

# Résumé du formalisme UML (Conception)

## Diagramme de Classe

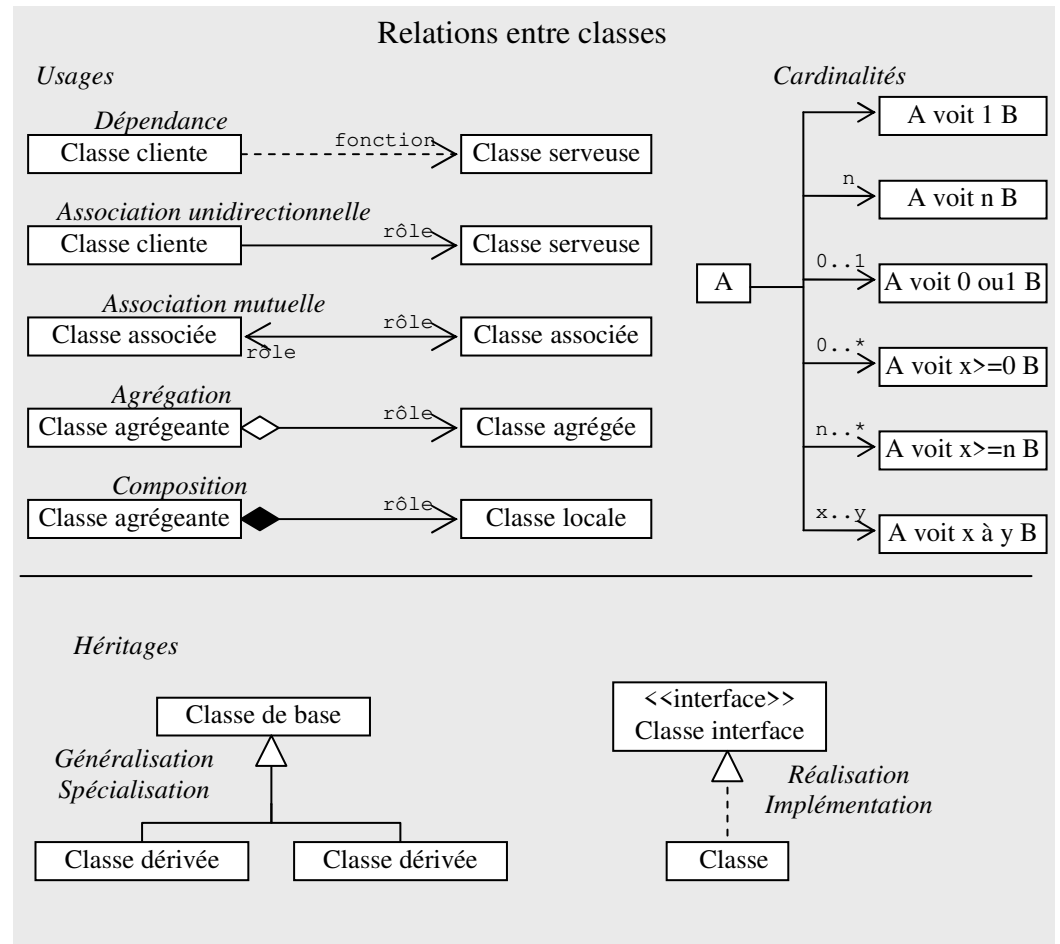
Classe abstraite

<<Interface>>  
Classe interface

<<Exception>>  
Classe

type  
Classe paramétrée

Classe externe  
(package)



### Niveau de raffinement

#### Diagramme global

Classe

#### Diagramme partiel

Classe  
attribut : type  
\$attribut de classe :type

#### Diagramme partiel

Classe  
opération  
opération abstraite  
\$opération de classe

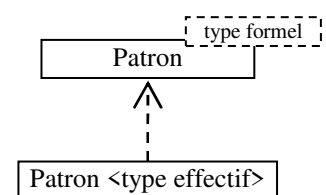
#### Diagramme complet

Classe  
attribut : type [=val initiale]  
\$attribut de classe :type [=val initiale]  
Opération(param : type)[ :type retour]  
opération abstraite(param : type)[ :type retour]  
\$opération de classe(param : type)[ :type retour]

**Visibilités**  
- membre privé  
+ membre public  
# membre protégé

\$ : de classe

### Instanciation de classe paramétrée



## Les mots clés

Fonction en POO	C#	Java	PHP	Python
Méthode abstraite	abstract	abstract		
Accéder Méthode classe de base	base	super	parent::	parent. ou super().
Définir une classe	class	class	class	class
Constructeur	NomClasse()	NomClasse()	__construct()	def __init__
Destructeur	~NomClasse()	finalize()	__destruct()	def __del__(self):
Définir une interface	interface	interface	interface	-
Fichiers d'un même assembly	internal			
Créer un objet	new	new	new	Nom de la classe
Surcharge d'une Méthode	override			
Attribut/Méthode privée	private	private	private	-
Attribut/Méthode accessible par la classe dérivée	protected	protected	protected	-
Attribut/Méthode publique	public	public	public	-
Empêche la surcharge dans classe dérivée	sealed	final	final	-
Attribut/Méthode de classe	static	static	static	Déclaration en tête de classe
Référence de l'objet en cours	this	this	\$this	self
Permet à ce que la classe dérivée surcharge la Méthode	virtual			
Classe dérivée	:	extends	extends	class child(parent)
Implementer une interface	:	implements	implements	-