

Description

- Ce cours traite de caractéristiques présentes dans les langages à objets les plus avancés.
- Les caractéristiques couvertes concernent principalement le typage statique, l'héritage et la méta-programmation.

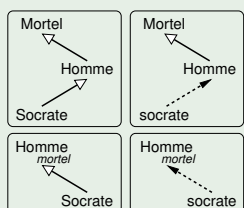
Objectifs

- Connaître les caractéristiques avancées des langages à objets.
- Comprendre les liens entre ces différentes caractéristiques.
- Être capable d'apprécier et de mettre en œuvre ces caractéristiques.

Concepts fondamentaux

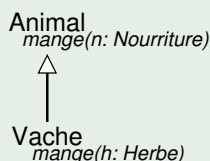
- Objet
- Classe
- Spécialisation
- Héritage
- Propriété
- Envoi de message

Syllogisme



« Tous les hommes sont mortels or Socrate est un homme donc Socrate est mortel » [Aristote]
→ 4 façons de voir les choses

Contravariance



```
Animal a = new Vache();
a.mange(new Saucisse());
→ la vache folle est-elle une erreur de typage?
```

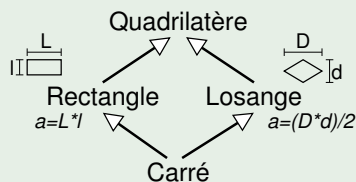
Typage et sélection de méthodes

- Sous-typage vs. spécialisation
- Covariance vs. contravariance
- Types paramétrés et généricité
- Coercitions
- Sélection multiple et multiméthodes
- Surcharge statique
- Appel à super

Héritage multiple et variations

- Conflits de propriétés
- Techniques de linéarisation
- Héritage d'interface
- Héritage non conforme
- Héritage de mixins, de traits

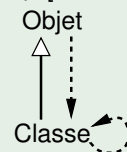
Conflit de propriétés



Comment calculer l'aire d'un carré ?

Réflexivité

Noyau du modèle réflexif ObjVlisp [Cointe 1987]



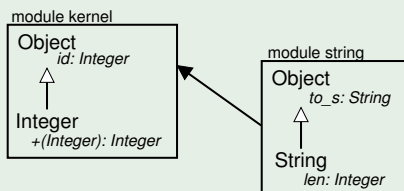
Méta-programmation

- Méta-modélisation
- Introspection
- Réflexivité

Hierarchies de modules et raffinement de classes

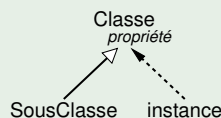
- Modules vs. classes
- Classes ouvertes
- Hierarchies d'ordre supérieur

Modules kernel et string de Nit



```
import string
123.to_s.length
# result is 3
→ Étonnant, non ?
```

Légende



Langages

- C++, C#, Clojure, Eiffel, Java, Nit, Python, Ruby, Smalltalk, etc.