

# *UML par l'exemple*

## **Introduction au langage de modélisation Unified Modeling Language**

*PEROUMALNAÏK M. 2006-2007*



# Plan de présentation

- L'héritage en deux mots
- Formalisme UML
- Umbrello : un outil complet de modélisation
- Exemple : définition de l'univers
- Exemple : mise en place de cet univers
- Exemple : génération du code
- Conclusion



# L'héritage en deux mots

- Propriété **exclusive** à la programmation orientée objet
- Permet de traduire simplement des **dépendances sémantiques** au sein d'un univers donné.
- Notions liées : **Classe abstraite**, **Polymorphisme**, Héritage Multiple (pas en C++).



# L'héritage en deux mots

- **Classe** : définition d'un objet dans un contexte sémantique donné (instance)
- **Classe abstraite** : classe avec des méthodes « génériques » dites abstraites
- **Polymorphisme** : mécanisme permettant la redéfinition de méthode(s) héritée(s).



# Formalisme UML

- **U. M. L.** : Unified Modeling Language
- Langage de **modélisation** objet
- **But** : clarifier et simplifier la conception et la représentation d'un système donné et/ou de son fonctionnement
- **Schématisation simple** compréhensible par l'homme et la machine



# Formalisme UML

- **Elements de base**
- **9 types** de diagrammes définissables dont :
  - Diagrammes de **classes**
  - Diagrammes d'**objets**
  - Diagrammes de **séquence**
  - Diagrammes de **cas d'utilisation**



# Umbrello

- Logiciel **Open-Source** (libre de droit et de modifications)
- Version actuelle : 1.5.61
- **Modeleur UML**
- Propose traduction en code :
  - C++
  - Java
  - Python
  - PHP...



# Umbrello

- (+) Schématisation proposée fidèle aux standards UML
- (+) Permet d'inclure du code aux diagrammes de classe
- (+) Prise en main rapide et intuitive
- (-) Ne fonctionne que sous Linux
- **Doc** en anglais:

(HTML) [http://docs.kde.org/stable/en\\_GB/kdesdk/umbrello/index.html](http://docs.kde.org/stable/en_GB/kdesdk/umbrello/index.html)

(PDF) <http://docs.kde.org/stable/en/kdesdk/umbrello/umbrello.pdf>



# Exemple

- Aborder la conception d'un système à travers un exemple simple
- On cherche à représenter simplement dans plusieurs dimensions les objets suivants :
  - Un point
  - Une droite
  - Un cercle



## Ex : Définition de l'univers (1/3)

- Associer ces objets à des **propriétés** précises
- Hierarchiser ces propriétés de la plus générale à la plus spécifique
- Effectuer le même travail avec les **méthodes** (actions ou groupe d'actions effectuées par/sur l'objet)



# Ex : Définition de l'univers (2/3)

- *Exemple du Point*
- **Propriétés**
  - Coordonnées
- **Méthodes**
  - Déplacer
  - Afficher
- *Problème* : les coordonnées varient suivant l'espace considéré.

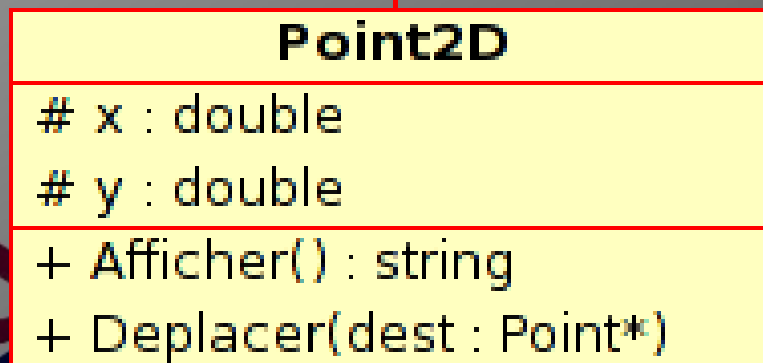
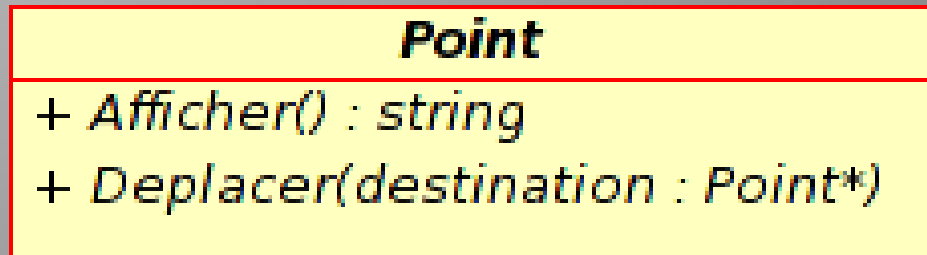


## Ex : Définition de l'univers (3/3)

- *Exemple du Point (suite)*
- Propriété coordonnée  $\Rightarrow$  niveau d'abstraction bas (type, nombre).
- Méthodes Afficher et Déplacer  $\Rightarrow$  niveau d'abstraction élevé, redéfinissable par la suite (polymorphisme)
- Définition de plusieurs classes traduisant l'objet Point



# Ex : Mise en place de l'univers



- Classe abstraite **Point**

- Héritage en UML

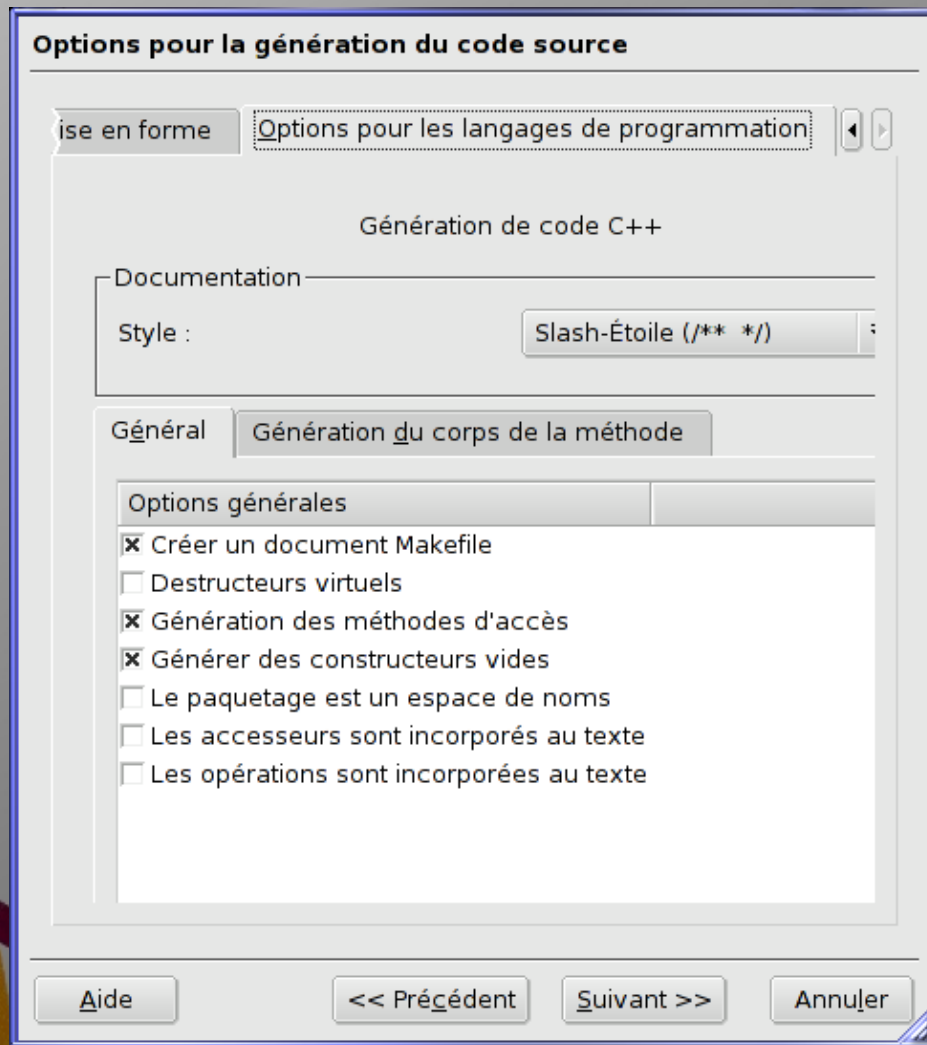
– Classe **Point2D**

– Classe **Point3D**

- Exemple du polymorphisme



# Ex : Génération du code



- Utiliser l'assistant de génération du code

Exemple de réglages en C++

- Commentaires

# Conclusion

- **UML** : langage de modélisation simple et concis, « indispensable » en ingénierie logicielle
- Plusieurs contextes d'utilisation possible
- **Umbrello** : alternative libre opposable aux modeleurs UML présents sur le marché
- Prise en main compétitive



# FIN

Fin de présentation,  
merci de votre attention

PEROUMALNAÏK M. 2006 – 2007

mail : [mperouma@univ-ag.fr](mailto:mperouma@univ-ag.fr)

web : <http://grimaag.univ-ag.fr/~mperouma>

