

# FRL



Prof. Santucci Jean-François

SPE – UMR 6134

UNIVERSITE DE CORSE

email : [santucci@univ-corse.fr](mailto:santucci@univ-corse.fr)

# PLAN



1. Introduction : Notions de Representation de la connaissance
2. Frames
3. Concepts de base d'un langage à base de frames
4. Implementation d'un langage à base de Frames

# PLAN



1. Introduction : Notions de  
Représentation de la connaissance
2. Frames
3. Concepts de base d'un langage à base de  
frames
4. Implementation d'un langage à base de  
Frames

# Problématique

- ❑ **La représentation de la connaissance** est un concept fondamental en Intelligence Artificielle.
- ❑ Chaque branche de recherche a une vision personnelle liée à son propre domaine de recherche.
- ❑ Il faut déterminer les **rôles** que jouent cette représentation et la place qu'elle occupe.

## choix ontologiques

- ❑ toutes les représentations sont des approximations imparfaites de la réalité
- ❑ chacune tient compte de certaines choses et en ignorant d'autres,
- ❑ choisir une représentation est un choix sur :  
‘**comment voir**’ et ‘**que voir**’ du monde
- ❑ c'est un choix ontologique.

## La Représentation des connaissances

Provient historiquement du travail d'autres secteurs :

- [?] La **logique mathématique** : variation du calcul formel et de la logique.
- [?] La **psychologie** : raisonnement comme une caractéristique du comportement humain
- [?] La **biologie** : architecture de la machine qui accomplit le raisonnement.
- [?] Les **statistiques** : notion d'incertitude et d'obéissance aux axiomes de probabilités ou possibilités
- [?] L'**économie** : ajoute les éléments de valeurs et de préférence.

## un moyen d'expression humain

- ❑ C'est le moyen à travers lequel nous exprimons les choses à propos du monde, le moyen d'expression et de communication par lequel nous disons à la **machine** (ou à l'autre) des choses sur le monde.
- ❑ Rôle de la représentation de connaissance est un moyen d'expression et de communication pour notre usage.

# Représentation des Connaissances et Objets

## *Notion d'objets*

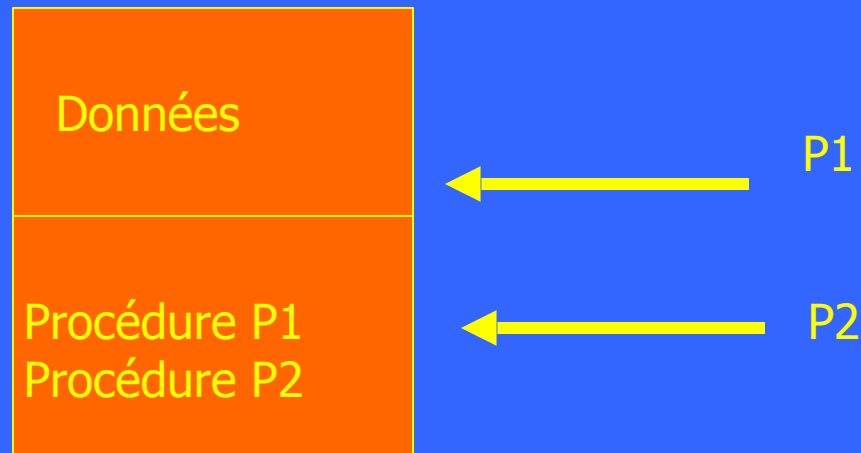
- ❑ Dans les deux cas, principes de base :
  - . Encapsulation
  - . Communication
  - . Création d'exemplaires
  - . héritage
  
- ❑ Élément de base : l'objet



# Représentation des Connaissances et Objets

## *Concepts de base*

- [?] Une encapsulation est un regroupement sous un même nom de données et des procédures qui les manipulent.



# Représentation des Connaissances et Objets

*Concepts de base*

## [?] Communication

. Par envoi de messages : on active les points d'entrée des encapsulations en leur envoyant des messages.

. A la réception d'un message, l'encapsulation exécute la procédure correspondante.

ATTENTION : Différence avec la Rep. Des Conn.

## [?] Création d'exemplaires

. Par instanciation : classes

## [?] Héritage : héritage de propriétés entre classes.

# Représentation des Connaissances et Objets

## *Encapsulation*

- ❑ Classiquement en informatique, on distingue deux entités bien distinctes : le code et les données.
- ❑ On déclare les données sans se soucier de l'utilisation qui en sera faite.
- ❑ Chaque donnée est donc visible de toutes les procédures.

# Représentation des Connaissances et Objets

## *Encapsulation*

- ❑ Grâce à la notion d 'encapsulation, une procédure ne peut manipuler que les données appartenant à une encapsulation
- ❑ Elle ne peut pas modifier les données d 'une autre encapsulation.

# Représentation des Connaissances et Objets

## *Encapsulation*

[?] Vue de l 'extérieur une encapsulation est :

- . Une région mémoire
- . Une entité unique et indépendante
- . Une liste de points d 'entrée.

[?] Vue de l 'intérieur une encapsulation est :

- . Des données structurées locales
- . Des procédures locales correspondant chacune à un point d 'entrée et qui manipulent ces données.

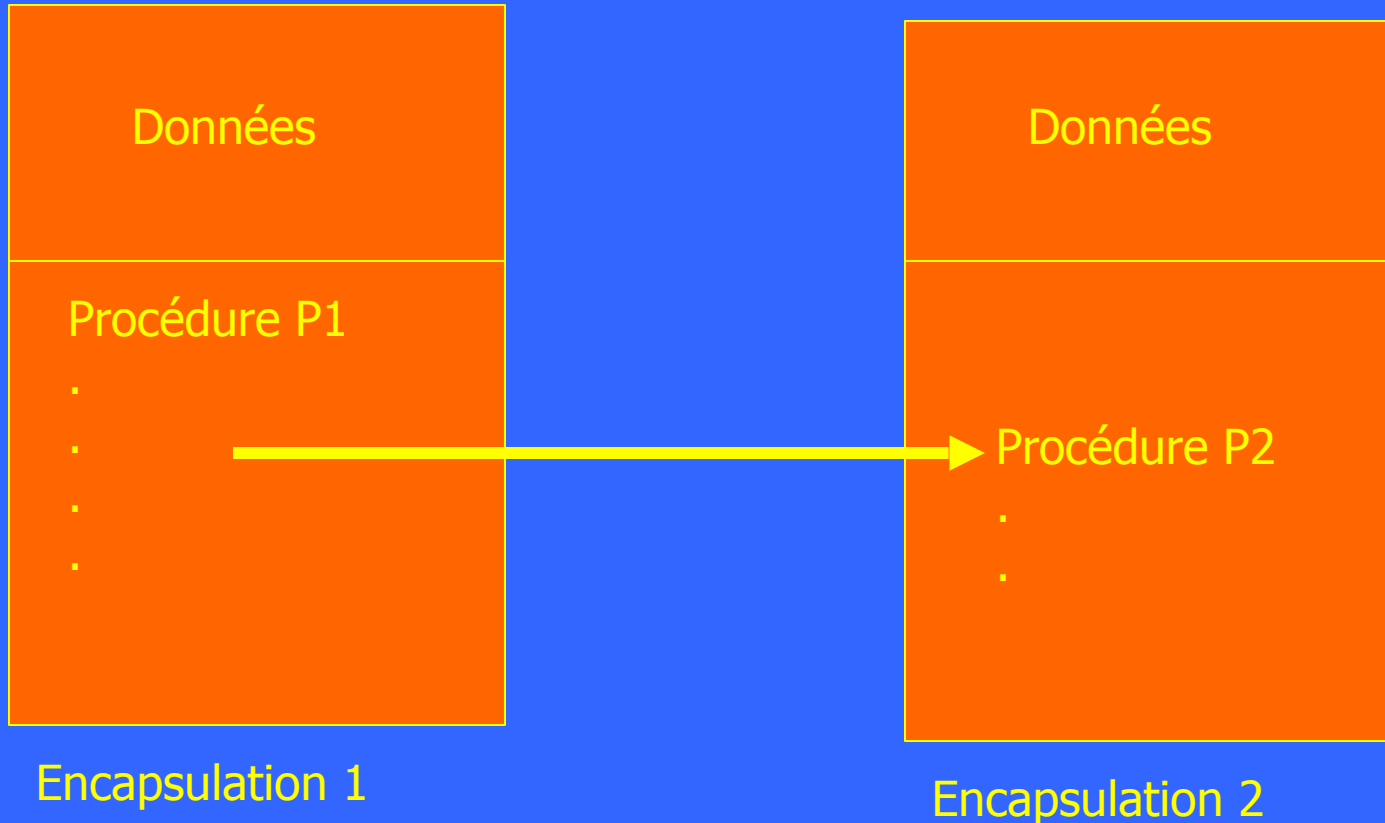
# Représentation des Connaissances et Objets

## *Envoi des messages*

- ❑ Une transmission de message doit spécifier :
  - . Le nom de l'encapsulation destinataire
  - . Le nom du point d'entrée à activer
  - . Des arguments éventuels si deux encapsulations s'échangent des valeurs.
- ❑ Le message ne sait rien de l'implémentation du destinataire. Il ne connaît que son nom.
- ❑ Cela permet de modifier la représentation interne sans changer les programmes appelants.

# Représentation des Connaissances et Objets

## *Envoi des messages*



# Représentation des Connaissances et Objets

## *Création d'exemplaires*

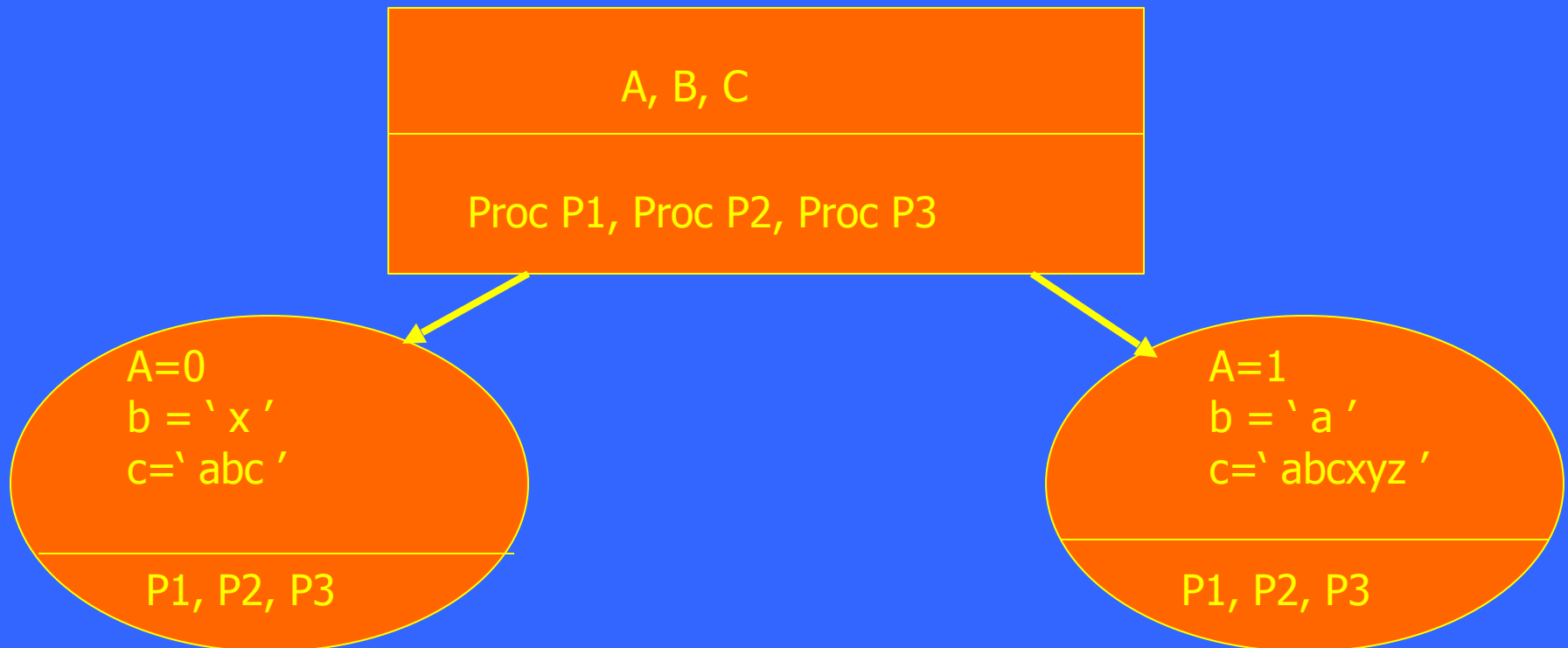
- ❑ Les objets ayant un même comportement seront représentés par un prototype qui définira leurs propriétés.
- ❑ Le prototype est une sorte de moule à partir duquel on fabrique autant d'éléments que l'on veut.
- ❑ Le prototype est un moyen de décrire un ensemble par définition.
- ❑ Les exemplaires d'un prototype sont les éléments de cet ensemble.
- ❑ Dans la terminologie des langages objets on parlera de classe.



# Représentation des Connaissances et Objets

## *Création d'exemplaires*

- [?] Les exemplaires ainsi créés sont appelés instances de la classe à partir de laquelle ils sont fabriqués.



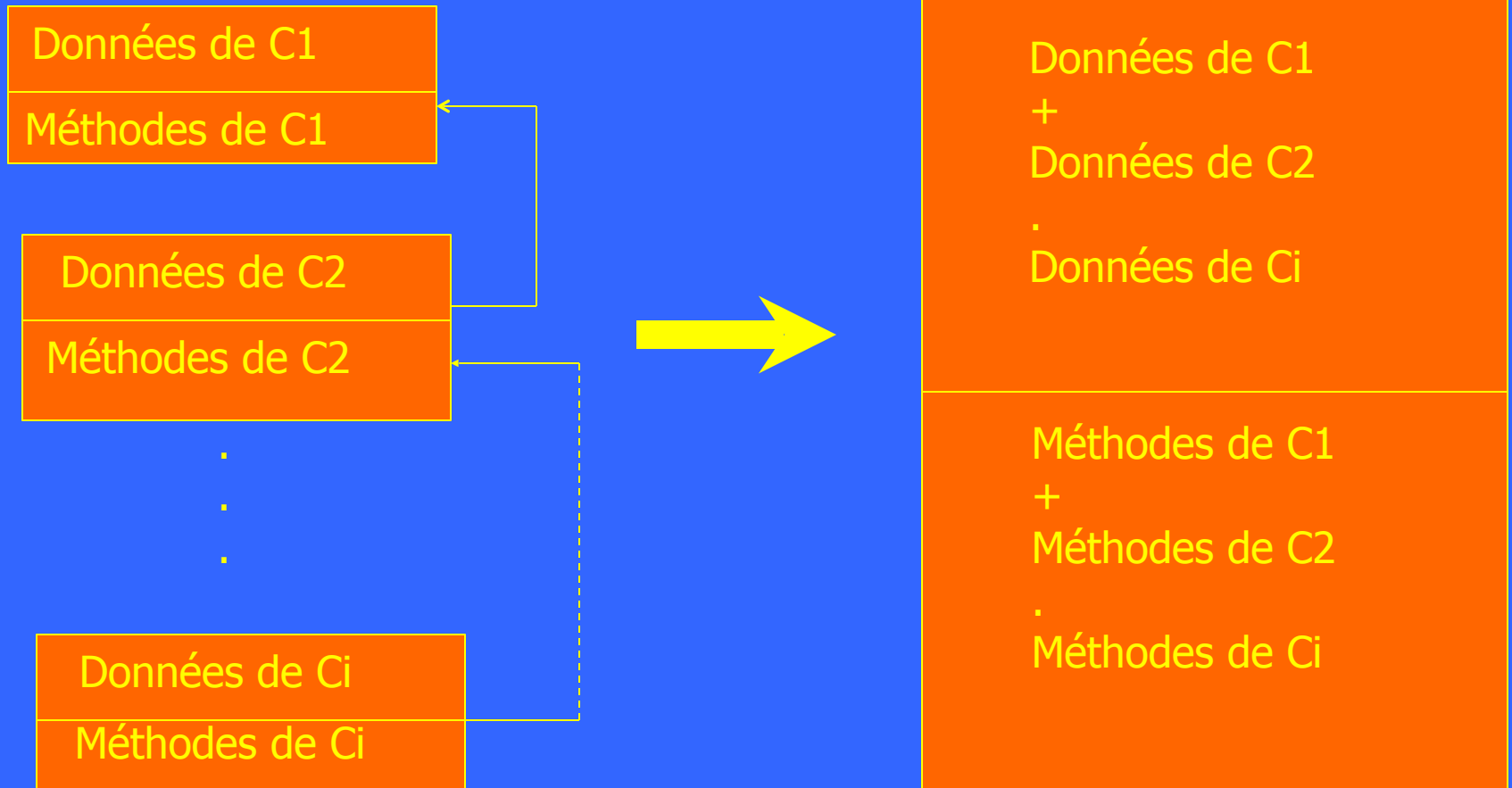
# Représentation des Connaissances et Objets

## *Héritage*

- ❑ On peut avoir envie de définir une classe possédant toutes les propriétés d 'une autre classe déjà définie, plus un certain nb d 'autres.
- ❑ Le mécanisme d 'héritage permet de ne définir que les propriétés particulières au nouvel objet.
- ❑ Créer un lien hiérarchique entre deux objets permet donc de faire hériter l'ensemble des données et des méthodes.

# Représentation des Connaissances et Objets

## *Héritage*



# Représentation des Connaissances et Objets

## *Arbre d'héritage*

- ❑ En conception orientée objet, tout objet du système que l'on crée doit être rattaché à un autre objet.
- ❑ Il existe dans un environnement de P.O.O. une classe prédéfinie : OBJET.
- ❑ Cette classe définit les fonctionnalités que doivent posséder toutes les classes.

# Représentation des Connaissances et Objets

## *Arbre d'héritage*

- ❑ En fait tout environnement de P.O.O est constitué d'un arbre d'héritage comprenant un certain nombre de classes prédéfinies.
- ❑ Toutes ces classes héritent de la classe OBJET.
- ❑ Quand on veut fabriquer une nouvelle classe, celle-ci doit s'insérer dans cette pyramide.

# Représentation des Connaissances et Objets

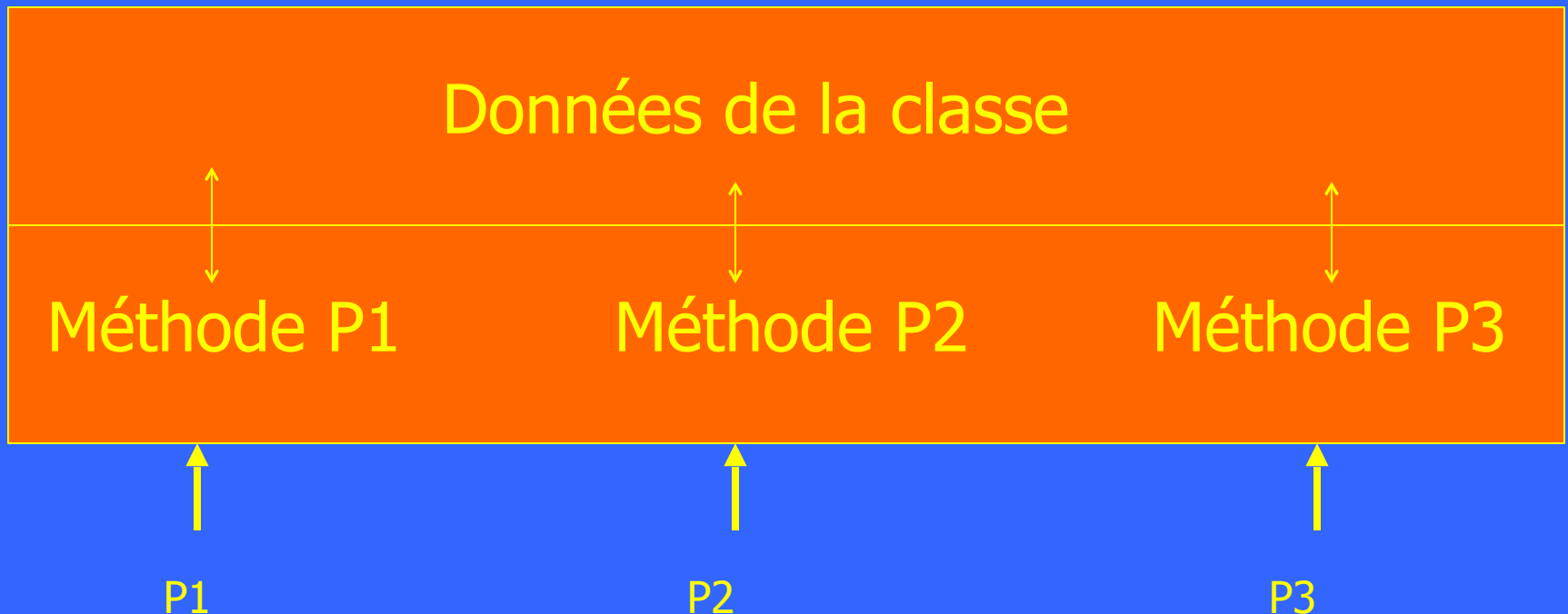
## *Arbre d'héritage*

- ❑ Toute nouvelle classe doit donc être fabriquée à partir d'une classe existante en créant un lien d'héritage.
- ❑ Une classe en P.O.O. ne peut en général avoir de lien d'héritage direct qu'avec un seul père : héritage simple.

# Représentation des Connaissances et Objets

## *Classe : exemple*

- [?] Les données internes ne sont accessibles que par les procédures locales à une classe qui s'appelle des méthodes.



# Représentation des Connaissances et Objets

## *Classe : exemple*

- [?]** Exemple : on décrit une pile comme étant un objet ayant les fonctionnalités suivantes :
- . Une pile doit pouvoir rendre le nb d 'éléments qu 'elle contient.
  - . On doit pouvoir initialiser la pile.
  - . On doit pouvoir savoir si une pile est vide ou non.
  - . On doit pouvoir empiler un nouvel élément en sommet de la pile.
  - . On doit pouvoir dépiler l 'élément situé en sommet de pile.

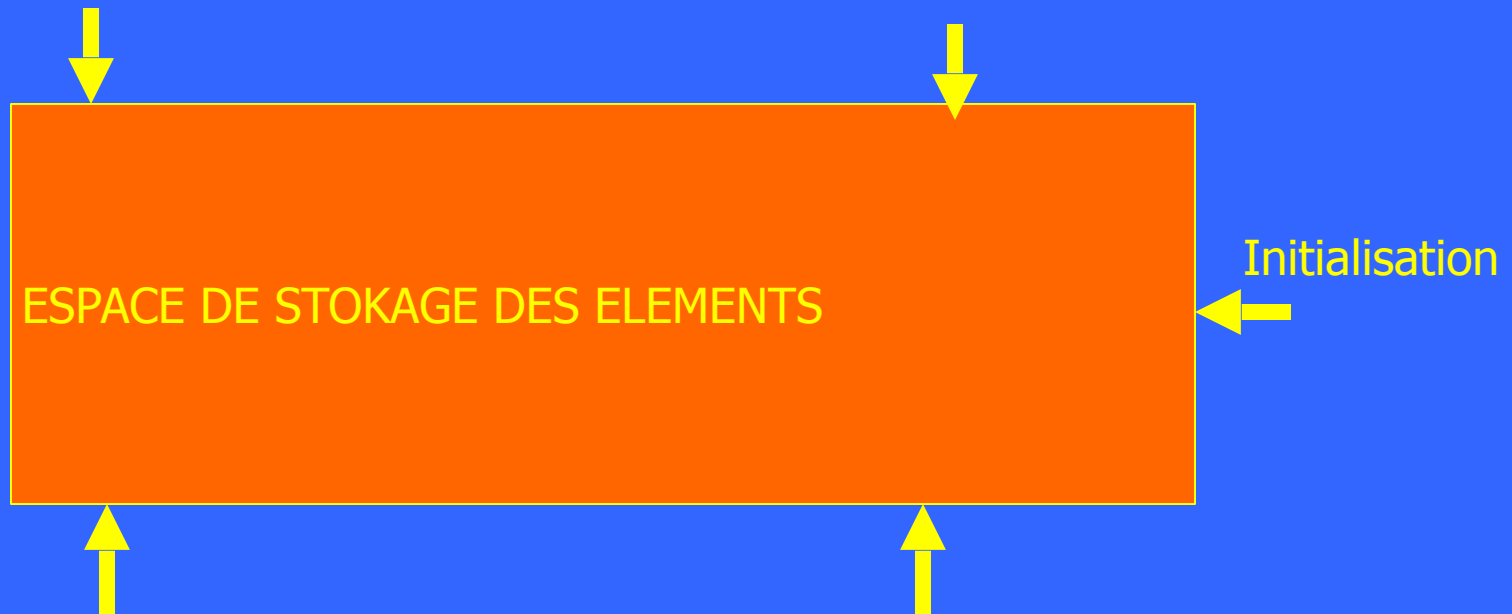


# Représentation des Connaissances et Objets

## *Classe : exemple d'une pile*

Empiler un élément

Dépiler un élément



Donner le nb d'éléments

La pile est-elle vide?

# Représentation des Connaissances et Objets

## *Classe : exemple*

- [?] Exemple : on décrit une pile comme étant un objet ayant les fonctionnalités suivantes :
- . Une pile doit pouvoir rendre le nb d 'éléments qu 'elle contient.
  - . On doit pouvoir initialiser la pile.
  - . On doit pouvoir savoir si une pile est vide ou non.
  - . On doit pouvoir empiler un nouvel élément en sommet de la pile.
  - . On doit pouvoir dépiler l 'élément situé en sommet de pile.