

Sommaire



Introduction

- Les limites de la programmation procédurale

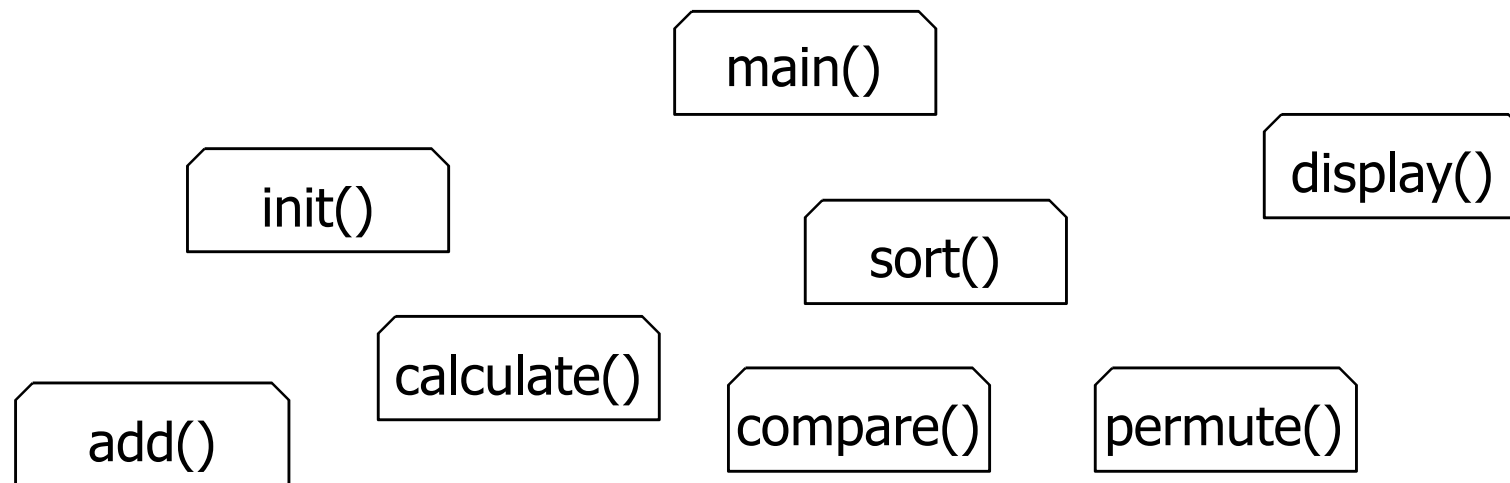
CONCEPTS ORIENTES OBJET

Les limites de la programmation procédurale



Programmation procédurale

- Programmation orientée fonction
 - Le programme est une fonction principale
 - Cette fonction appelle d'autres fonctions
 - Qui en appellent d'autres...



CONCEPTS ORIENTES OBJET

Les limites de la programmation procédurale



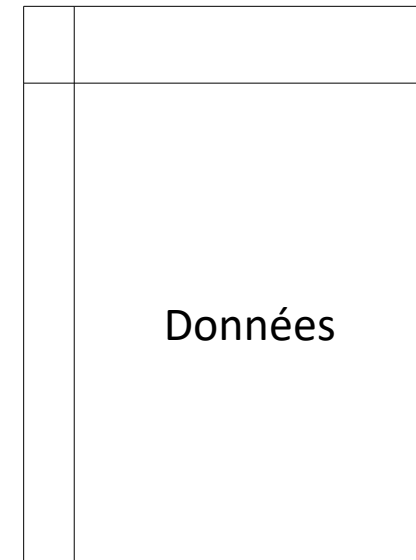
Programmation procédurale

- Séparation des données et des traitements

fonction1()

fonction2()

fonction3()



CONCEPTS ORIENTES OBJET

Les limites de la programmation procédurale



Fonctions moyennement réutilisables.

La modification d'une structure de données entraîne de multiples corrections.

Programmes peu évolutifs et maintenance très couteuse.

Décomposition pas assez structurante pour des programmes complexes.

Ressemble peu à notre schéma de pensée. Très près du langage machine.

Analyse du code existant peu intuitive.

Modularité et abstraction absente « code spaghetti »

Travail d'équipe difficile car pas de modularité. Qualité du code laisse à désirer.

CONCEPTS ORIENTES OBJET

Historique



Années 60 : Simula

Années 70 : Smalltalk

Années 80 : Objective C, C++, Lisp...

- Année 1983 : La POO devient populaire (C++)

Années 90 : Java, ...

Années 2000 : C#, PHP5

Et aussi : JavaScript, ActionScript, ...

CONCEPTS ORIENTES OBJET

Les limites de la programmation procédurale



En Programmation Orientée Objet :

- On retrouve les « ingrédients » de la programmation procédurale
- Le programme est structuré autour des objets manipulés qui portent les données
- Les traitements sont associées aux objets

CONCEPTS ORIENTES OBJET

La programmation orientée objet



Programmation centrée sur les données

Tout tourne autour des objets : ensemble d'entité qui représentent leurs propriétés.

- Exemples
 - Un chat : 4 pattes, une queue, appartient à la famille des félins
 - Une automobile : 4 portes, familiale, moteur 6 cylindres, automatique, de couleur rouge, essence...
 - Une chaussure : En cuir, montante, avec des lacets ...

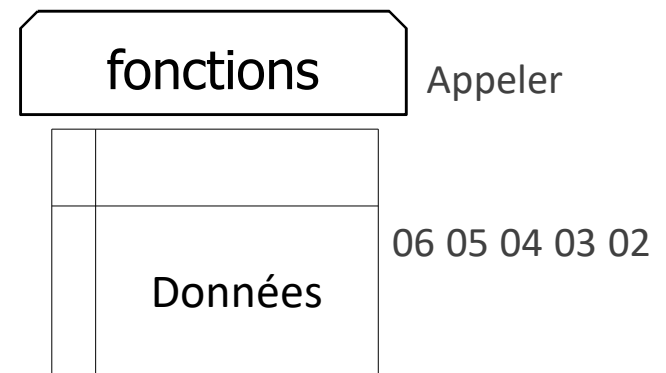
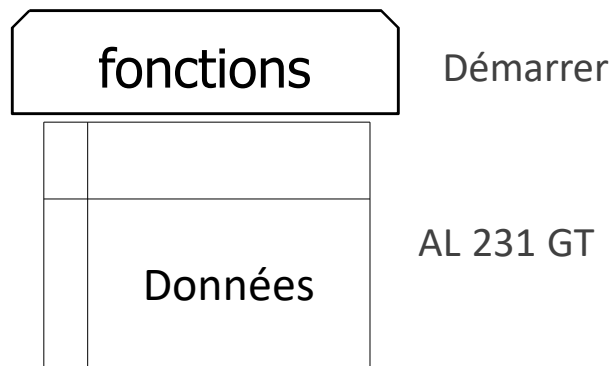
CONCEPTS ORIENTES OBJET

Le paradigme Objet



Un objet :

- est une « brique » logicielle dont se compose le programme.
- représente une entité du monde réel ou un concept.



CONCEPTS ORIENTES OBJET

Classes d'objet et structuration du code



En OO, tout est objet.

D'où séparation entre

- Classes représentant des objets du monde réel (Personne, Livre, Commande, Véhicule...)
→ Modèle métier
- Classes techniques (algorithmes, structuration du code, accès à la base de données...)

CONCEPTS ORIENTES OBJET

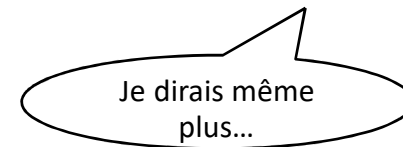
Le paradigme Objet



Un objet :

- est défini par
 - une identité
 - des propriétés
 - et des comportements

Dupond Dupont



- interagit avec les autres objets du système pour réaliser des fonctionnalités attendues.

CONCEPTS ORIENTES OBJET

Les avantages de l'Orienté Objet



Une organisation du code

- Intuitive
- Structurée
- Factorisée

Qui permet :

- Une meilleure lisibilité
- Une programmation simplifiée
- Une réutilisabilité accrue
- Une bonne maintenabilité
- Une meilleure sécurité des comportements
- Un travail en équipe

CONCEPTS ORIENTES OBJET

Définitions (1/2)



Classe: description des caractéristiques d'un ou de plusieurs objets. Chaque objet créé à partir de cette classe est une instance de la classe en question.

Prototype : Description d'un objet. Utilisé pour créer d'autres projets par clonage.

Méthode : suite d'instructions qui manipule les caractéristiques et l'état d'un objet.

Propriété : champ, attribut, field. Caractéristique d'un objet donné. Ex: nom, prénom, âge...

CONCEPTS ORIENTES OBJET

Définitions (2/2)

- **Etat d'un objet** : forme de l'objet à un instant donné. L'ensemble de ses propriétés disposent de valeurs. Ex: nom = gharbi prénom=safia ...
- **Sous classe /super classe** : Une classe hérite des caractéristiques d'une autre classe. Ses instances partagent les caractéristiques des 2 classes. Ex: classe Mouton/classe Animal.
- **Polymorphisme** : Lorsqu'il existe plusieurs sous classes d'une même super classe. Chaque sous classe dispose de sa propre variante d'une même méthode. Ex: classe Chat/ classe Mouton/classe Animal.
- **Encapsulation** : Méthodes, propriétés, état d'un objet manipulables à partir d'autres objets uniquement si ils sont déclarés comme publics.

CONCEPTS ORIENTES OBJET

Classes et instances



Une classe

- une structure définissant l'identité, les propriétés et les comportements d'un type d'objets

Une instance = Un objet

- un exemplaire bien identifié de cette classe

Classe : Personne

-Nom
-Prénom
-Date de naissance

Instance : Martin, Michel, 04/12/1950

Instance : Wagner, Paul, 12/04/1985

Instance : Dupuy, Isabelle, 21/09/1977

CONCEPTS ORIENTES OBJET

Exemple



Classes d'objets du domaine métier d'une médiathèque :

- Adhérent
- Média (Livre, revue, CD, DVD, ...)
- Réservation
- Emprunt

CONCEPTS ORIENTES OBJET

Exercice



**Classes d'objets d'un domaine métier
à choisir ?**

CONCEPTS ORIENTES OBJET

L'abstraction : définition



En [philosophie](#), l'[abstraction](#) désigne à la fois une opération qui consiste à isoler par la pensée une ou plusieurs qualités d'un objet concret pour en former une représentation intellectuelle, et le produit de cette opération. *Wikipedia*.

En un mot : abstraction = simplification

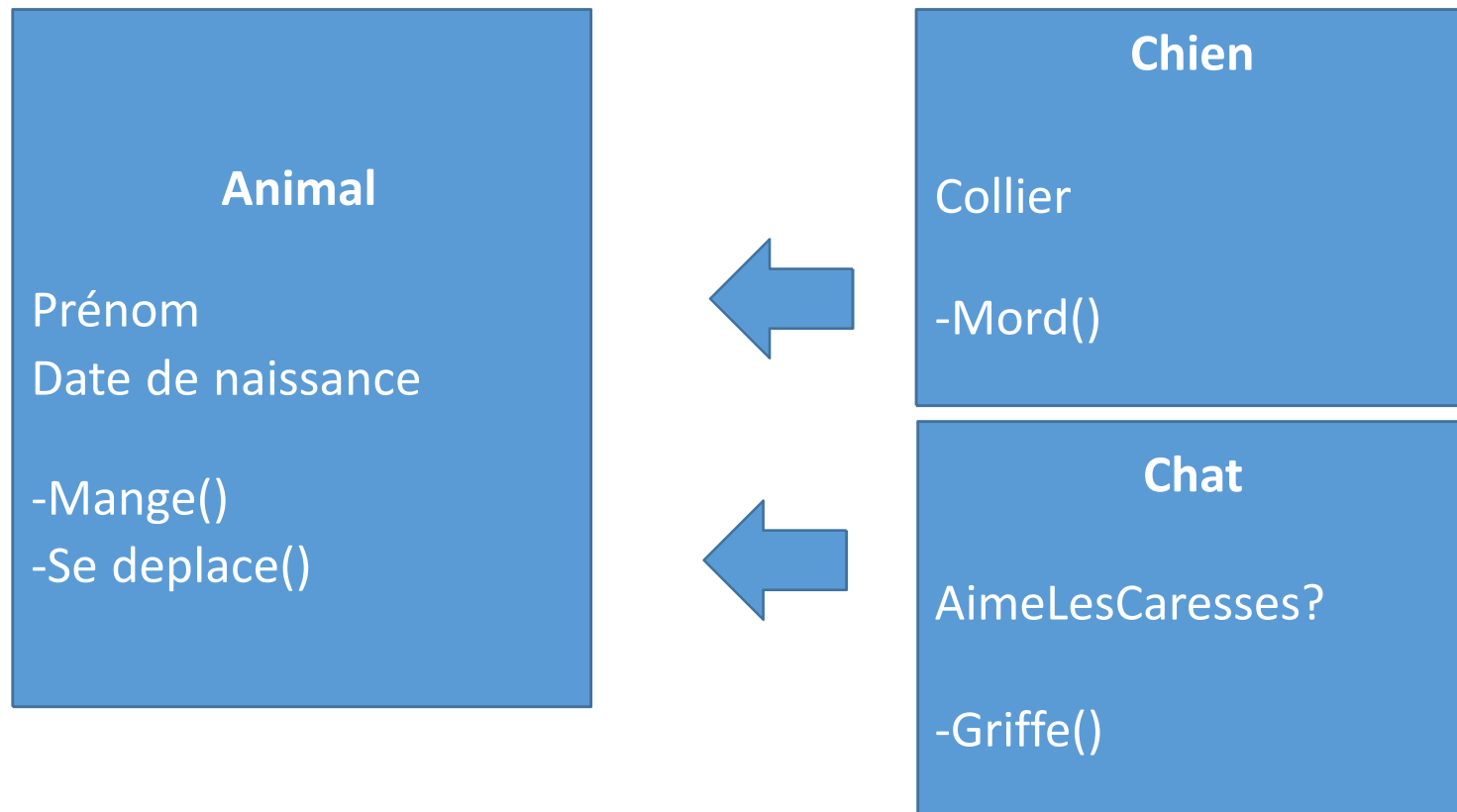
CONCEPTS ORIENTES OBJET

L'abstraction : classification



Principe

- Classification des objets : regroupement des caractéristiques et des comportements communs à des entités



CONCEPTS ORIENTES OBJET

L'abstraction : *information hiding*



Principe

- Exposition d'éléments choisis, permettant de cacher le comportement de l'applicatif à l'utilisateur
- Exemple : un téléphone
 - Comportement exposé : répondre à un appel
 - Comportement caché : mettre en contact, enregistré la voix, la renvoyer, etc.

CONCEPTS ORIENTES OBJET

L'encapsulation



Principe

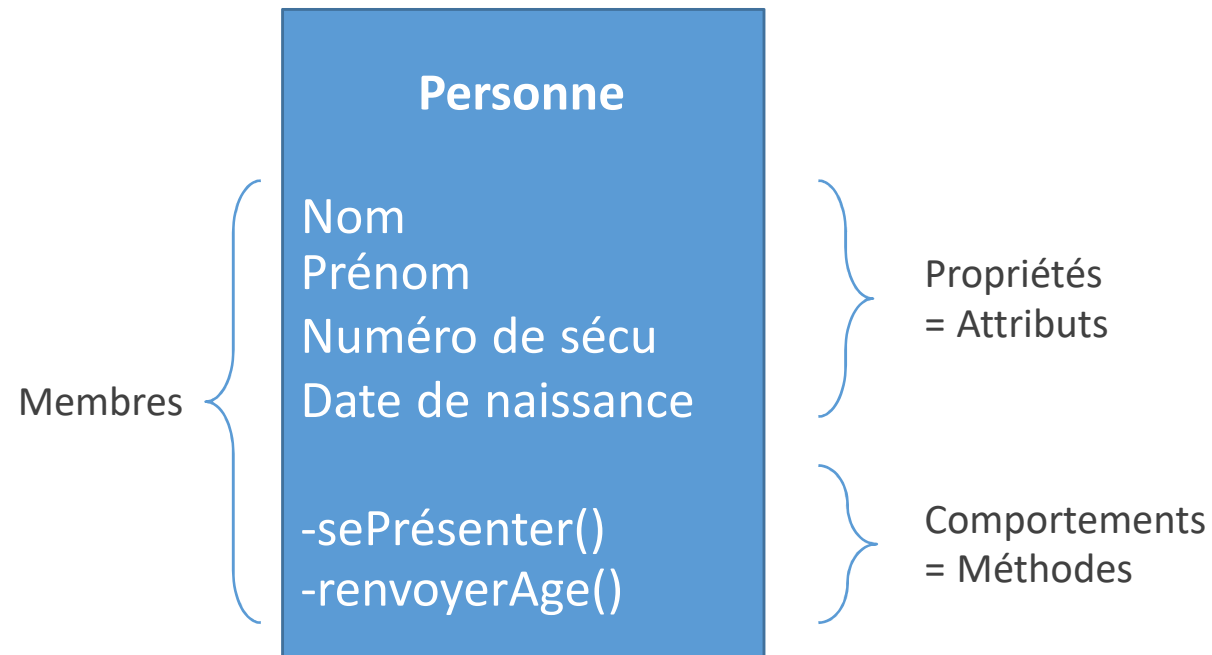
- Une classe dispose de membres qui lui sont propres : propriétés et comportements.
- Il s'agit de masquer la structure interne des données et fournir les fonctionnalités nécessaires sous forme de services.
- L'accès aux données (lecture/écriture) se fait via les méthodes mises à disposition.
(getters/setters)

Avantages

- Protection des données.
- Découplage vis-à-vis de la structure de données.

CONCEPTS ORIENTES OBJET

L'encapsulation



CONCEPTS ORIENTES OBJET

L'encapsulation & abstraction



Objet « boîte noire »

- Visibilité des services rendus par l'objet
- Fonctionnement interne non visible



Services rendus :

- Démarrer
- Accélérer
- Freiner
- Climatiser

CONCEPTS ORIENTES OBJET

L'encapsulation et abstraction



Personne

- Nom
- Prénom
- Numéro de sécu
- Date de naissance

- sePrésenter()
- calculateAge()
- getNom()
- setNom()
- getPrénom()
- getNuméroDeSécu()
- setNuméroDeSécu()
- getDateDeNaissance()

Accesseur (= getter) : Permet d'accéder en lecture à un attribut de la classe

Mutateur (= setter) : Permet de modifier la valeur d'un attribut de la classe

CONCEPTS ORIENTES OBJET

L'encapsulation



Quand définit-on les accesseurs/mutateurs pour un attribut ?

- Quand on doit pouvoir accéder/modifier l'attribut
 - Depuis l'extérieur de la classe
 - Après la création de l'objet

- Exemples dans la classe Adhérent :
 - Accesseur/mutateur pour le nom ?
 - Accesseur/mutateur pour le prénom ?
 - Accesseur/mutateur pour la date de naissance ?
 - Accesseur/mutateur pour le numéro de sécu ?

CONCEPTS ORIENTES OBJET

Exercice



Quelles sont les données des objets principaux des modèles métier à choisir?

Pour quelles données définirons nous

- Un accesseur ?
- Un mutateur ?

CONCEPTS ORIENTES OBJET

Héritage

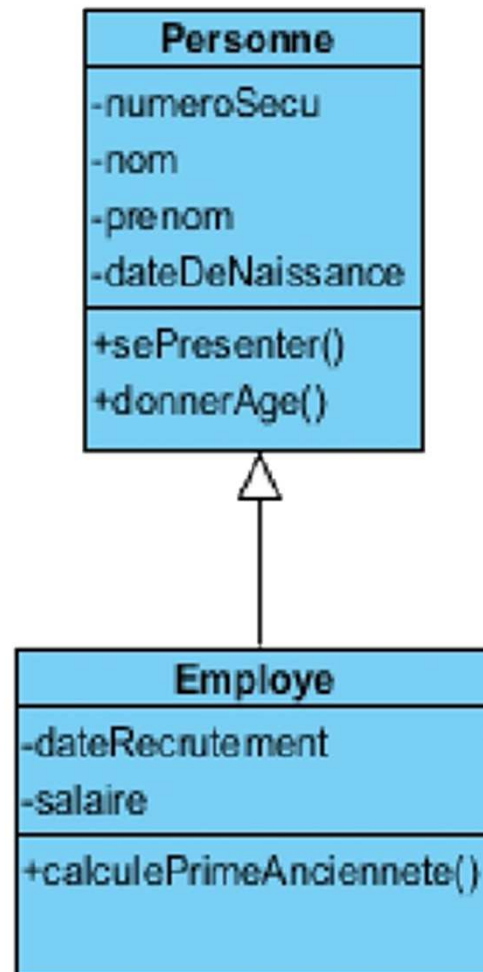


Relation de spécialisation/généralisation entre 2 classes :

- Une classe fille est une spécialisation de sa classe mère
- Une classe fille hérite des propriétés et des méthodes de sa classe mère
- Une classe fille dispose aussi de ses propres membres
 - /!\ La classe mère ne dispose pas des membres propres à sa classe fille !

CONCEPTS ORIENTES OBJET

Héritage



CONCEPTS ORIENTES OBJET

Exemple

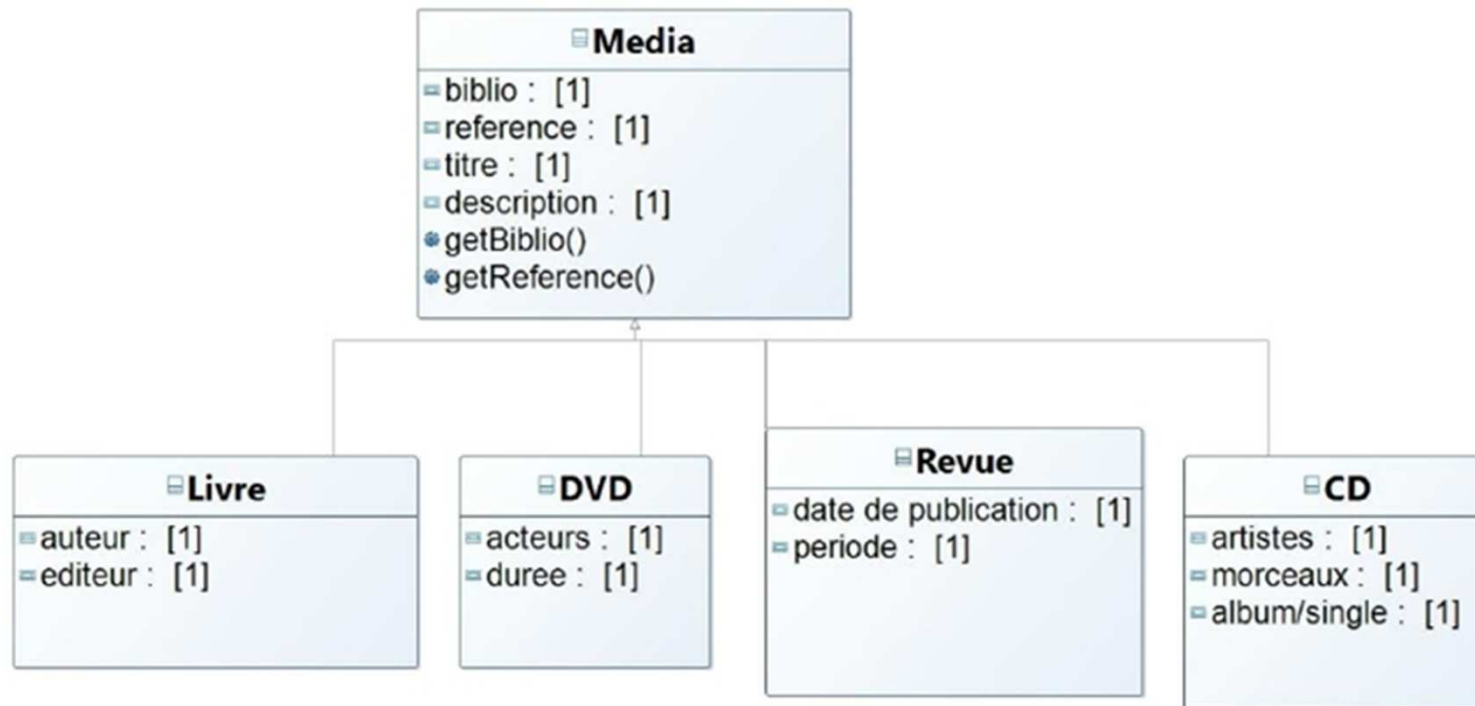


Cas de généralisation/spécialisation à modéliser pour un héritage dans l'ex de la médiathèque.

- Classe mère : Média
- Classes filles : Livre, Revue, CD, DVD

CONCEPTS ORIENTES OBJET

Exemple



CONCEPTS ORIENTES OBJET

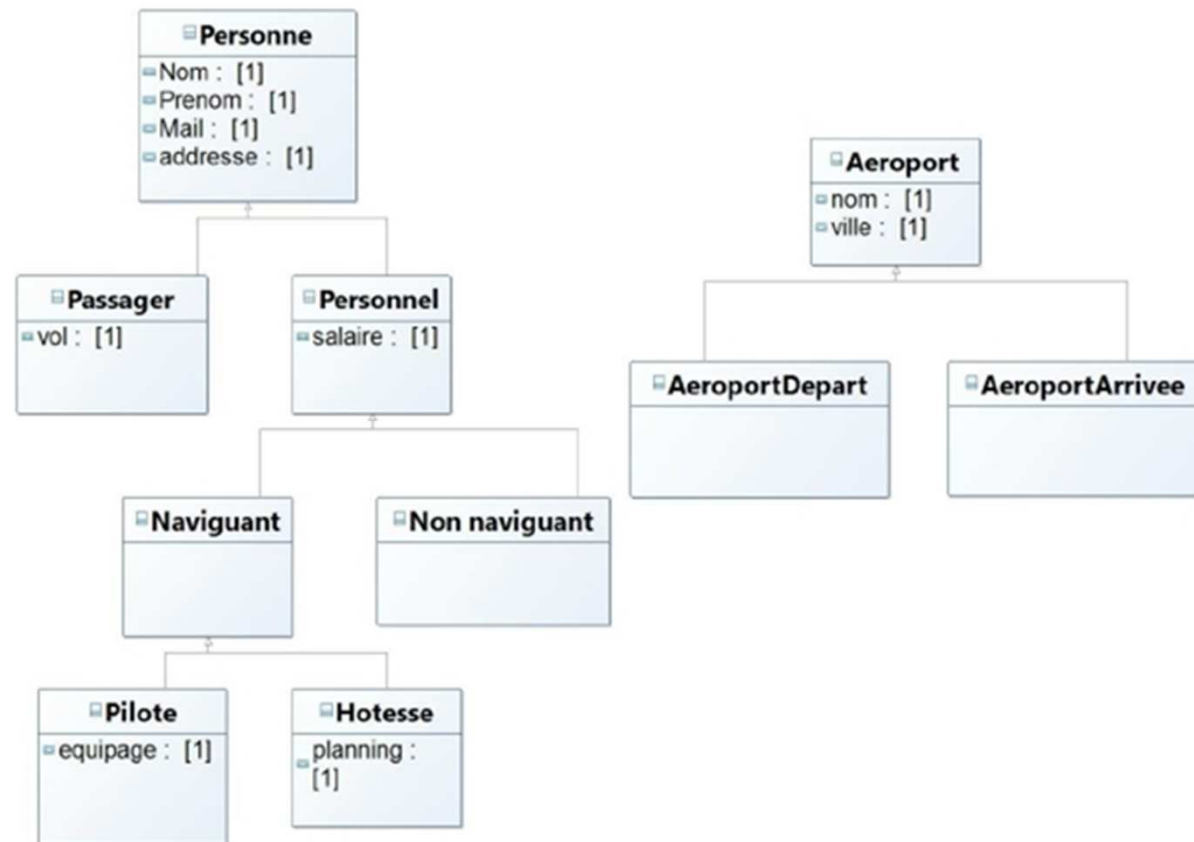
Exercice



**Cas de généralisation/spécialisation
à modéliser par un héritage dans la
thématique choisie.**

CONCEPTS ORIENTES OBJET

Exercice



CONCEPTS ORIENTES OBJET

Classes abstraites, l'héritage au service de l'abstraction



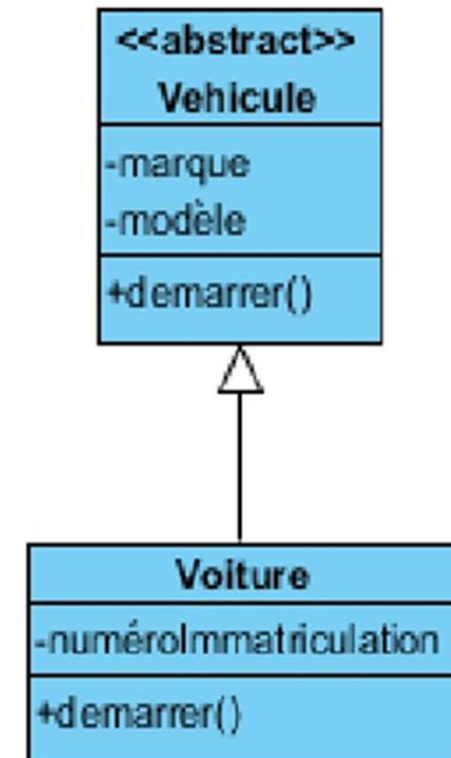
Classe « incomplète » ne pouvant pas être instanciée.

Base pour les classes dérivées : classes filles.

Contient des méthodes abstraites devant être définies dans les classes filles.

Peut également contenir des

- Méthodes concrètes
- Propriétés



CONCEPTS ORIENTES OBJET

Exemple



Dans l'exemple de la médiathèque la classe Média est abstraite.

- Ca n'a aucun sens d'instancier un « Média »
- La classe Média peut définir par contre les données (propres à tous les médias) :
 - Numéro de média
 - Date d'achat par la médiathèque
 - ...

CONCEPTS ORIENTES OBJET

Polymorphisme



Littéralement : faculté à prendre plusieurs formes.

En POO, deux définitions possibles :

- ~~Redéfinition (Overriding)~~ : Définition de plusieurs classes (plusieurs formes) définissant la même méthode
- ~~Surcharge (Overloading)~~ : Définition de plusieurs méthodes avec le même nom mais des paramètres différents
 - = Polymorphisme paramétrique, ou dit *ad hoc*

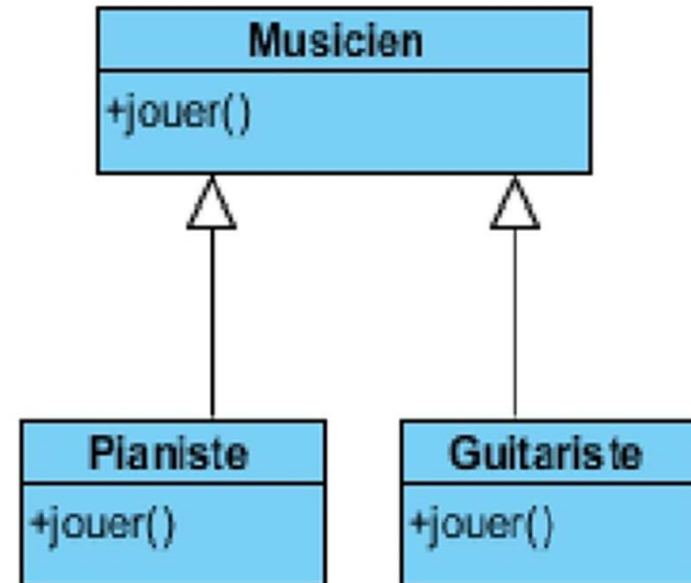
CONCEPTS ORIENTES OBJET

Polymorphisme



Avantages :

- Généricité
- Homogénéité



CONCEPTS ORIENTES OBJET

Polymorphisme



La redéfinition

- Implémentation dans une classe fille d'une méthode déjà définie dans la classe mère.
- Alternatives :
 - La méthode de la classe fille **remplace** la méthode de la classe mère.
 - La méthode de la classe fille fait appel à la méthode de la classe mère et la **complète**.

CONCEPTS ORIENTES OBJET

Polymorphisme

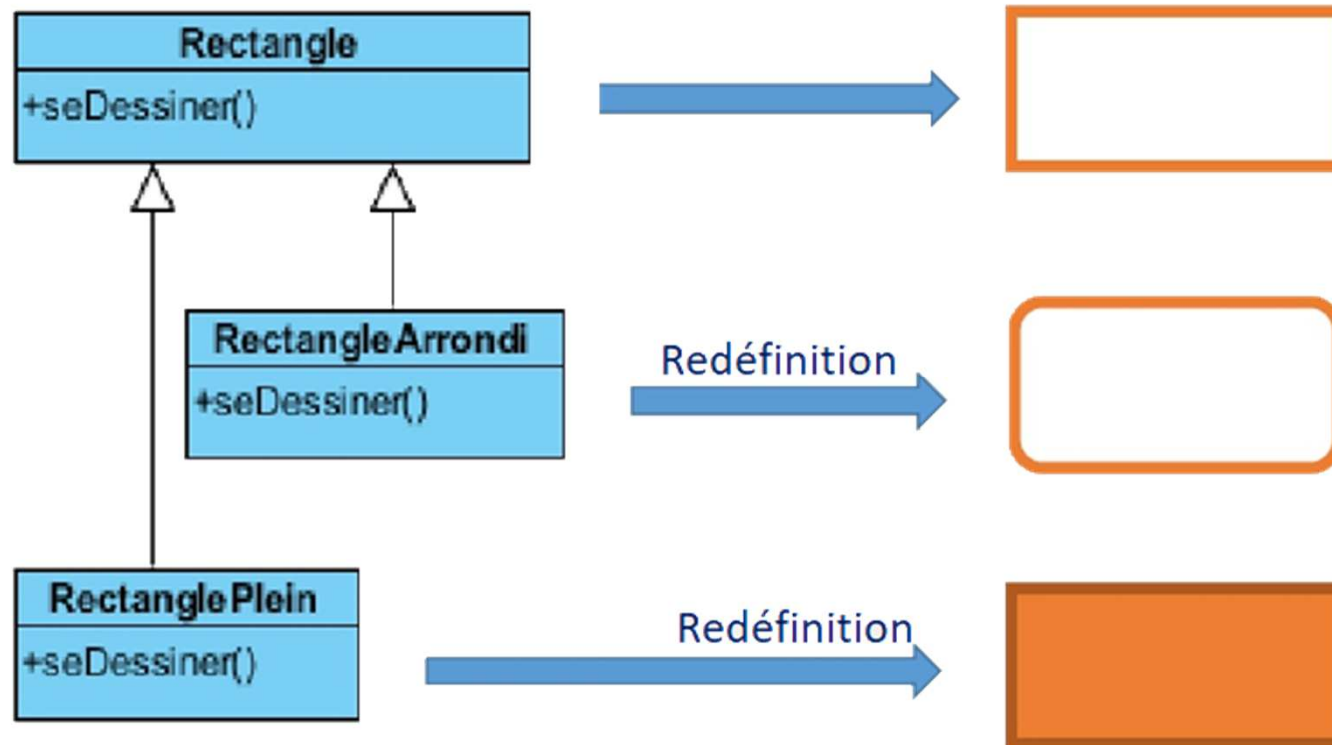


La surcharge : polymorphisme paramétrique.

- Implémentation d'une méthode déjà définie mais avec des paramètres différents (type ou nombre des paramètres)
- Exemple :
 - `int addition(int, int)`
 - `int addition(int, int, int)`
 - `float addition(float, float)`

CONCEPTS ORIENTES OBJET

Polymorphisme



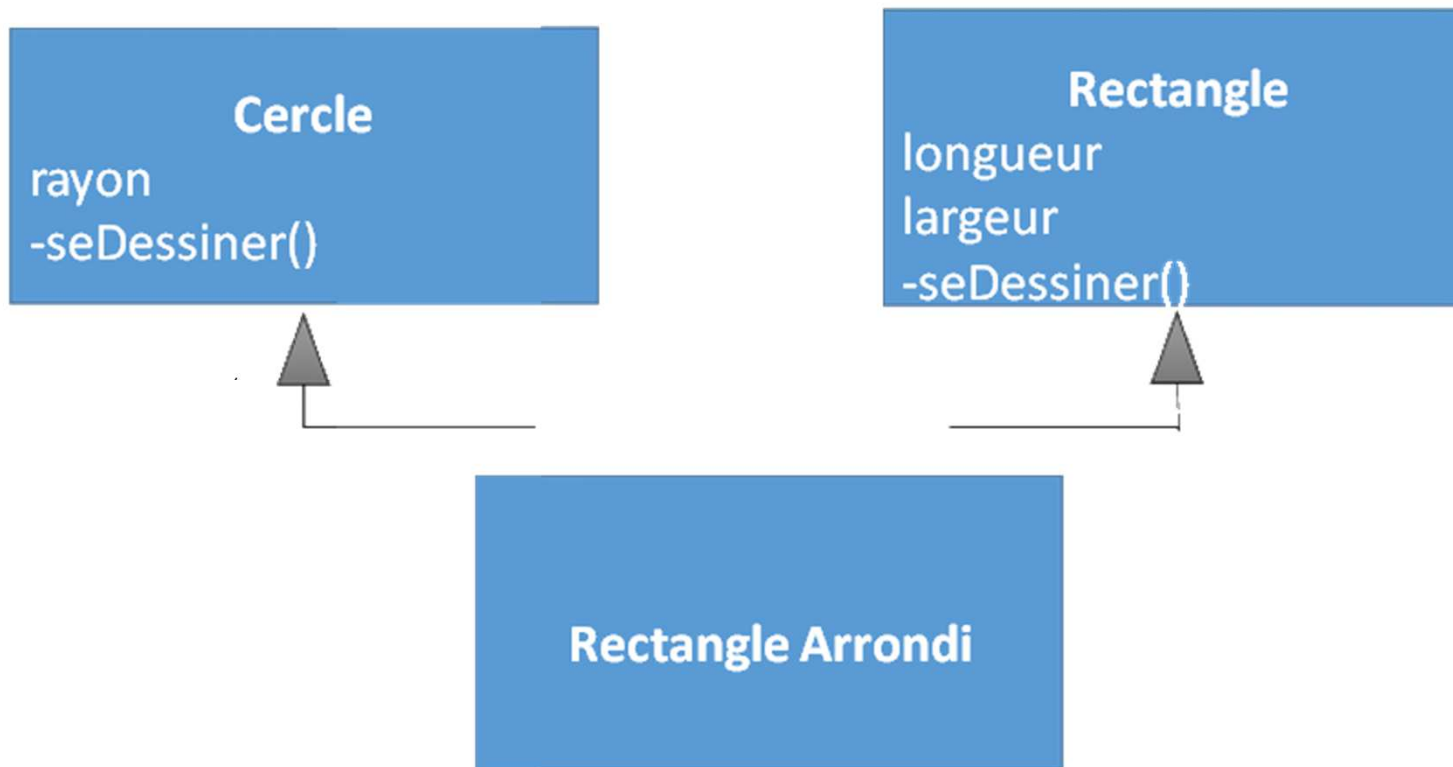
CONCEPTS ORIENTES OBJET

Héritage multiple (1/2)



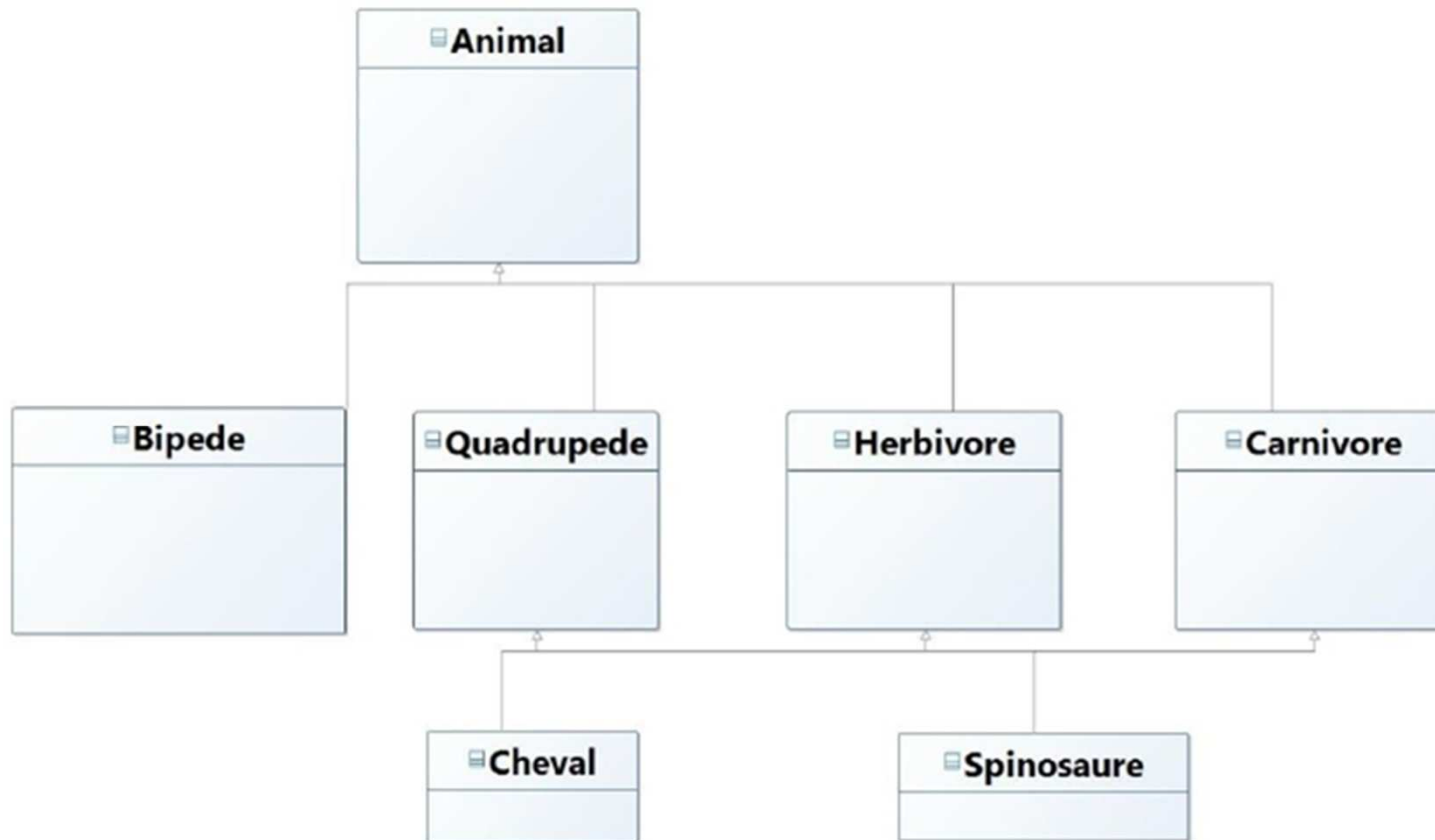
Certains langages permettent l'héritage multiple.

- Une classe peut hériter de 2 classes mères différentes.



CONCEPTS ORIENTES OBJET

Héritage multiple (2/2)



CONCEPTS ORIENTES OBJET

Polymorphisme



Avantages

- Factorisation
- Pas de duplication de code

Inconvénients

- /!\ Principe de généralisation
 - Un rectangle est il un cercle ?
- Héritage des méthodes
 - De quelle méthode « seDessiner() » ma classe fille hérite-t-elle ?

CONCEPTS ORIENTES OBJET

Thèmes d'exercices



Gestion de comptes bancaires en ligne.

Location de salles et bureaux.

Programme de fidélité.

Organisation de tournois sportifs.

Gestion d'une école.

Service RH d'une entreprise.