



# ENCAPSULATION

INTRODUCTION À L'APPROCHE ORIENTÉ OBJET

- Définition
- Attributs
- Méthodes
- Constructeurs
- Modificateur d'accès
- Accesseurs
- Implémentations UML
- Exercices

## ENCAPSULATION

Introduction à l'approche Orienté Objet

# DÉFINITION

L'encapsulation est un principe fondamental au sein de l'orienté objet.

Elle va nous permettre de déclarer les membres, de les protéger et de contrôler leur accessibilité afin de garantir l'intégrité des informations qu'ils contiennent.

Pour gérer l'accessibilité, nous devons donner à nos membres un **modificateur d'accès** qui limitera l'accès à ceux-ci.

# ATTRIBUTS



Une personne peut être caractérisée par :

- Numéro de registre national
- Nom
- Surnom
- Prénom
- Date de naissance
- Sexe
- Est en vie

Les attributs sont utilisés pour stocker des valeurs, on parle également de variables membres.

Ces valeurs sont souvent de types primitifs, c'est-à-dire de type :

- Entier
- Décimal
- Alphanumérique
- Date
- Booléenne
- ...

# MÉTHODES

les méthodes décrivent les fonctionnalités comportementales des objets.

Ces méthodes peuvent prendre des valeurs en entrée et/ou retourner une valeur.

Elles ont pour but principal de travailler avec les attributs pour les modifier ou produire des résultats sur base de ceux-ci.



Une personne peut :

- Parler
- Manger
- Dormir
- Pleurer
- Réfléchir

# CONSTRUCTEURS



Pour créer une personne il me faut :

- Son numéro de registre national
- Son nom
- Son prénom
- Sa date de naissance
- Son sexe

Les constructeurs sont des méthodes particulières qui ne sont utilisées qu'au moment de l'instanciation.

Ils servent le plus souvent à initialiser les variables membres à la création de nos objets.

En règle générale, ces méthodes portent le nom de la classe et n'ont pas de type de retour spécifié\*.



\*Peut varier en fonction des langages de programmation

# MODIFICATEUR D'ACCÈS

Les modificateurs d'accès sont utilisés pour limiter l'accès des membres par rapport à leur environnement.

Nous retrouvons le plus souvent 4 niveaux d'accès pour nos membres :

- Public
- Private
- Protected
- Package

Accessibilité	Signification
public	L'accès n'est pas limité.
protected	L'accès est restreint à la classe conteneur ou à ses types dérivés. (Notion d'héritage)
private	L'accès est restreint au type conteneur
package	L'accès est restreint au package courant



Peut varier en fonction des langages de programmation

# ACCESSEURS



Les accesseurs pour personnes

- Numéro de registre national
  - Getter public
  - Setter private
- Nom
  - Getter public
  - Setter private
- Surnom
  - Getter public
  - Setter public
- ...

Dans la bonne pratique, nos **attributs** se verront attribuer un modificateur d'accès « **private** ».

La raison est principalement fonctionnelle, une variable est un conteneur où nous pourrions lire la valeur ou l'affecter directement.

Cependant, nous n'avons aucun contrôle sur la valeur affectée ni sur la valeur retournée.

De plus, le modificateur d'accès de la variable gère à la fois l'accès à la récupération et l'accès à l'affectation de manière indissociable.

Pour résoudre cela, nous utiliserons des accesseurs qui ne sont rien d'autre que des méthodes dédiées à la gestion des attributs.



# ACCESSEURS



Les accesseurs pour personnes

- Numéro de registre national
  - Getter public
  - Setter private
- Nom
  - Getter public
  - Setter private
- Surnom
  - Getter public
  - Setter public
- ...

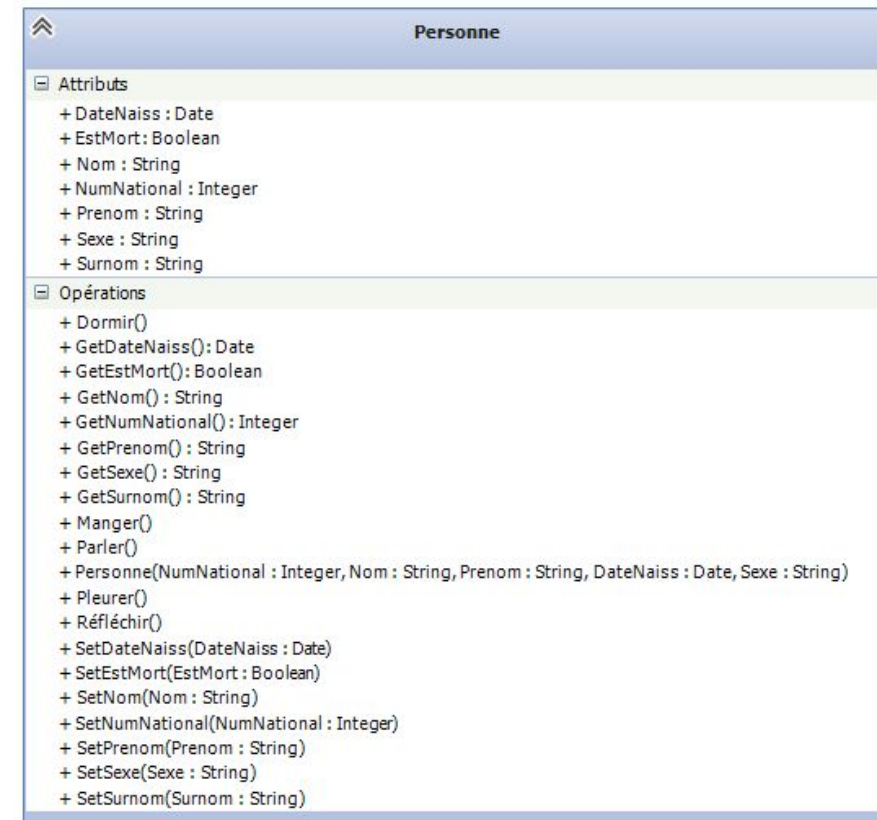
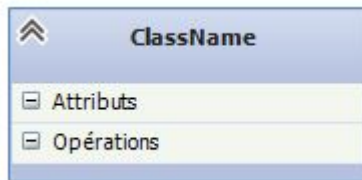
Nous retrouvons deux types d'accesseurs :

- **Getter** : Retourne la valeur de l'attribut.
- **Setter** : Affecte la valeur à l'attribut.

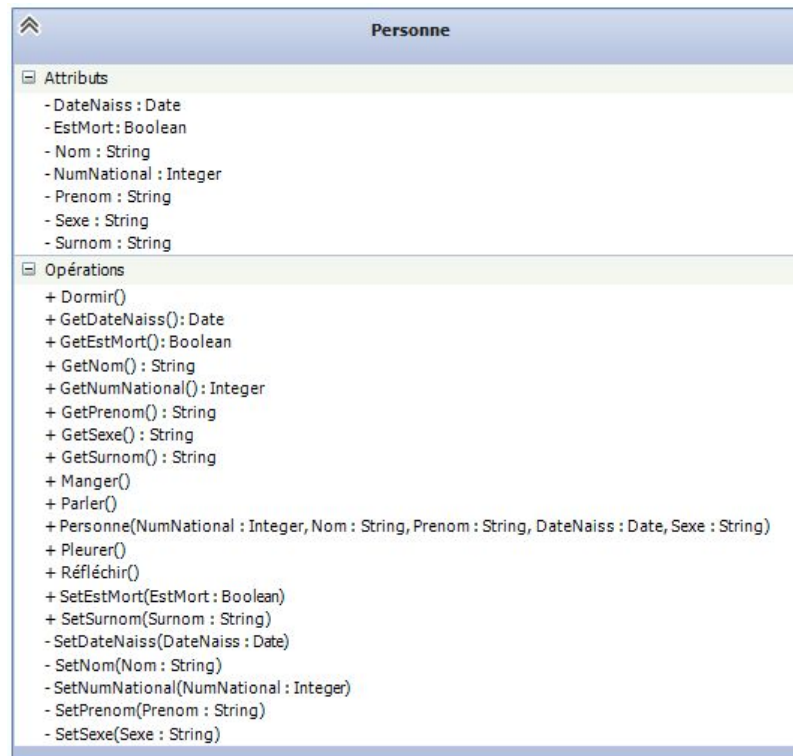
# IMPLÉMENTATION UML

Pour nous représenter nos classes et les relations qu'elles ont entre elles, nous pouvons utiliser les diagrammes de classes UML.

Ces derniers nous permettent de schématiser rapidement et simplement nos classes.



# IMPLÉMENTATION UML



Pour les modificateurs d'accès, nous devons l'indiquer avec un symbole qui précède le membre.

Accessibilité	Symbole
public	+ (Plus)
protected	# (Dièse)
private	- (Moins)
package	~ (Tilde)

# EXERCICES

- Déterminer sur papier les attributs, les accesseurs, les méthodes, le(s) constructeur(s) et les modificateurs d'accès pour chacun d'entre eux afin de représenter un compte bancaire type compte courant.
- Définir la classe à l'aide d'un diagramme de classe