



# **Introduction au monde des objets : les concepts**

**Pierre Lefebvre**  
(adapté par Denis Bureau)

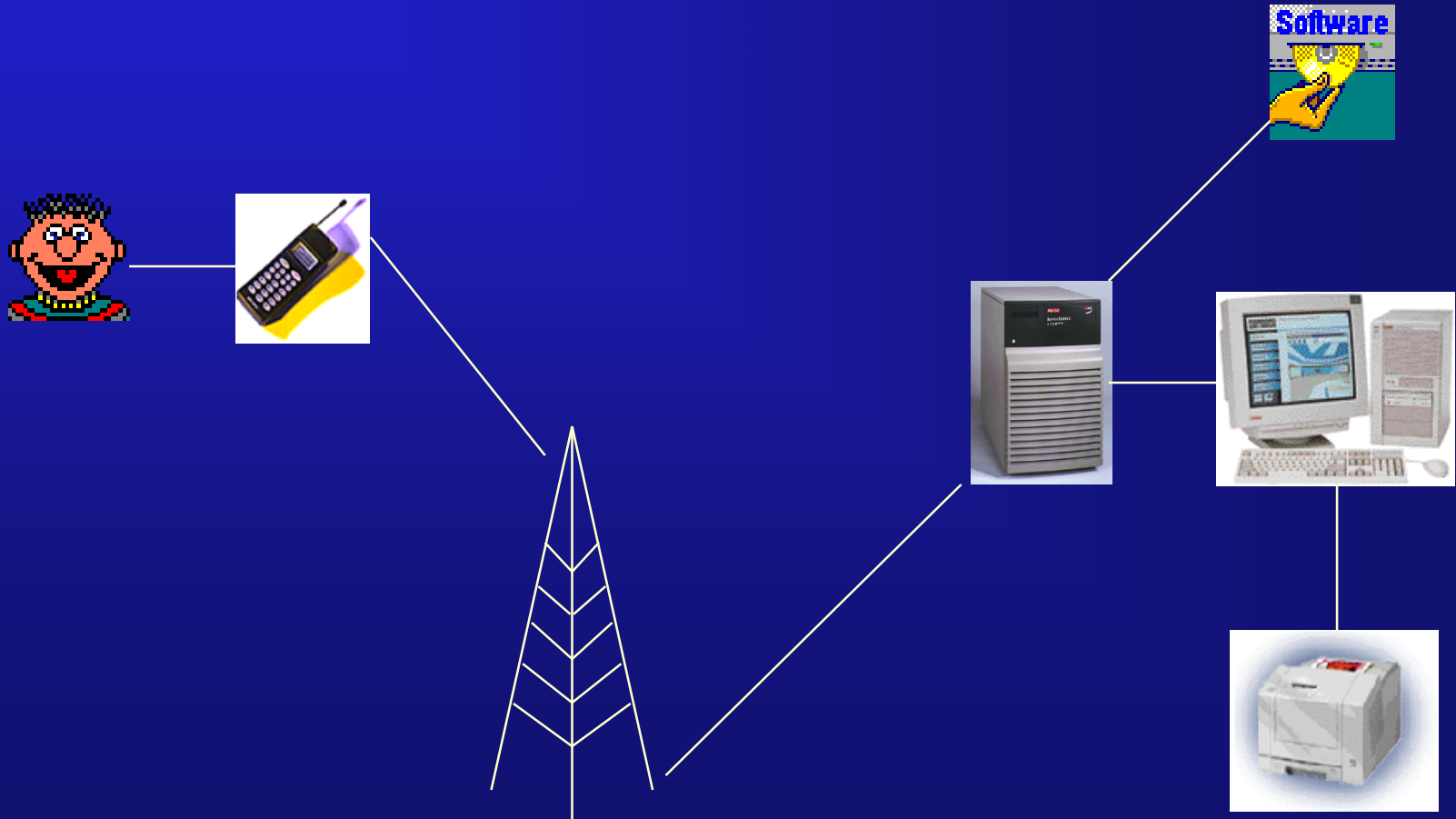
# Principes

- **L'approche orientée objet est un paradigme différent pour la conception et la programmation du logiciel**
- **L'approche orientée objet est basée sur la modélisation des objets du monde réel**
- **L'approche orientée objet crée des programmes réutilisables et évolutifs**

# Objets de la vie courante



# Les objets coopèrent



# Objet : boîte noire

- **Services rendus par l'objet :**

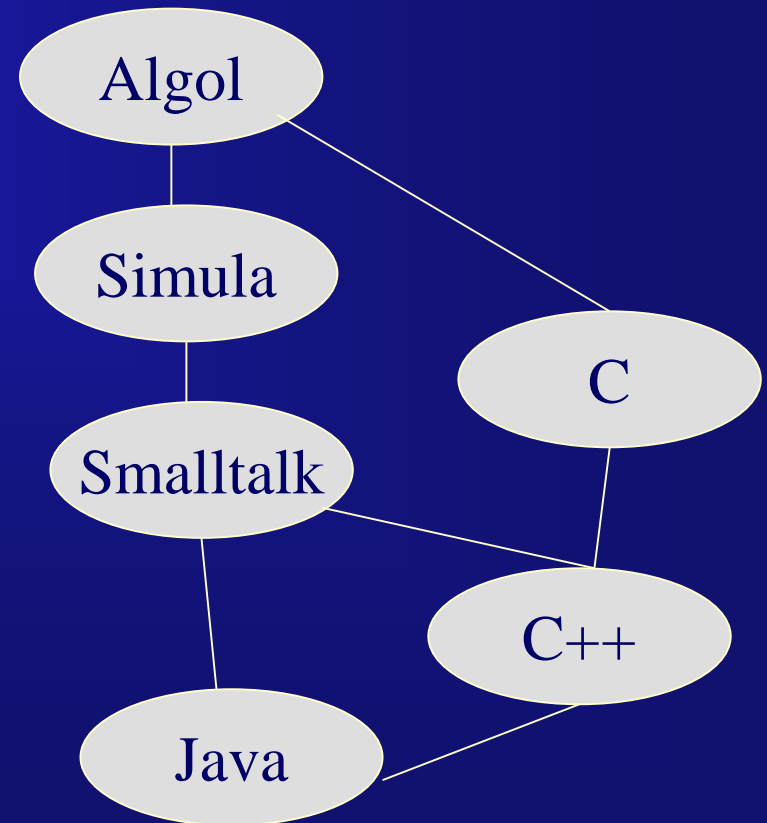
- Démarrer ;
- Arrêter ;
- Accélérer ;
- Freiner ;
- Climatiser ...

- **Fonctionnement interne ???**



# Origines du concept objet

- Deux origines prépondérantes :
  - La simulation
  - Les interfaces graphiques



# Limites des approches traditionnelles

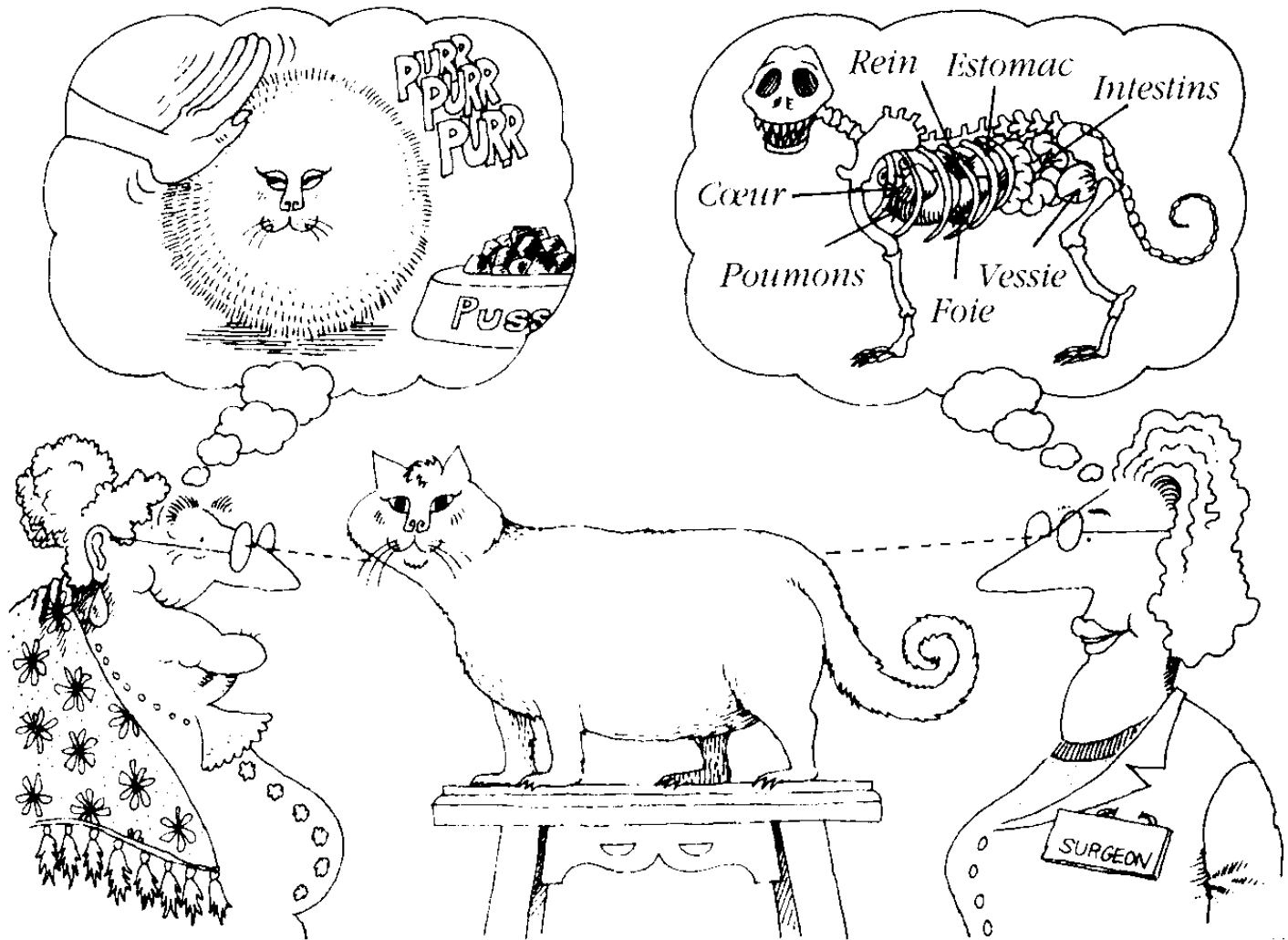
- **Taille et complexité croissantes des logiciels**
- **Manque de localisation**
  - éparpillement à travers les programmes des différents aspects concernant une même entité
- **Faible réutilisabilité**
- **Difficulté d'évolution**
- **Logique étrangère à la réalisation des interfaces utilisateur**
- **Conclusion :**
  - Le bon sens réclame la meilleure adéquation possible entre le modèle informatique et la réalité

# Abstraction

- L'abstraction est la principale technique pour comprendre le monde qui nous entoure
- C'est un processus mental qui simplifie la complexité des entités du monde réel en identifiant et retenant l'essentiel d'une entité
- L'abstraction se concentre sur les caractéristiques essentielles d'un objet, selon le point de vue de l'observateur



# Abstraction

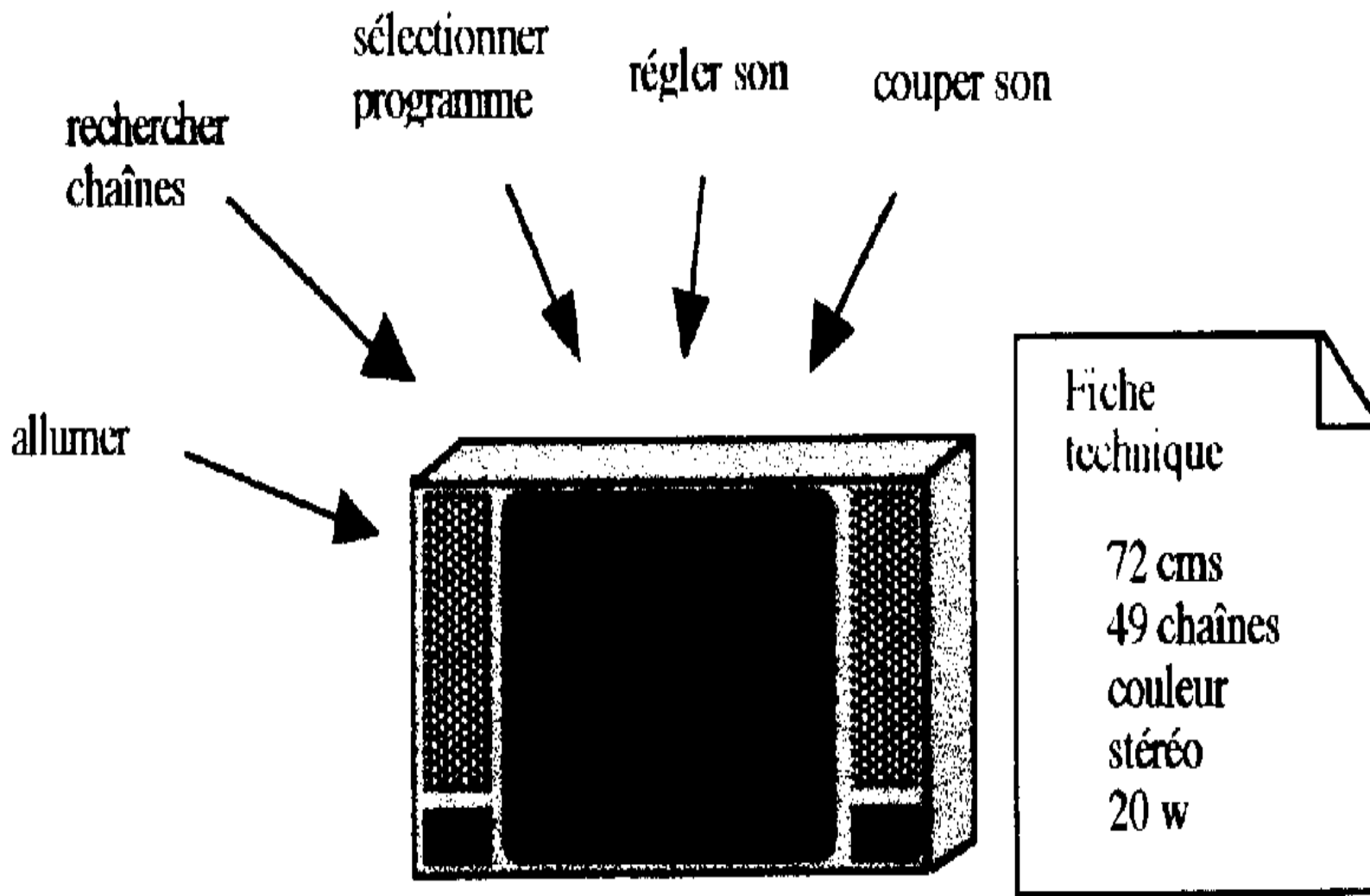


L'abstraction se concentre sur les caractéristiques essentielles d'un objet, selon le point de l'observateur

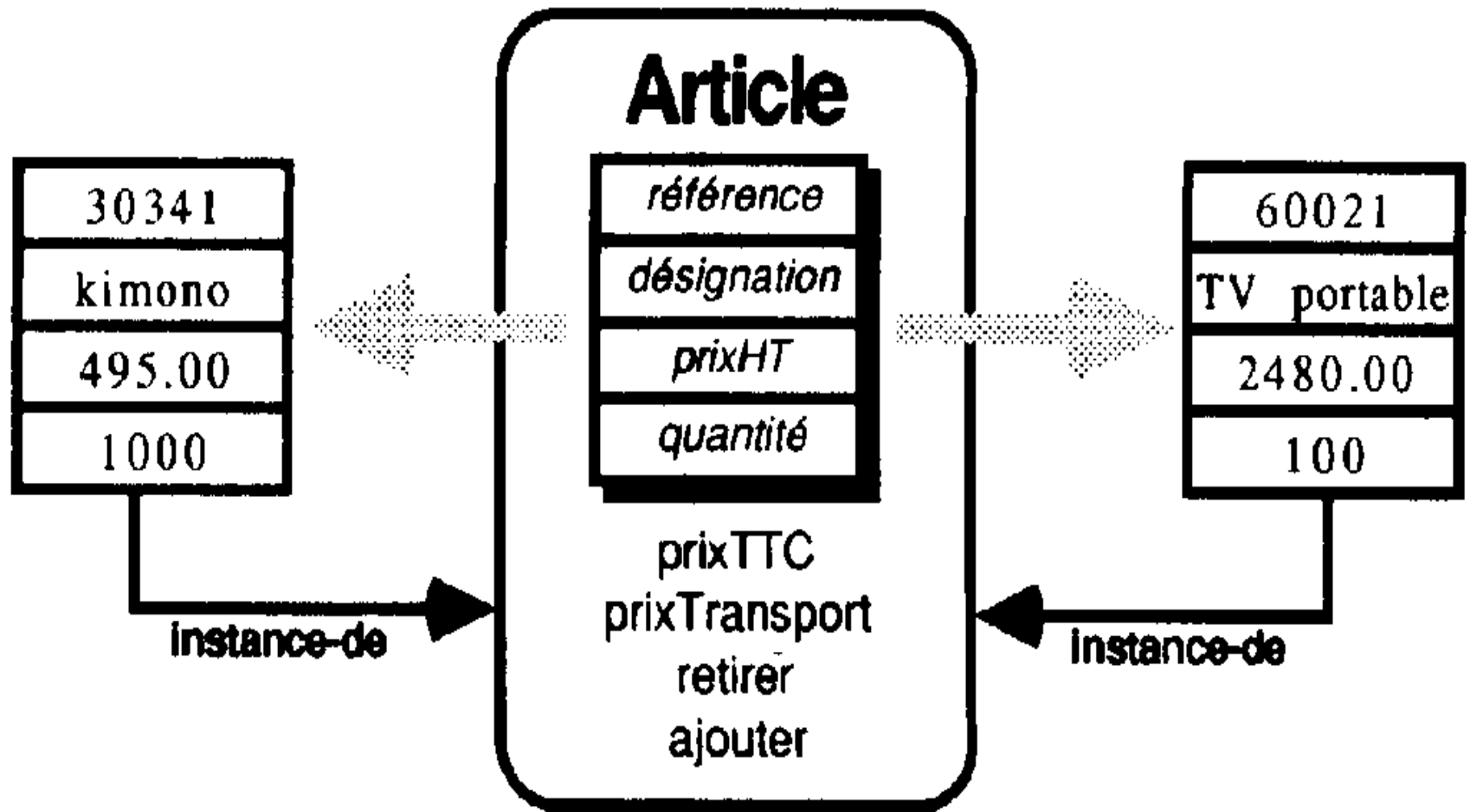
# Le concept d'objet

- Un objet est une abstraction permettant de modéliser des éléments du monde réel
- Un objet n'existe pas d'une façon isolée. Il doit fournir un ensemble de services ou demander un ensemble de services
- Les services que l'objet fournit sont le miroir de l'entité modélisée
  - Ex: une banque propose des services tels que le retrait d'argent, le dépôt d'argent, etc.
- Les objets pertinents pour le système sont ceux que l'utilisateur identifie naturellement
- Les méthodes de l'objet sont déclenchées par des messages provenant des autres objets de l'application
- Conclusion : Observer, c'est modéliser

# Exemples d'objets

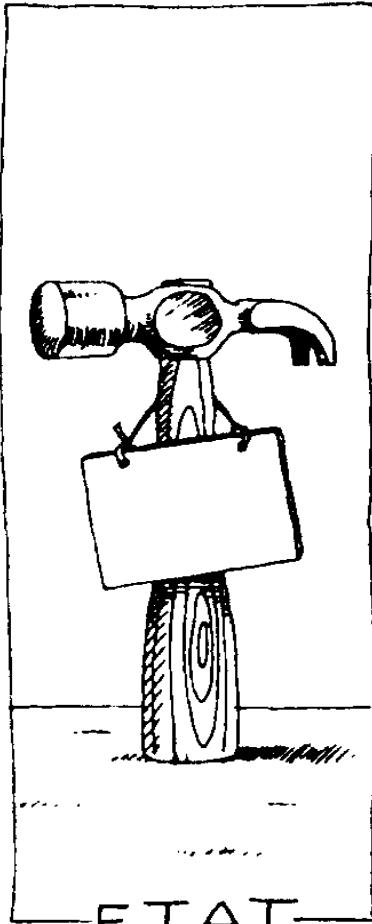


# Exemples d'objets

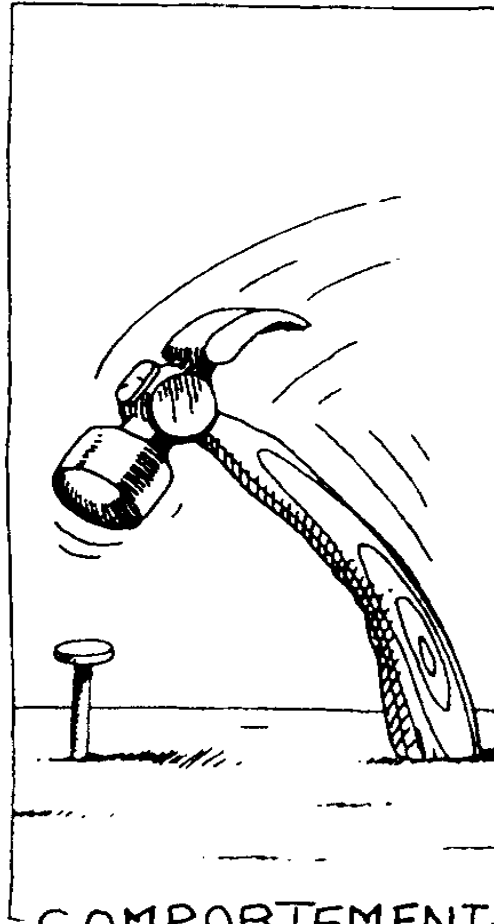


Instanciation par moulage de la classe **Article**.

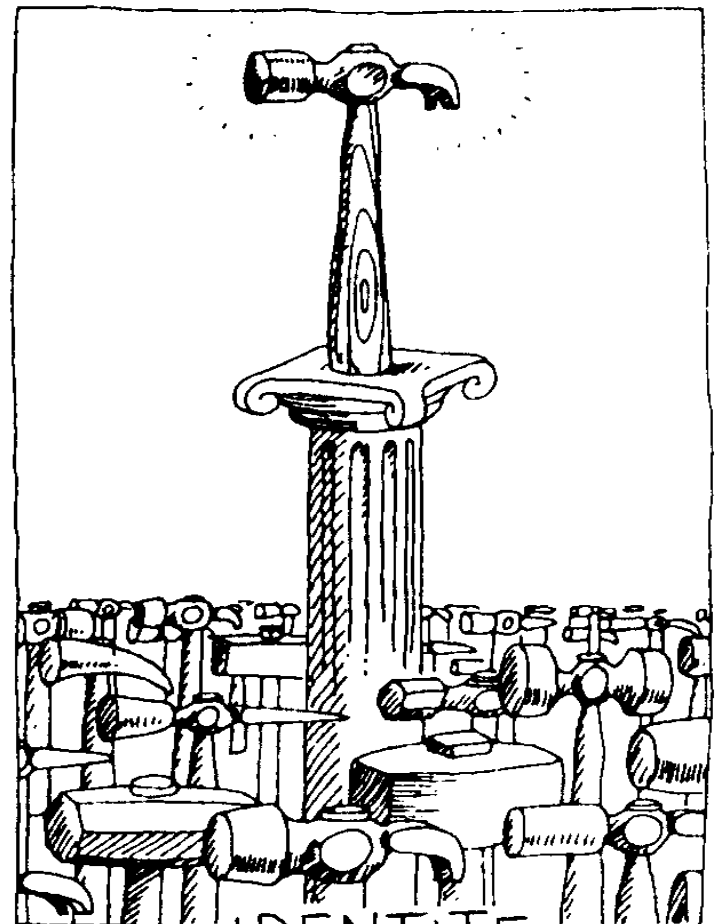
# Caractéristiques d'un objet



ETAT



COMPORTEMENT



IDENTITE

Un objet a un état, il exhibe un comportement bien défini, et il a une identité unique

# Etat d'un objet

- **Attribut**

- Information qui qualifie l'objet qui le contient
- Peut être une constante

- **État**

- Valeurs instantanées de tous les attributs d'un objet
- Évolue au cours du temps

attribut  
variable

attribut  
constant

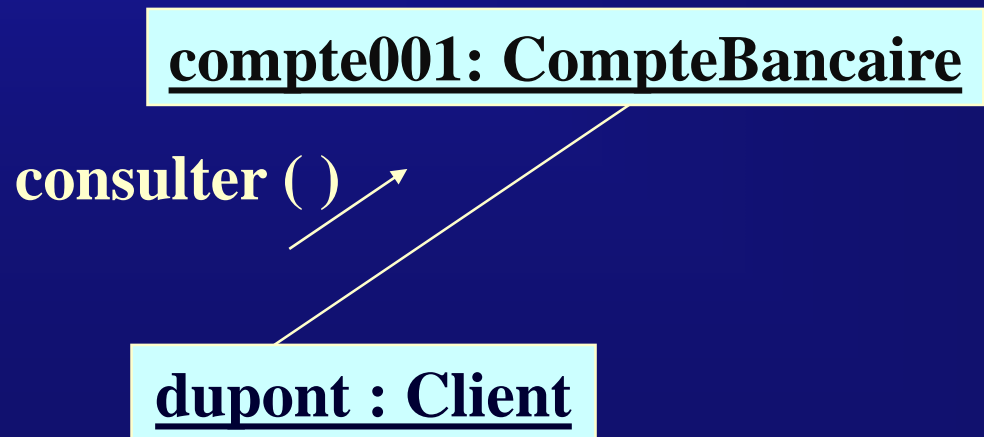
Identité  
d'objet

compte001 : CompteBancaire

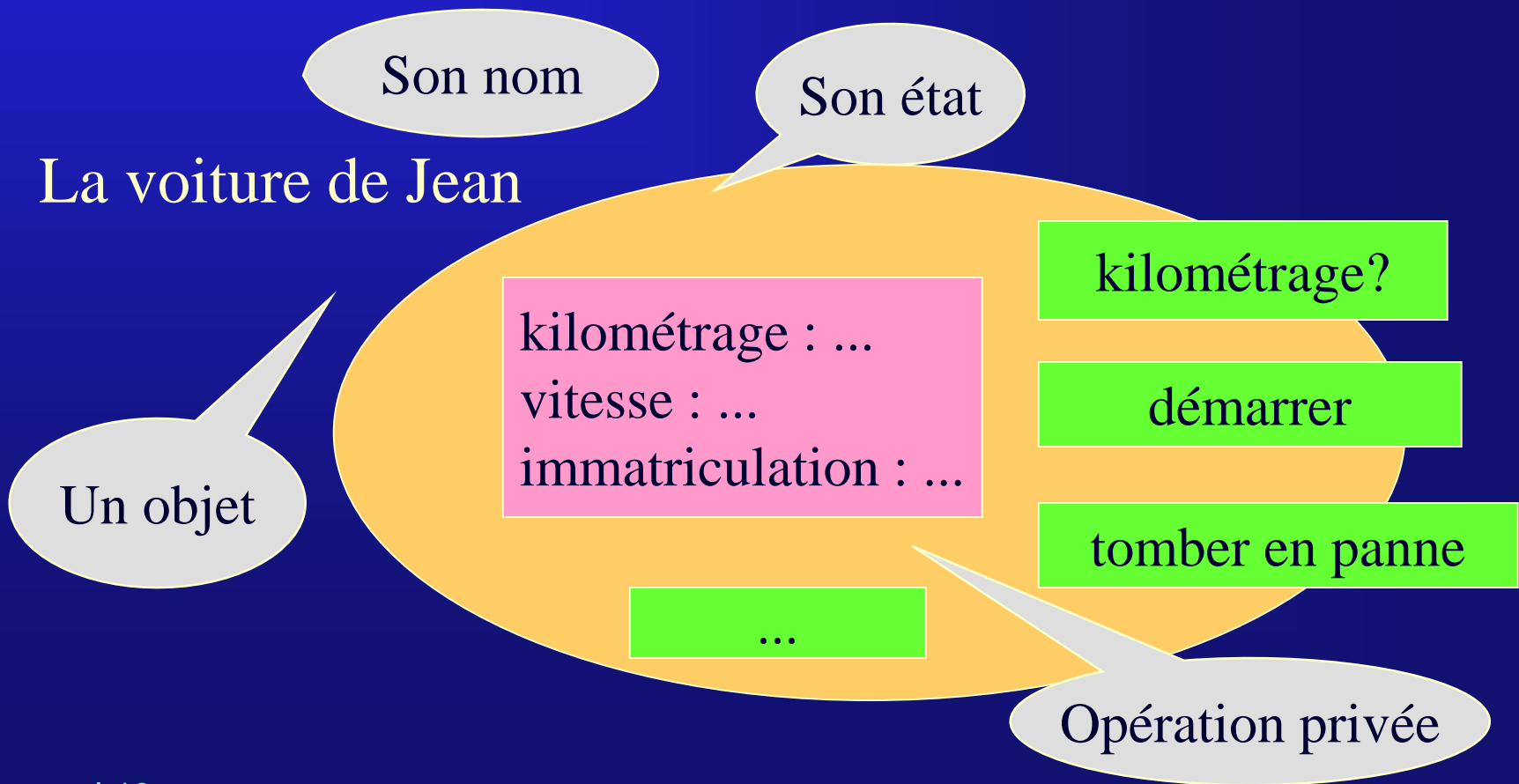
- solde
- **DEBITAUTORISE**

# Comportement d'un objet

- **Décrit les actions et les réactions d'un objet**
  - Compétences d'un objet
- **Service = opération = méthode**
  - Comportement déclenché par un message



# Représentation d'un objet

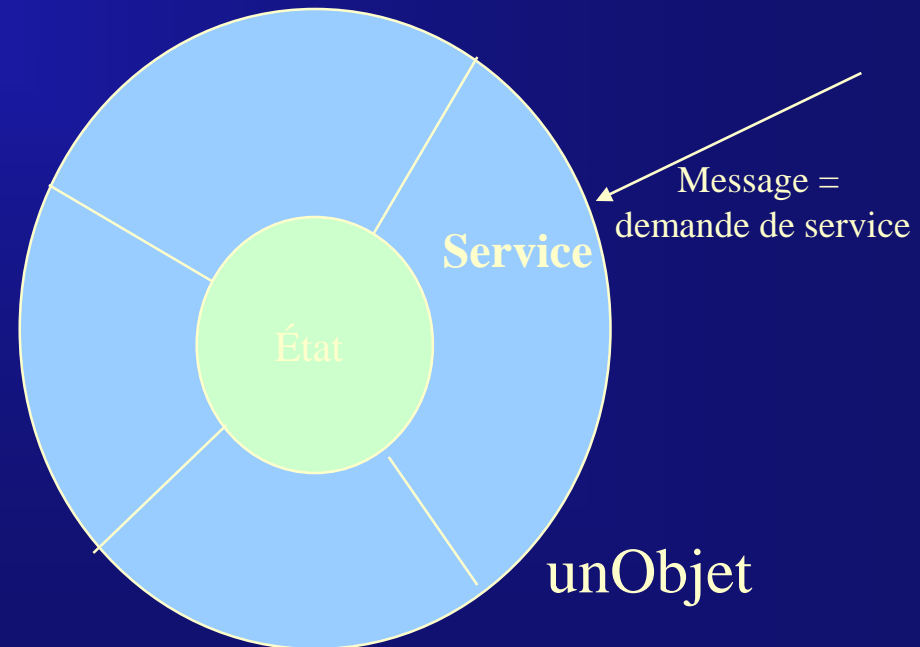




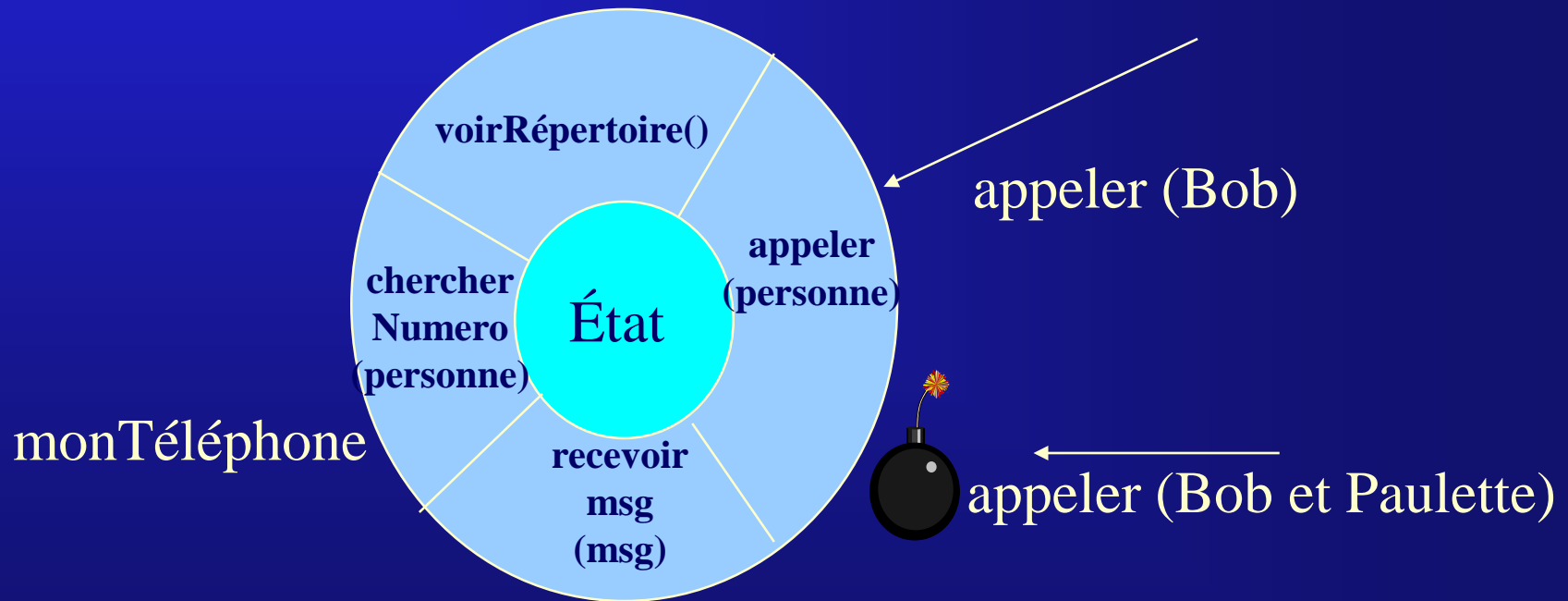
# Objet logiciel

- **Abstraction**
  - Représentation abstraite des entités du monde réel ou virtuel dans le but de les piloter ou de les simuler
- **Programme, logiciel**

**Objet =**  
**État +**  
**Comportement +**  
**Identité**



# Exemple : un téléphone portable

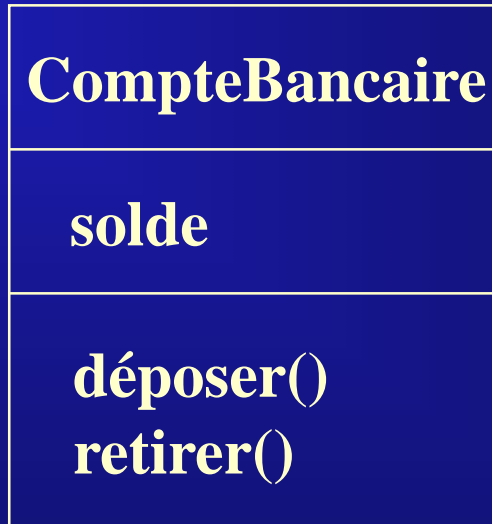


# Classe

- La classe est le moule permettant d'engendrer des objets
- La classe est un patron qui définit les attributs et les méthodes d'une collection d'objets
- Elle représente une collection d'objets avec des attributs similaires mais des valeurs distinctes
- Quand un objet reçoit un message, il regarde la définition de la classe et applique la méthode à ses propres attributs
- Le mécanisme par lequel l'exécution d'un programme produit un objet à partir d'une classe est l'instanciation

# Classe CompteBancaire

**CompteBancaire**

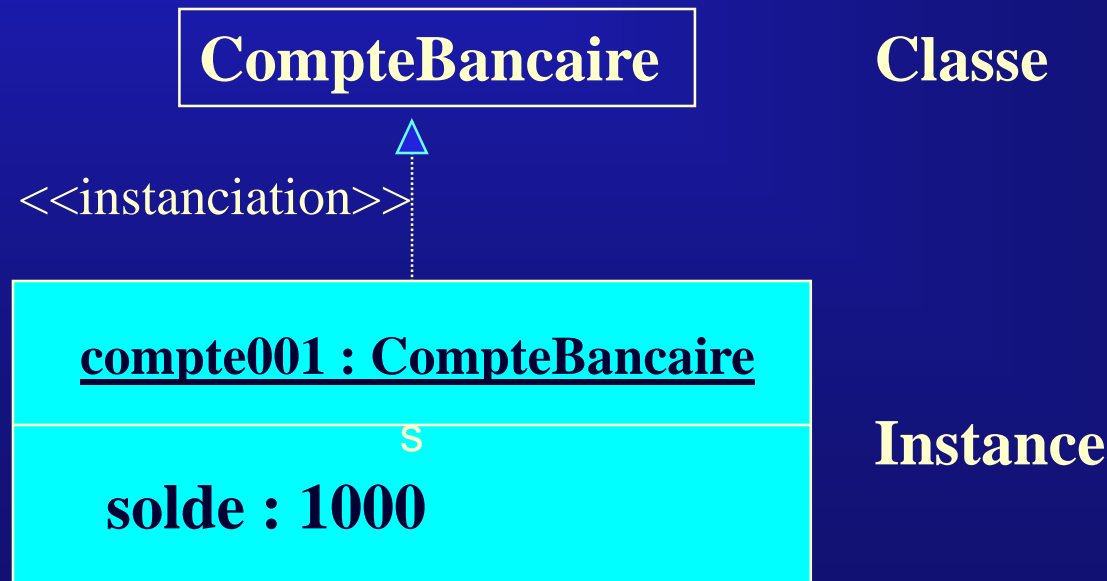


**attributs**

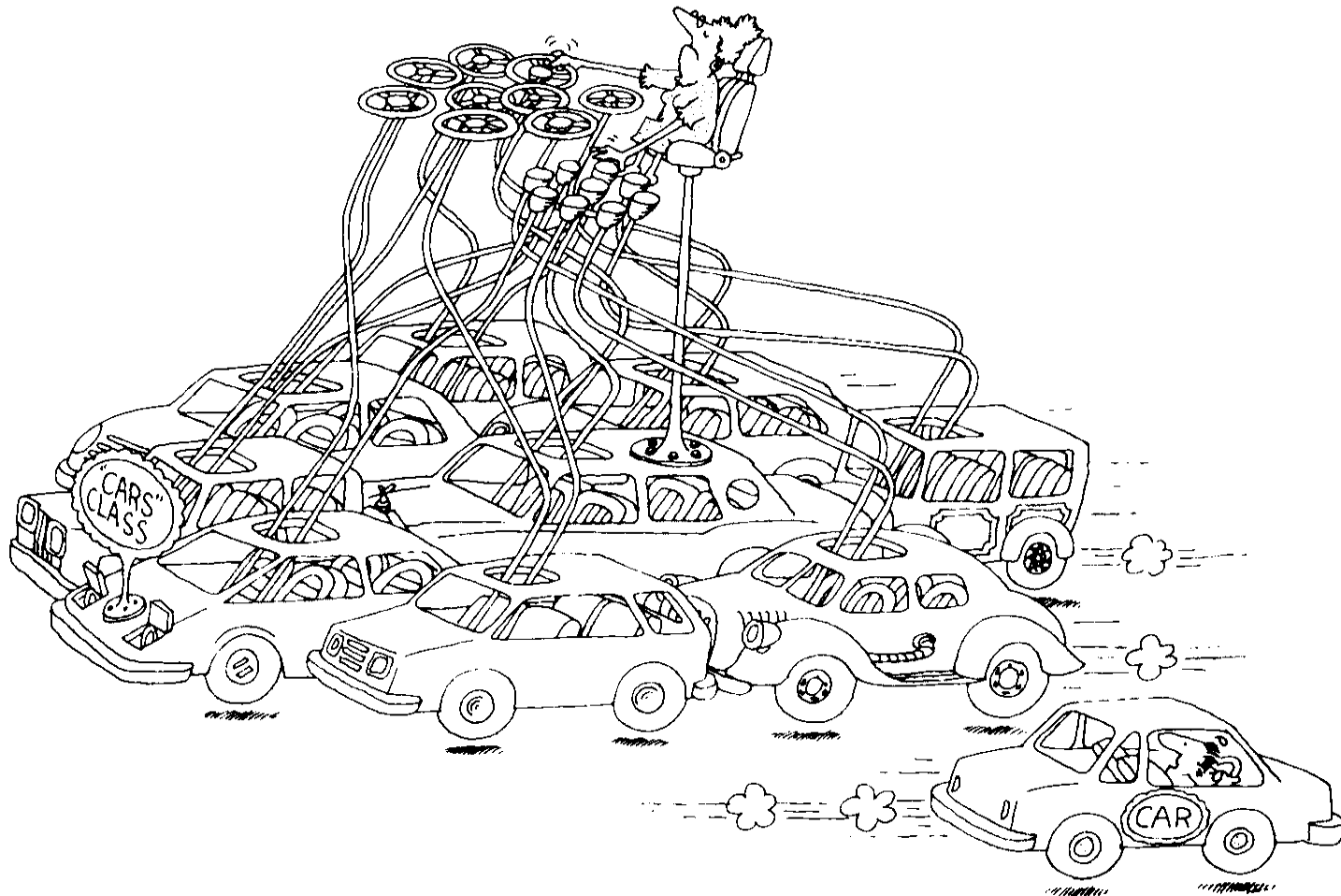
**opérations**

# Instance

- Chaque objet appartient à une classe
- Relation d'instanciation « instance de »



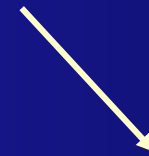
# Concept de classe



Une classe représente un ensemble d'objets qui partagent une structure commune et un comportement commun

# Définir les classes, un art difficile

**Univers du problème  $\neq$  Univers réel**

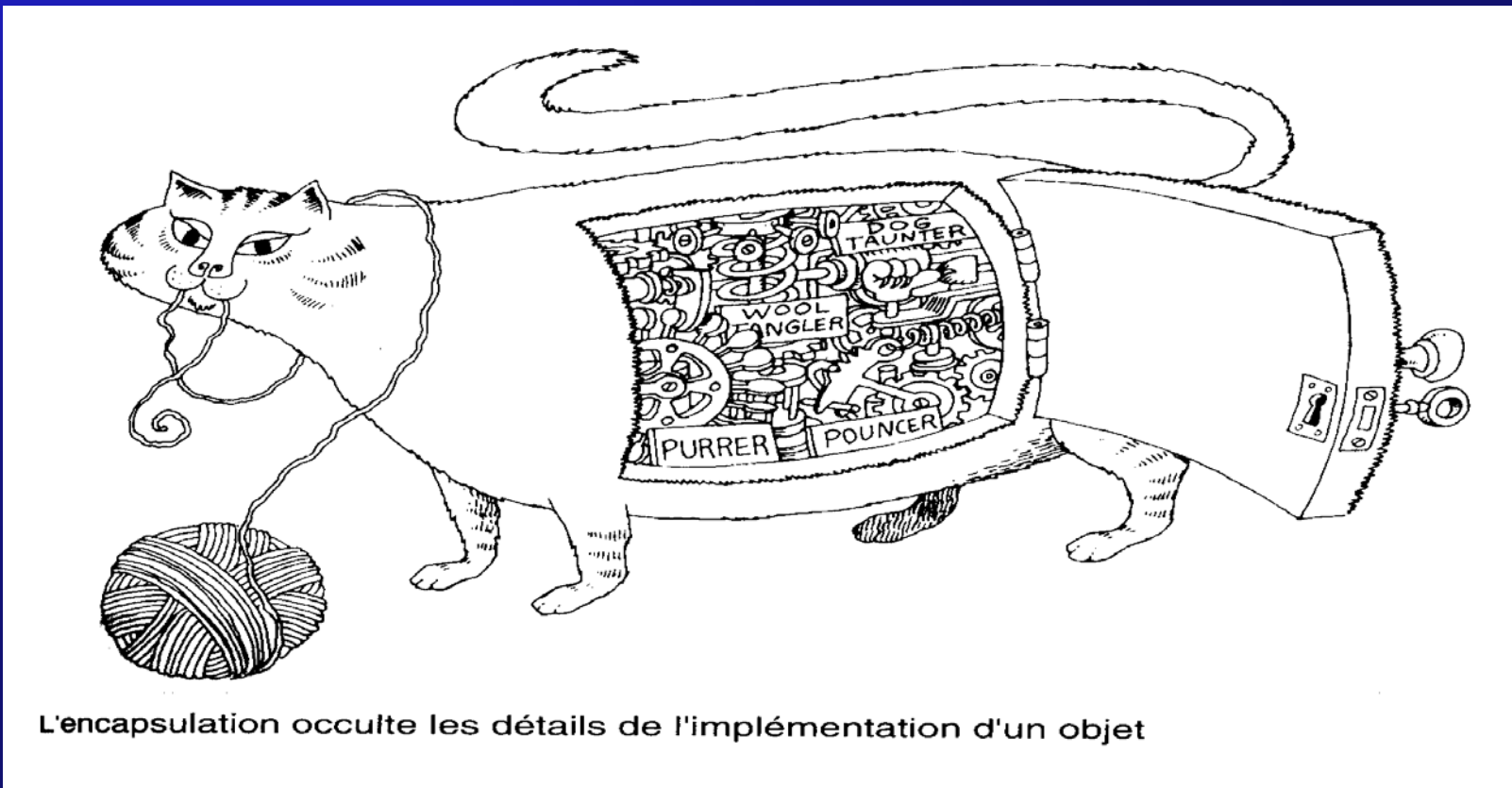


- Trouver le modèle le plus efficace pour traiter le problème
- Délimité

- Modèle perceptif dépendant de nombreux facteurs (psychologiques, culturels, ..)
- Extrêmement vaste

# Encapsulation

- L'encapsulation est le mécanisme par lequel le développeur cache une partie de l'information pour préserver l'intégrité de l'objet.  
Ce faisant, il établit une séparation entre une interface (publique) et une implémentation (invisible)

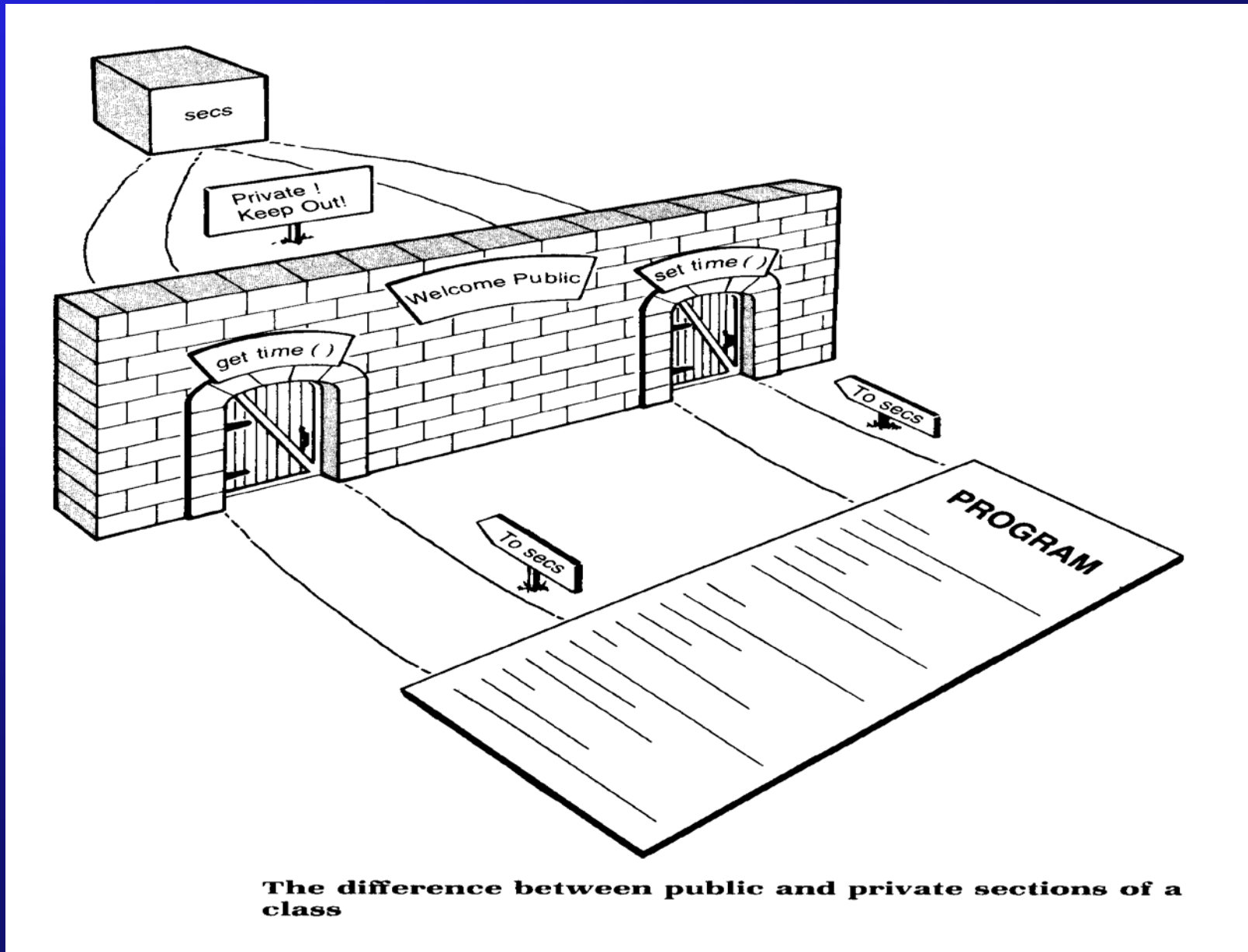




# Support de l'encapsulation

- **Pour supporter l'encapsulation :**
  - l'association des attributs et des méthodes : une information soumise à une contrainte ne sera mise à jour que par une méthode dans laquelle on aura programmé la condition à vérifier
  - des clauses de visibilité permettant de protéger un attribut (lecture, écriture)
  - la séparation entre l'interface (spécification) et l'implémentation (privée, invisible et inaccessible de l'extérieur de l'objet)

# Section publique et section privée

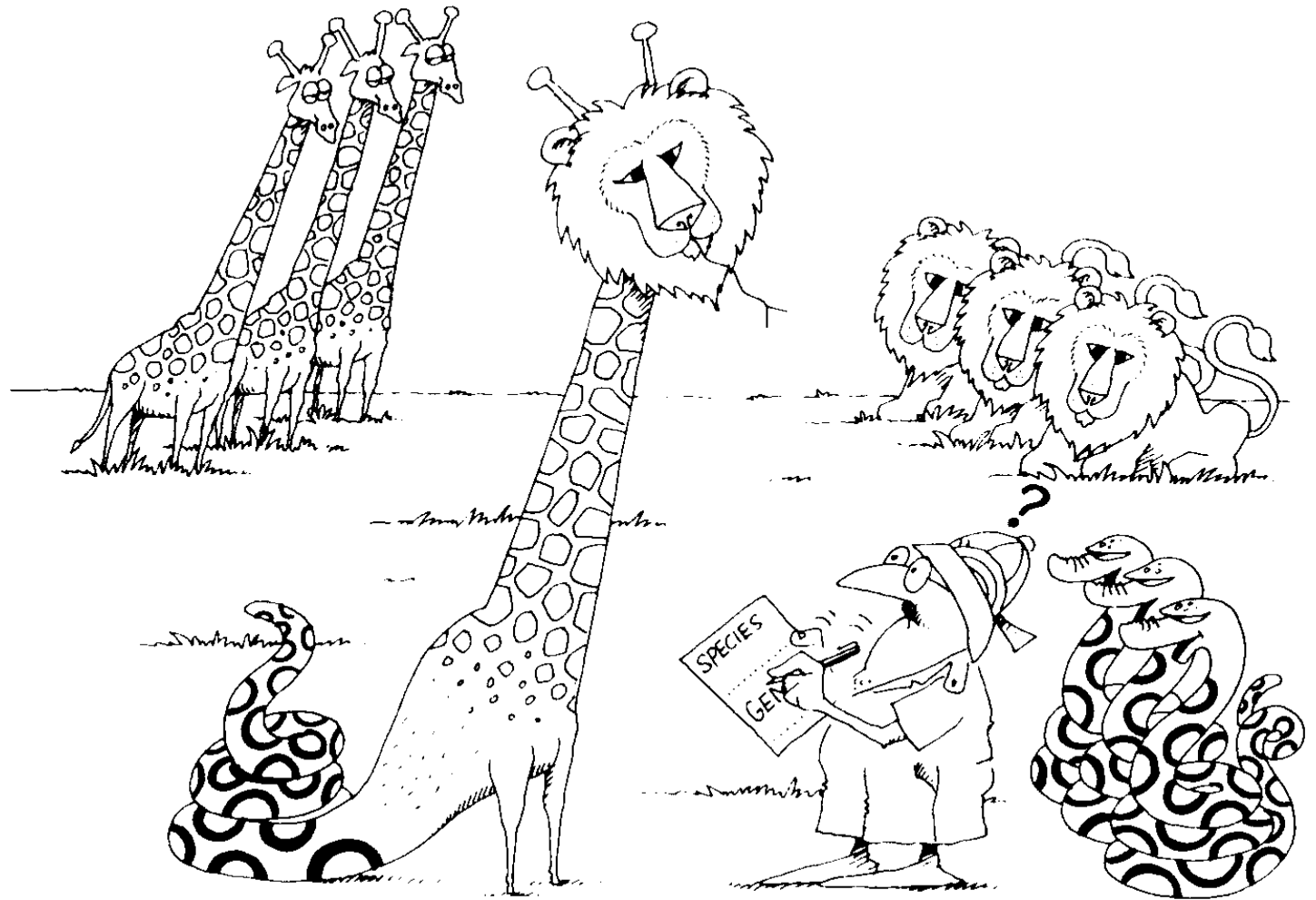


**The difference between public and private sections of a class**

# Intérêt de l'encapsulation

- **Une association stricte entre les données et opérations sans possibilité d'intervention extérieure**
- **Une maintenance facilitée**
- **Une compréhension et une documentation simplifiée**
- **Une structuration plus modulaire**

# La classification

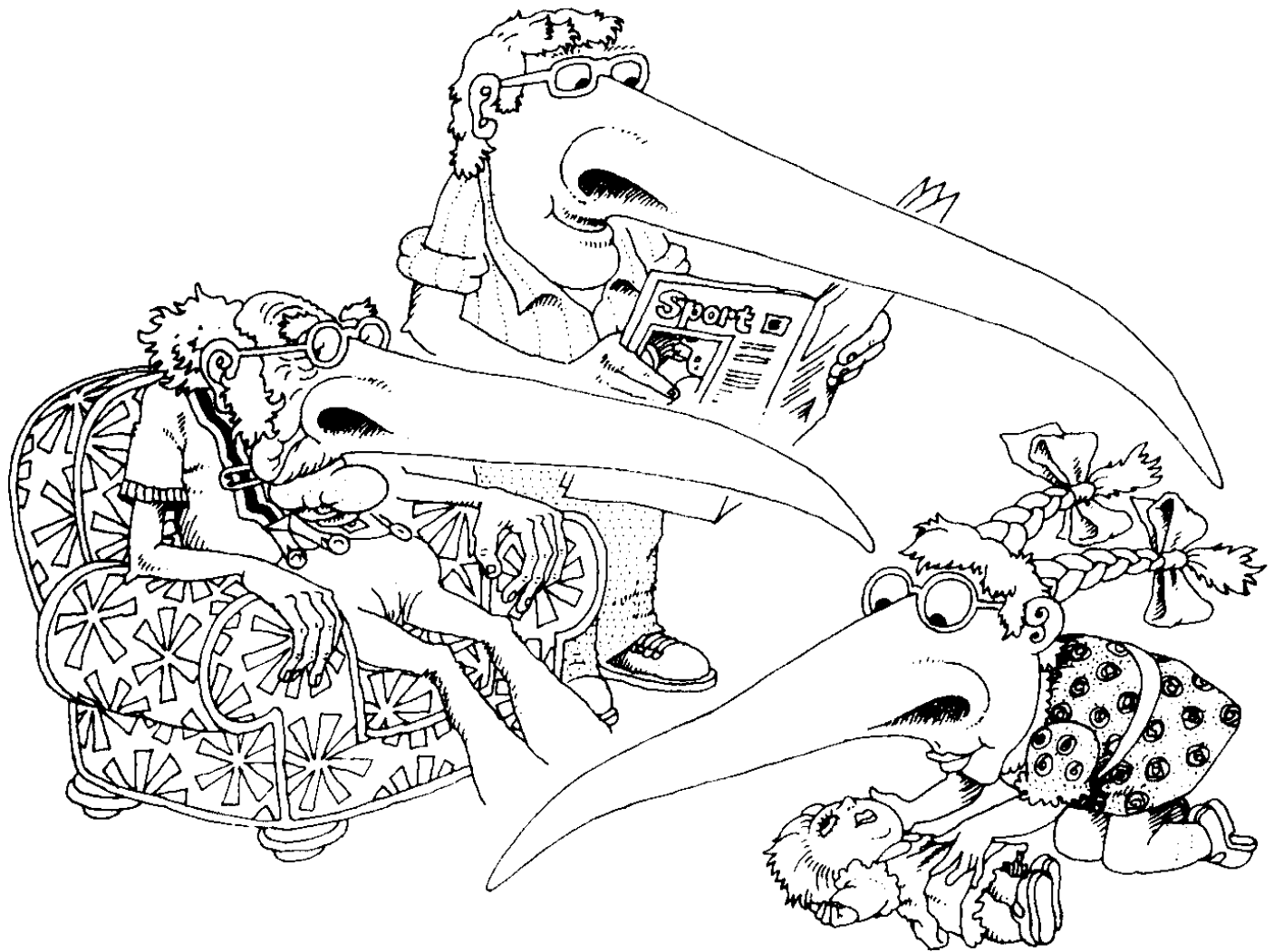


La classification est le moyen par lequel nous ordonnons nos connaissances

# La classification

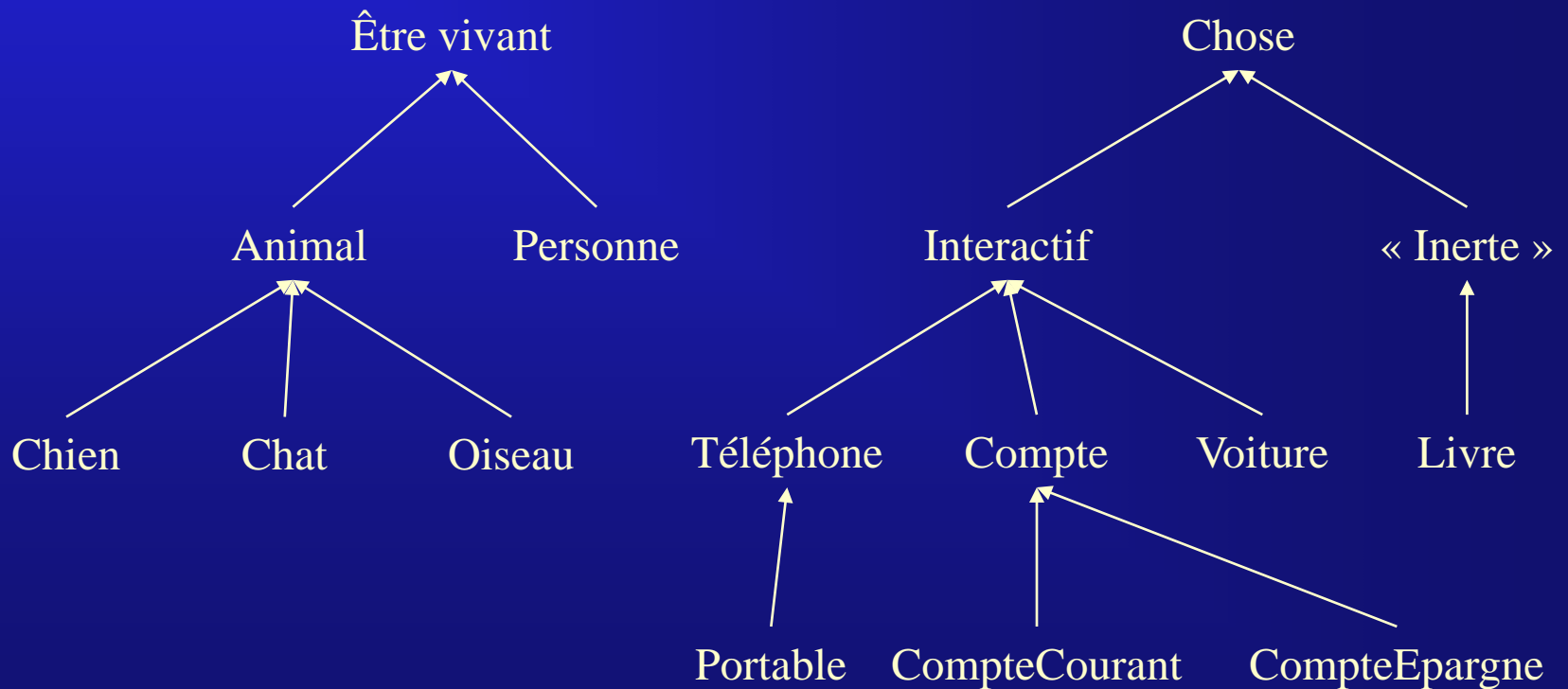
- Elle permet d'ordonner les connaissances du monde réel
- C'est un processus mental qui examine un ensemble d'entités, identifie les similitudes et différences, et groupe les entités similaires ensemble
- La classification est une technique de base pour établir une hiérarchie et décomposer un système
- La difficulté inhérente avec la classification est due au fait que différentes personnes peuvent avoir différents points de vue sur un sujet

# Héritage

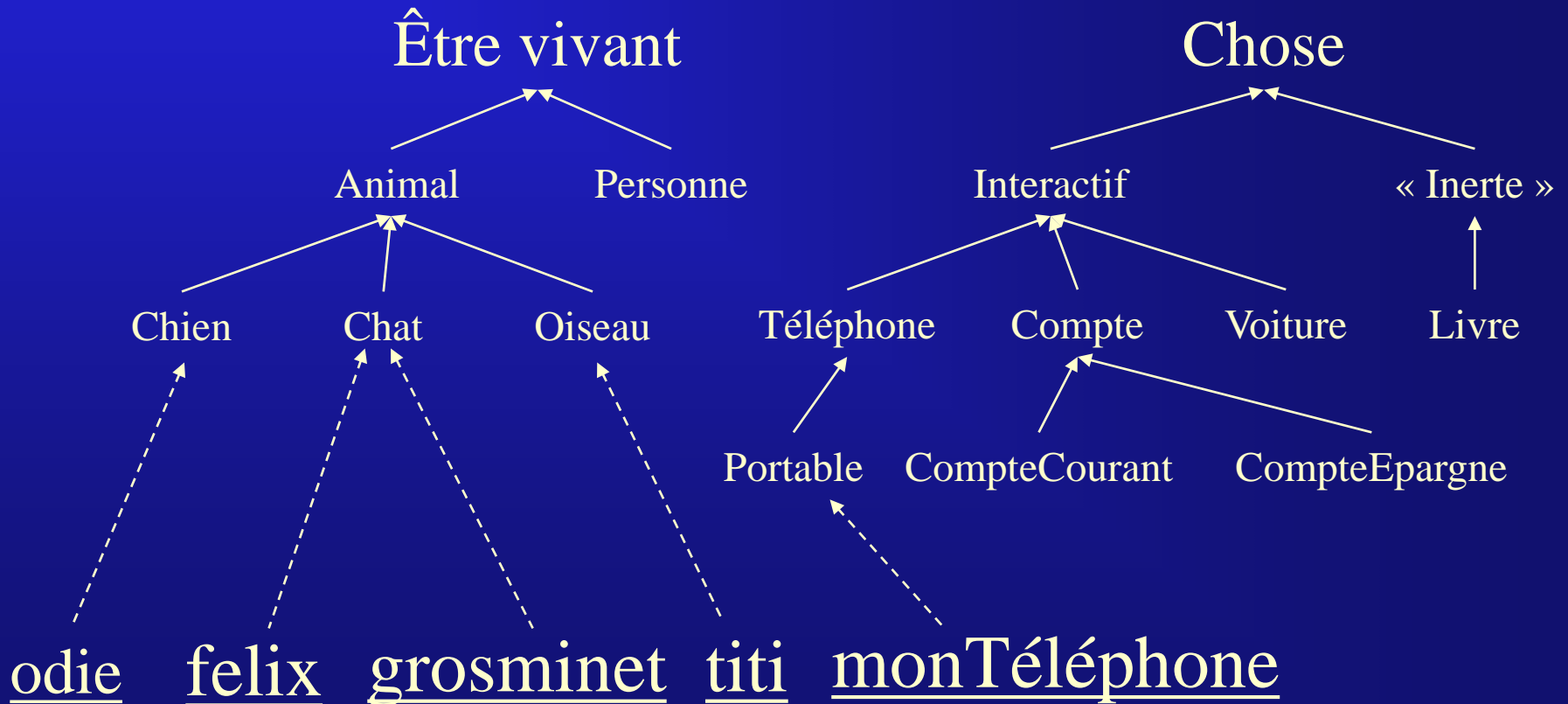


Une sous-classe peut hériter de la structure et du comportement de sa superclasse

# Catégoriser les classes



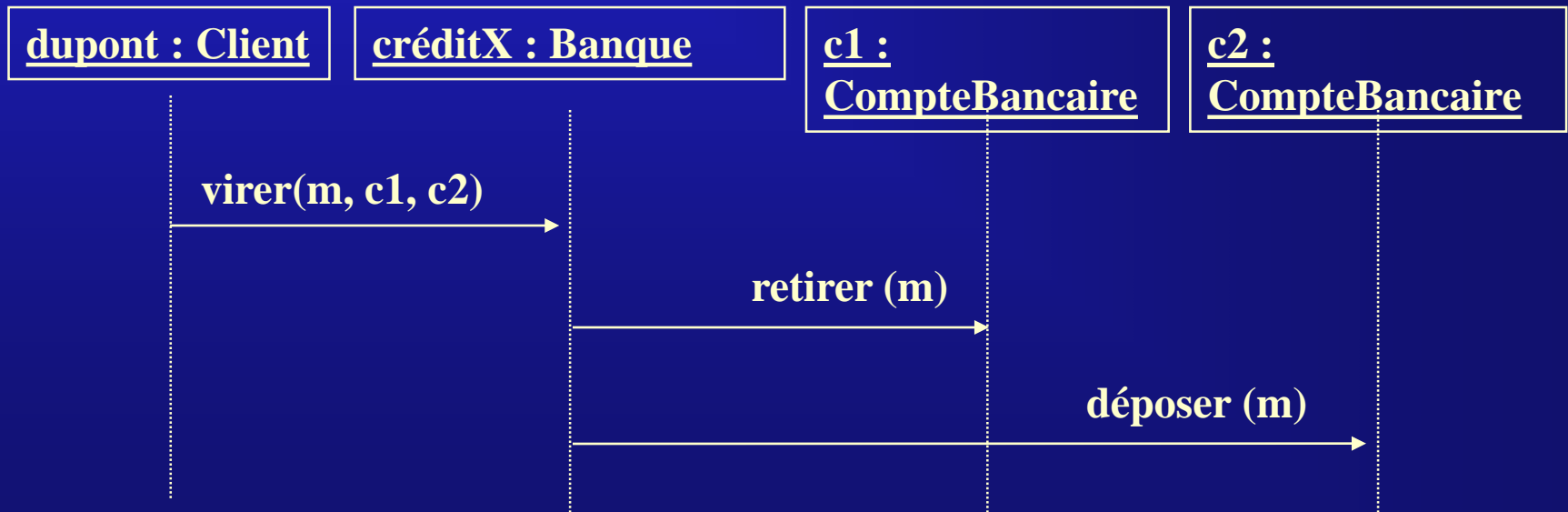
# Les objets sont rattachés à leur classe





# Les objets communiquent

- Permet de reconstituer une fonction par une mise en collaboration d'un groupe d'objets : envois de messages successifs



# Un point sur le vocabulaire

	Synonymes
<b>Objet</b>	<b>Instance</b>
<b>Identité d'objet</b>	<b>Référence</b>
<b>Classe</b>	
<b>Attribut (d'instance):</b> <b>1. Constant</b> <b>2. Variable</b>	<b>Variable (d'instance) :</b> <b>1. Constante</b> <b>2. Variable</b>
<b>Méthode</b>	<b>Opération, fonction, service</b>
<b>Message</b>	<b>Événement, appel, interrogation</b>
<b>Constructeur</b>	<b>Initialisateur</b>

# Les impacts du concept objet sur le développement

- Meilleure maîtrise de la complexité
- Meilleure réutilisation
- Meilleure productivité
- Plus grande fiabilité des logiciels mis en exploitation
- Meilleure évolutivité
- Penser objet oblige à **changer de mentalité !**