89.33.69.65

Sommaire

- Les objets
- Les collaborations entre objets
- Les classes
- Les contrats de classes



Les objets

- Les objets du monde réel nous entourent, ils naissent, vivent et meurent
- Les objets informatiques définissent une représentation simplifiée des entités du monde réel
- Les objets représentent des entités concrètes (avec une masse) ou abstraites (concept)



Représentation graphique des objets

Un objet

Un autre objet

Encore un objet

Les objets sont des abstractions

- Une abstraction est un résumé, un condensé
- Mise en avant des caractéristiques essentielles
- Dissimulation des détails
- Une abstraction se définit par rapport à un point de vue



Exemples d'abstractions

- Une carte routière
- Un nombre complexe
- Un téléviseur
- Une transaction bancaire
- Une porte logique
- Une pile

Caractéristiques fondamentales des objets

- L'état
- Le comportement
- L'identité

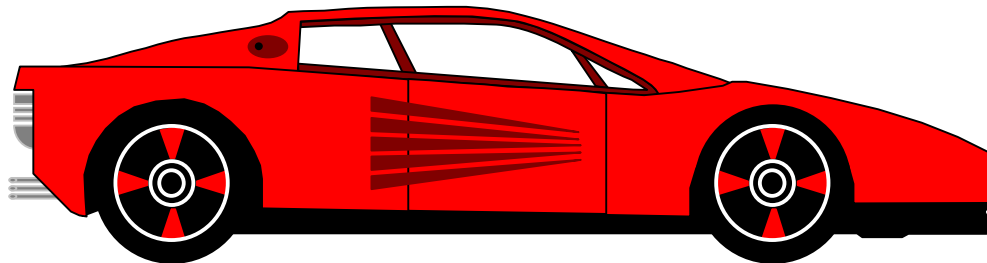


L'état

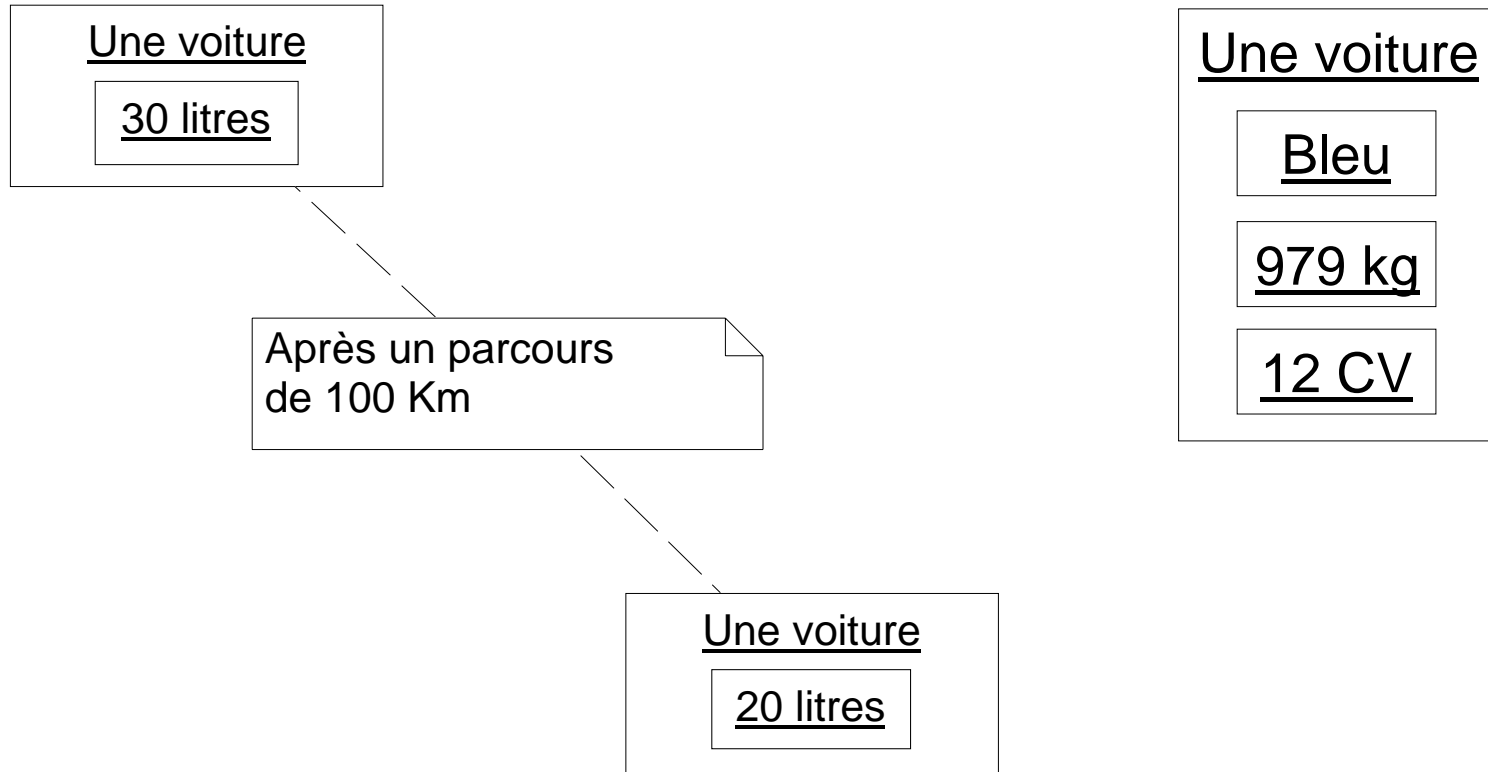
- L'état regroupe les valeurs instantanées de tous les attributs d'un objet
- L'état évolue au cours du temps
- L'état d'un objet à un instant donné est la conséquence de ses comportements passés

Exemples

- Pour un signal électrique : l'amplitude, la pulsation, la phase, ...
- Pour une voiture : la marque, la puissance, la couleur, le nombre de places assises, ...



Exemple

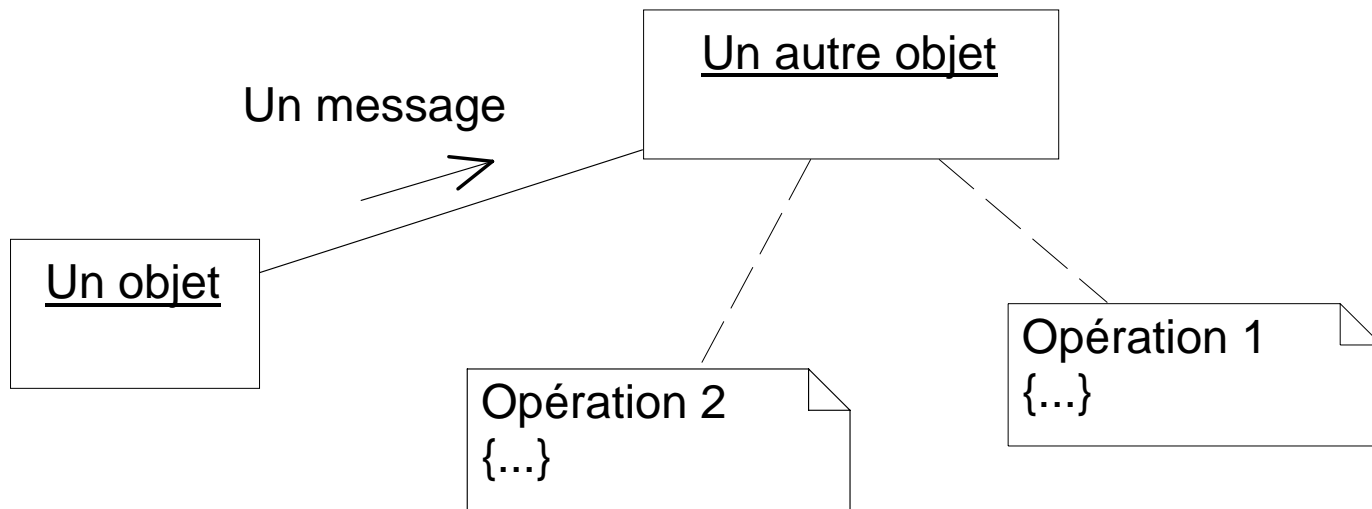


Le comportement

- Le comportement décrit les actions et les réactions d'un objet
- Le comportement regroupe toutes les compétences d'un objet
- Le comportement se représente sous la forme d'opérations

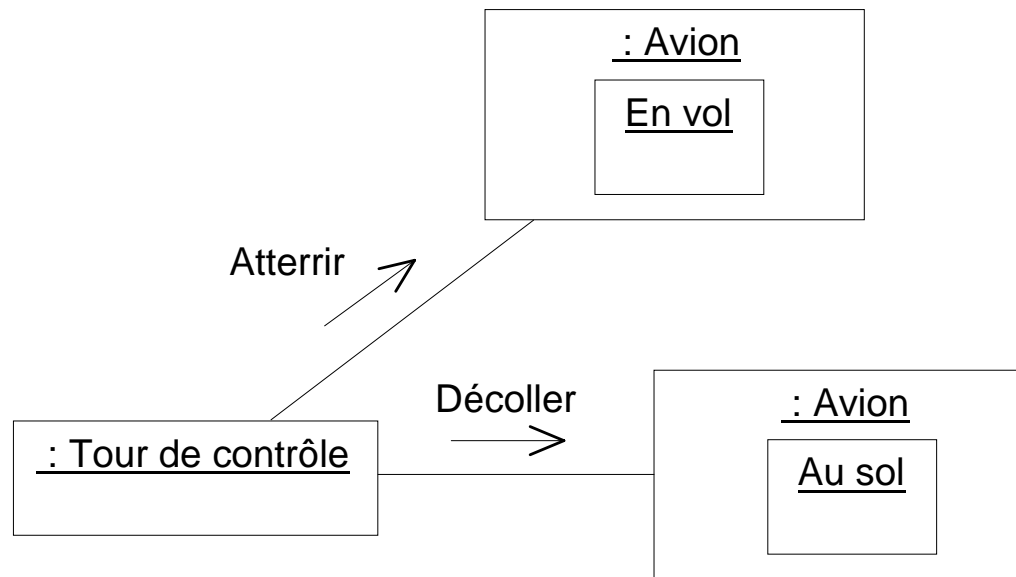
Comportement (suite)

- Un objet peut faire appel aux compétences d'un autre objet



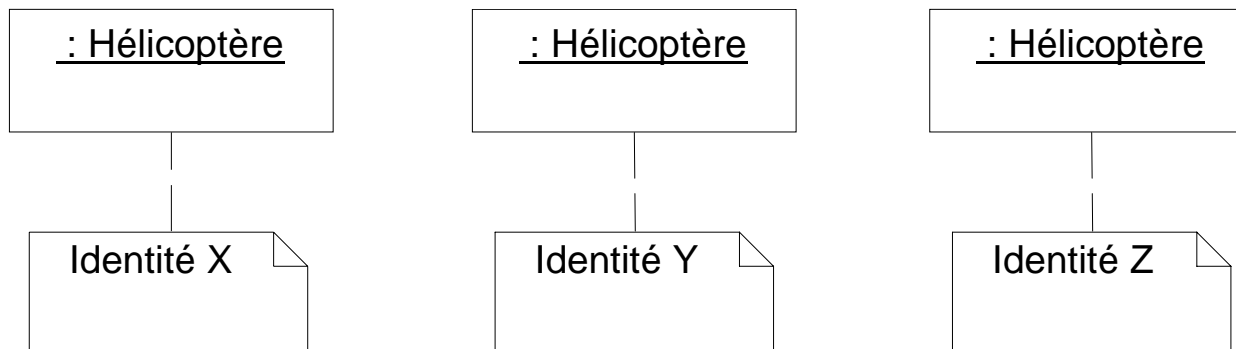
Comportement (suite)

- L'état et le comportement sont liés
 - Le comportement dépend de l'état
 - L'état est modifié par le comportement



L'identité

- Tout objet possède une identité qui lui est propre et qui le caractérise
- L'identité permet de distinguer tout objet de façon non ambiguë, indépendamment de l'état

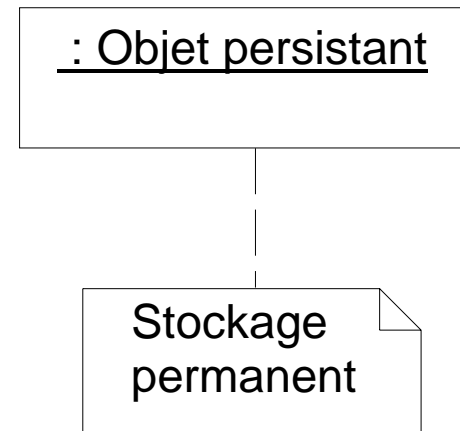


Identité (suite)

- Les langages objets utilisent généralement des pointeurs pour réaliser un identifiant
- Une clé primaire dans une base de donnée relationnelle est une manière de réaliser l'identité (en l'insérant dans l'état)
- Exemple : notre numéro de sécurité sociale

La persistance

- L'existence d'un objet transcende le temps et l'espace
- Sauvegarde de l'état et de la classe d'un objet
 - Passivation / Activation
 - Objets transitoires
 - Objets persistants

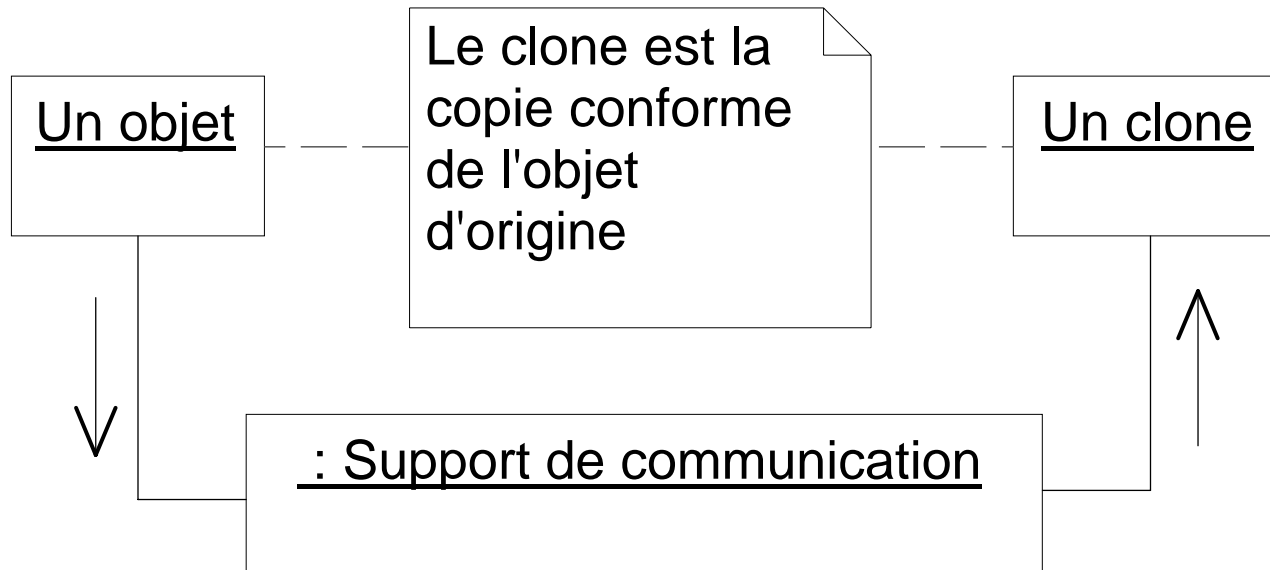


Persistance (suite)

- Pas de support de la part des langages de programmation
- Bases de données objet
- Couche objet sur une base de données relationnelle

Transmission des objets

- Problématique proche de la persistance



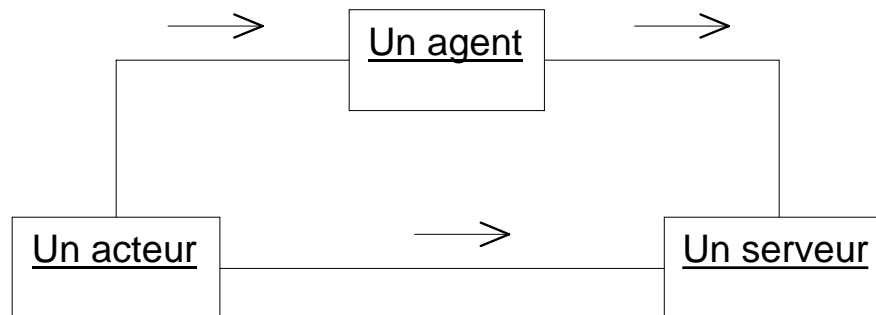
Communication entre objets

- Application = société d'objets collaborants
- Les objets travaillent en synergie afin de réaliser les fonctions de l'application
- Le comportement global d'une application repose donc sur la communication entre les objets qui la composent



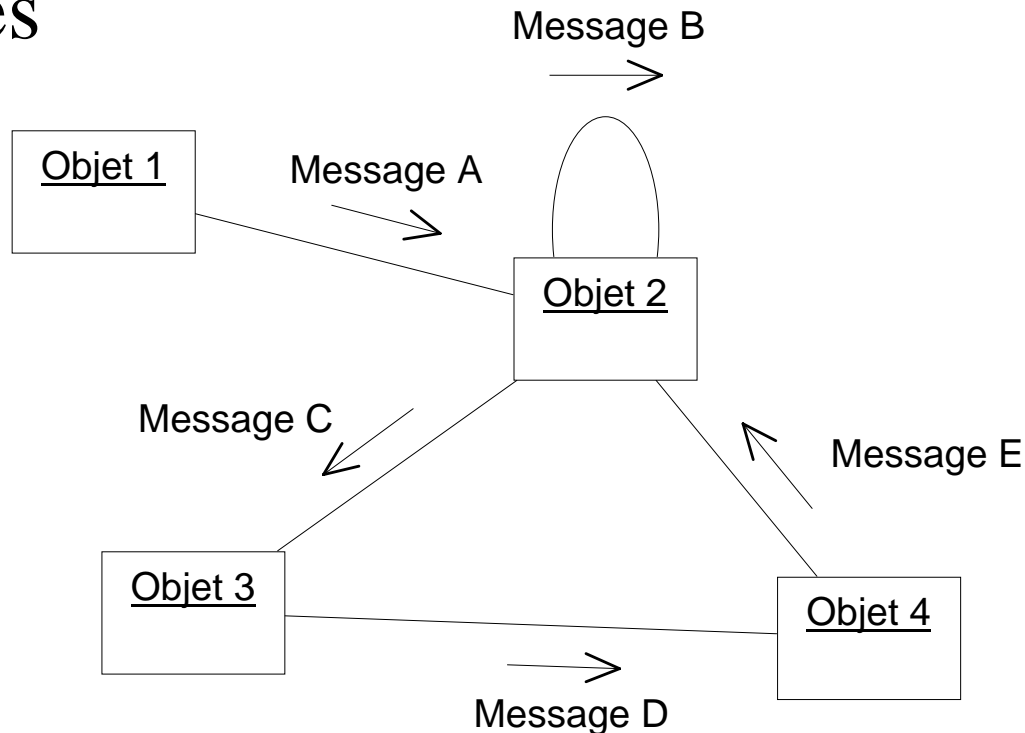
Catégories de comportement

- Un *acteur* est un objet qui est toujours à l'origine d'une inter-action
- Un *serveur* est un objet qui n'est jamais à l'origine d'une inter-action
- Un *agent* est à la fois un acteur et un serveur



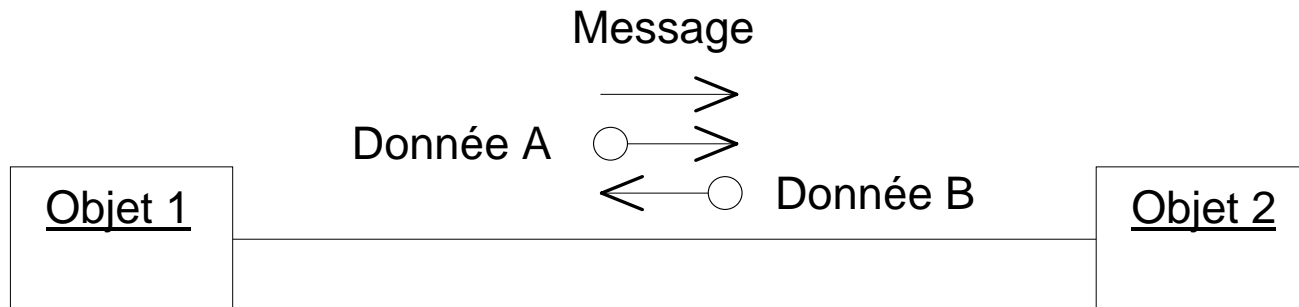
Les objets ne vivent pas en ermites

- Les objets interagissent les uns avec les autres



Communication entre objets

- Les objets communiquent en échangeant des messages



Le concept de messages

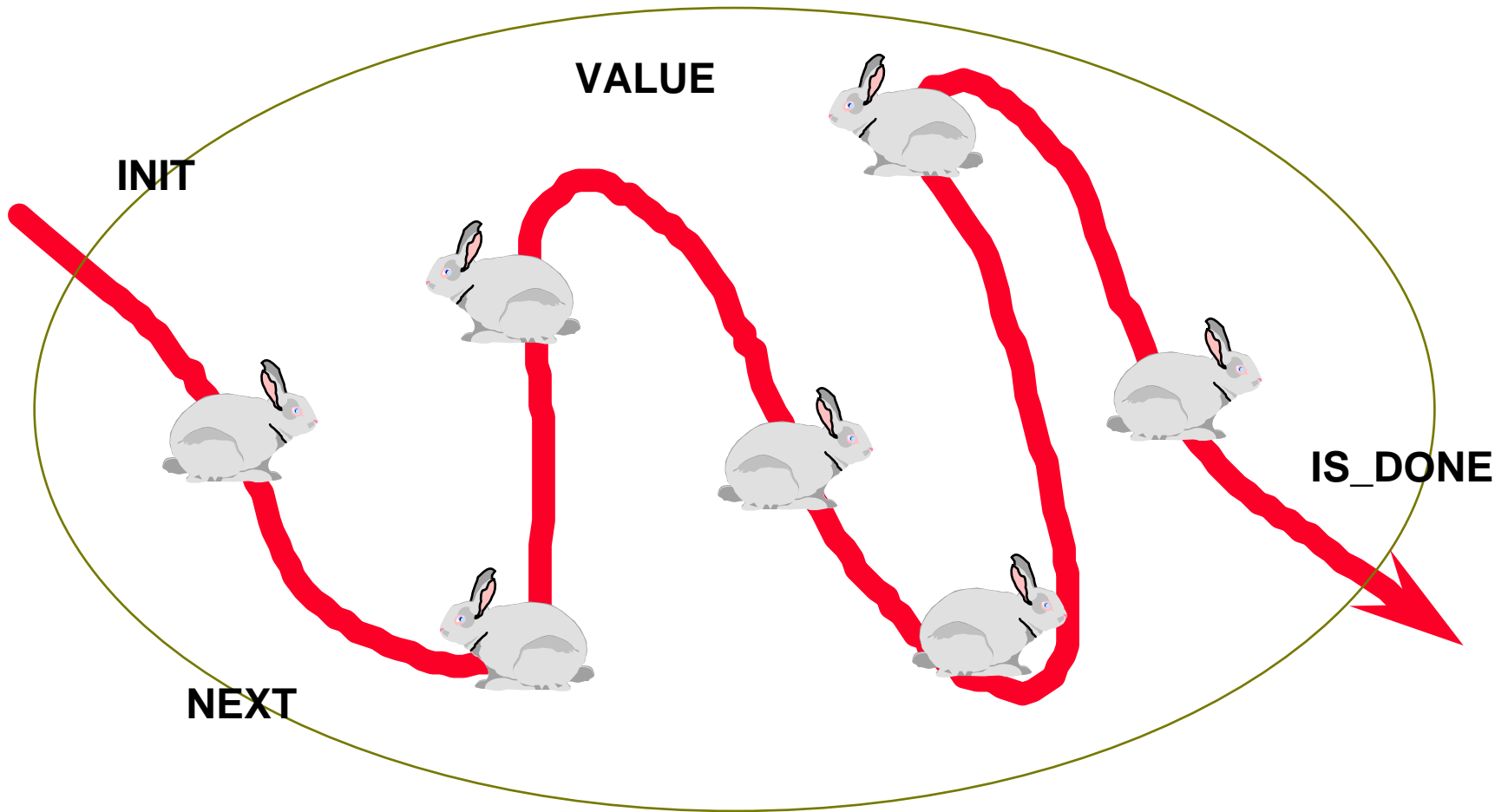
- L'unité de communication entre objets
- Concept très général pouvant être mis en oeuvre suivant de nombreuses variantes
- Regroupe les flots de contrôle et les flots de données
- Représente également les événements

Catégories de messages

- Les **constructeurs** qui créent des objets
- Les **destructeurs** qui détruisent des objets
- Les **sélecteurs** qui renvoient tout ou partie de l'état
- Les **modifieurs** qui changent tout ou partie de l'état
- Les **itérateurs** qui traversent une collection d'objets



Exemple d'itérateur actif



Le chaos des objets

- Le monde qui nous entoure est constitué de très nombreux objets
- Pour comprendre le monde, l'être humain a tendance à regrouper les éléments qui se ressemblent
- Regrouper des objets suivants des critères de ressemblance s'appelle classer
- Les humains ont classé les animaux, les plantes, les champignons, les atomes, ...

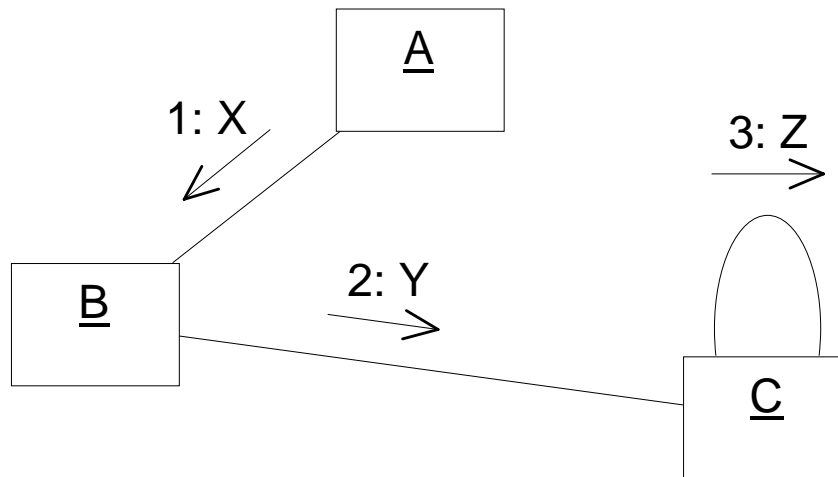


Les diagrammes de collaboration

- Des objets dans une situation donnée
- Des liens relient les objets qui se connaissent
- Les messages échangés par les objets sont représentés le long de ces liens
- L'ordre d'envoi des messages est matérialisé par un numéro de séquence

Exemple

- *Un objet **A** envoie un message **X** à un objet **B**, puis l'objet **B** envoie un message **Y** à un objet **C**, et enfin **C** s'envoie un message **Z**.*

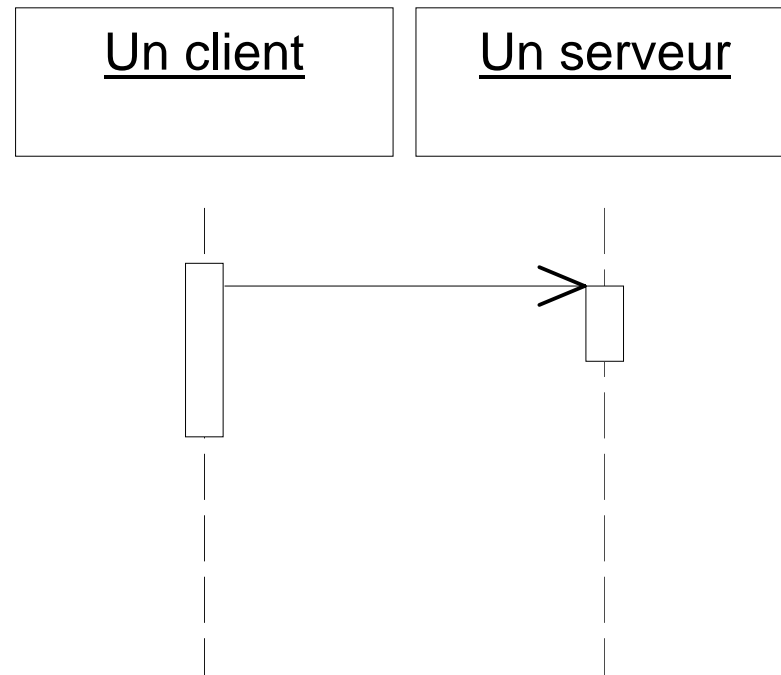


Les diagrammes de séquence

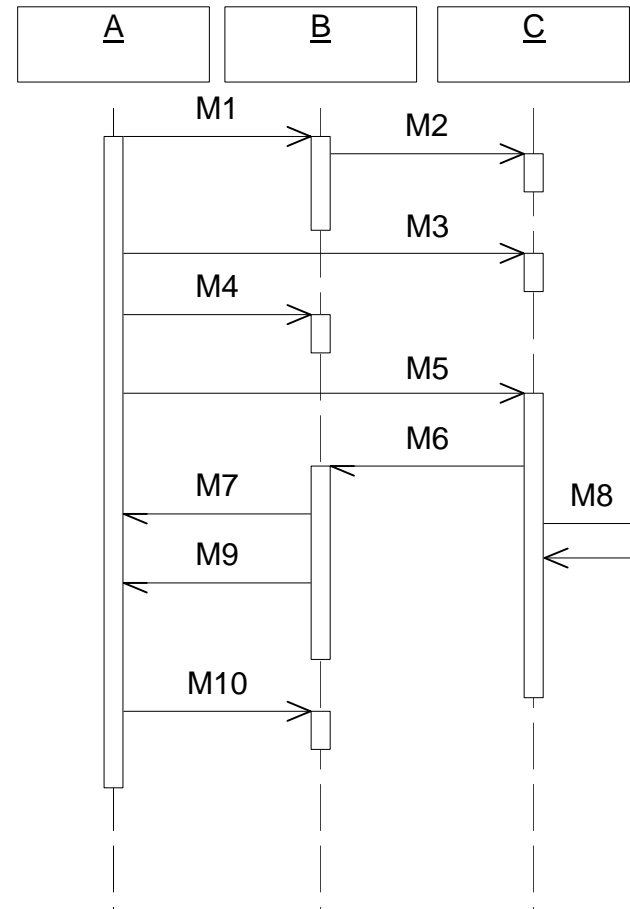
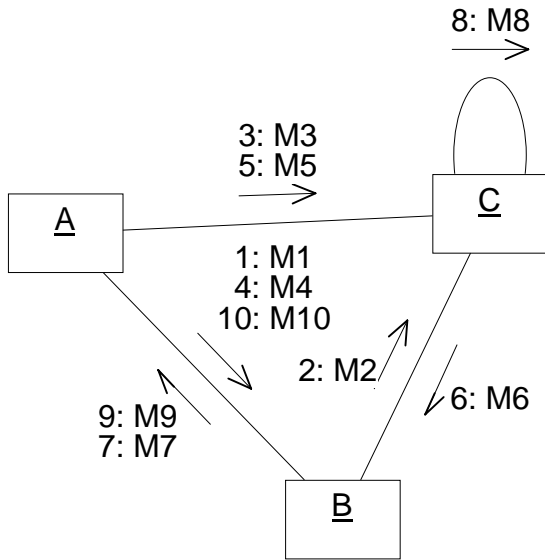
- L'accent est mis sur la communication, au détriment de la structure spatiale
- Chaque objet est représenté par une barre verticale
- Le temps s'écoule de haut en bas, de sorte que la numérotation des messages est optionnelle.



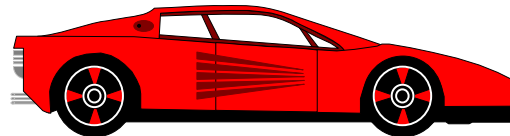
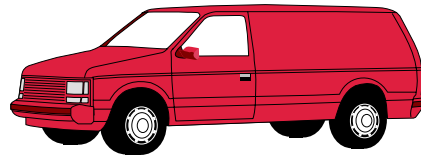
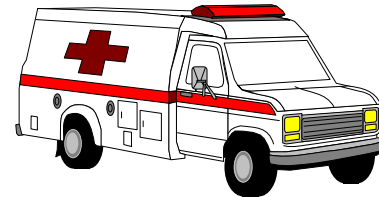
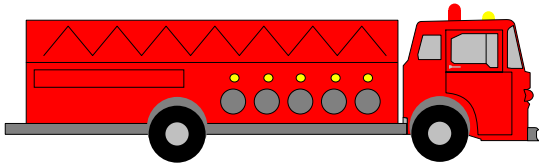
Exemple



Comparison



Le chaos des objets (suite)



Les classes

- La classe est une description abstraite d'un ensemble d'objets
- La classe peut être vue comme la factorisation des éléments communs à un ensemble d'objets
- La classe décrit le domaine de définition d'un ensemble d'objets

Représentation graphique des classes

Nom de classe
Attributs
Opérations()

Motocyclette
Couleur Cylindrée Vitesse maximale
Démarrer() Accélérer() Freiner()

Nombre complexe
Additionner() Soustraire() Multiplier() Diviser() Prendre le module() Prendre l'argument() Prendre la partie réelle() Prendre la partie imaginaire()

Téléviseur
Allumer() Eteindre() Changer de programme() Régler le volume()

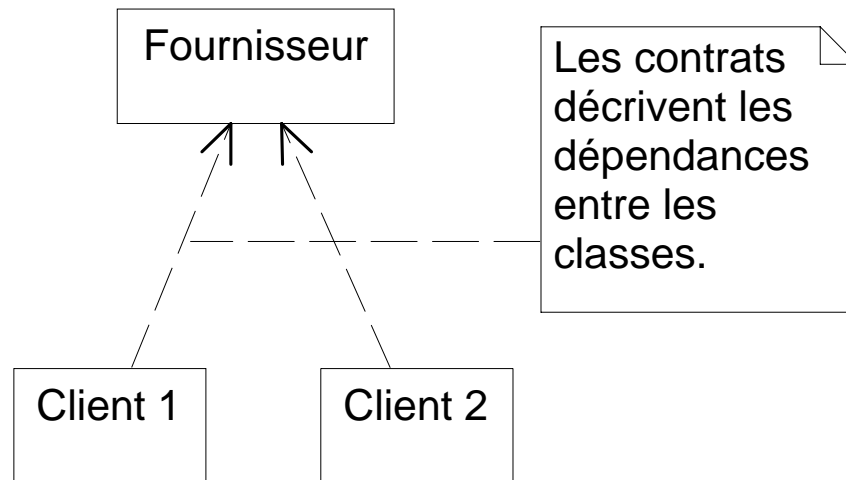


Description des classes

- Séparée en deux parties
 - La *spécification* d'une classe qui décrit le domaine de définition et les propriétés des instances de cette classe (type de donnée)
 - La *réalisation* qui décrit comment la spécification est réalisée

La notion de contrat

- Une classe s'engage à fournir les services publiés dans sa spécification



L'encapsulation

- Chaque objet encapsule des données dans son état
- Le genre des données encapsulées et les opérations applicables à un objet sont décrites dans la classe
- Les classes permettent de décrire des contrats qui seront valables pour tous les objets issus de ces classes



Encapsulation (Suite)

- La spécification d'une classe définit la partie visible des objets, le reste est caché dans la réalisation

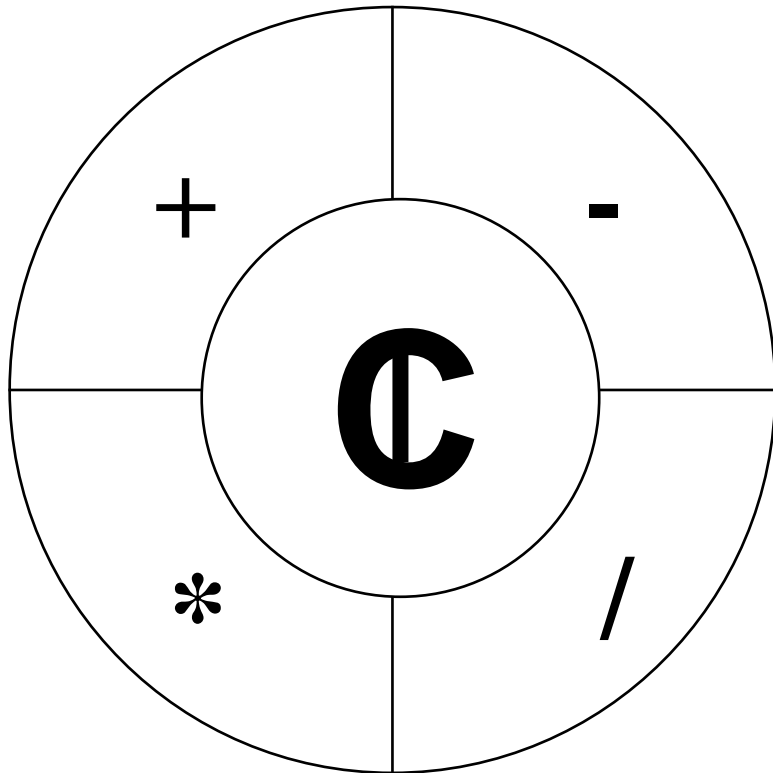
Règles de visibilité
+ Attribut public # Attribut protégé - Attribut privé
+ Opération publique() # Opération protégée() - Opération privée()

Bénéfices de l'encapsulation

- L'encapsulation présente deux avantages
 - Les données encapsulées sont protégées des accès intempestifs, ce qui permet de garantir leur intégrité
 - Les clients d'une abstraction ne dépendent pas de la réalisation de l'abstraction, mais seulement de sa spécification



Exemple d'encapsulation



Complexe
- Module - Argument
+ Addition() + Soustraction() + Multiplication() + Division()

Complexe
- Partie réelle - Partie imaginaire
+ Addition() + Soustraction() + Multiplication() + Division()

Conclusion

- Les objets naissent, vivent et meurent
- Les objets interagissent entre eux
- Les objets sont regroupés dans des classes qui les décrivent de manière abstraite
- La classe intègre les concepts de type, de module et d'encapsulation

