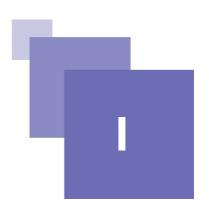
Cours 9 - CLOS

Table des matières

I - Présentation	5
A. Les classes	5
B. Les instances	6
C. L'héritage	7
D. Les méthodes	.
E. Envoi de message	8

Présentation



Les classes	5
Les instances	6
L'héritage	7
Les méthodes	7
Envoi de message	8

CLOS est une extension de Common Lisp.

Il permet de définir :

- Des classes
- Une hiérarchie de spécialisation.
- Des méthodes appelées fonctions génériques.

Le modèle est décrit par un méta-modèle

A. Les classes



Définition : Defclass

Une classe est définie à l'aide de la macro defclass



Exemple : Classe \$person

image1



Méthode : Le deuxième argument : ()

est utilisé pour indique les super-classes



Méthode : Le troisième argument est la liste de propriétés ou slots : nom, nationalite

- L'option :accessor définit une fonction à la valeur du slot dans une instance.
- L'option :initarg permet d'initialiser un slot au moment de la création d'une instance.
- L'option :initform permet de donner une valeur par défaut au slot au moment de la création d'une instance.
- L'option :type permet de typer le slot (type lisp ou bien un nom de classe).

B. Les instances



Définition: make-instance

La création d'une instance se fait à l'aide de la fonction make-instance.



Exemple

(setq \$0 (make-instance '\$person :nom "Albert" :nationalite "belge"))



Définition : describe

Le contenu d'un objet peut être examiné à l'aide de la fonction describe.



Exemple

(describe \$0)



Définition : Interrogation d'un slot

La valeur d'un slot peut être récupérée à l'aide de la fonction précisée au moment

de la définition de la classe par l'option :accessor.



Exemple

>(nom \$0)

"Albert"

>

Modification de la valeur d'un slot

La valeur de la valeur d'un slot peut être modifiée en utilisant son accesseur et la forme spéciale setf.



Exemple

>(setf (nom \$0) "Jacques")

C. L'héritage

Une classe peut être définie par rapport à une classe déjà existante :

La classe plus générale est précisée dans le deuxième argument de defclass.



Exemple : La classe student

(defclass \$student (\$person) ((\$takes :type list :accessor UV :initarg :prend)))



Remarque

On peut alors créer des étudiants en utilisant les propriétés définies dans \$student mais aussi celles héritées de \$person.



Exemple: instance \$1

(setq \$1 (make-instance '\$student :nom "Jean" :prend (list \$3))



Attention : Héritage multiple

Dans CLOS, une sous classe peut hériter de plusieurs classes simultanément. (defclass actionnaire (personne employe) ...)

D. Les méthodes



Définition : defmethod

Une méthode est définie à l'aide de la macro defmethod.



Exemple

(defmethod =print-self ((xx \$person)) (format t "~& ~A ~A" (nom xx) (nationalite xx)))



Remarque

On note la forme particulière de la liste d'arguments qui précise pour le premier argument xx la classe \$person de celui-ci.



Remarque

La classe étant précisée, on peut utiliser les accesseurs correspondants aux slots de celle-ci.

E. Envoi de message

La syntaxe d'un envoi de message en CLOS ressemble à un appel de fonction.



Exemple

>(=print-self \$0)



Exemple

Si on essaie d'appliquer cette méthode à un objet qui n'est pas de la bonne classe, le comportement par défaut de CLOS est de signaler une erreur.



Attention

On peut définir une méthode =print-self pour une autre classe.

Il faut toutefois que la liste d'arguments soit de la même forme que dans la méthode qui a été définie (nombre d'arguments).