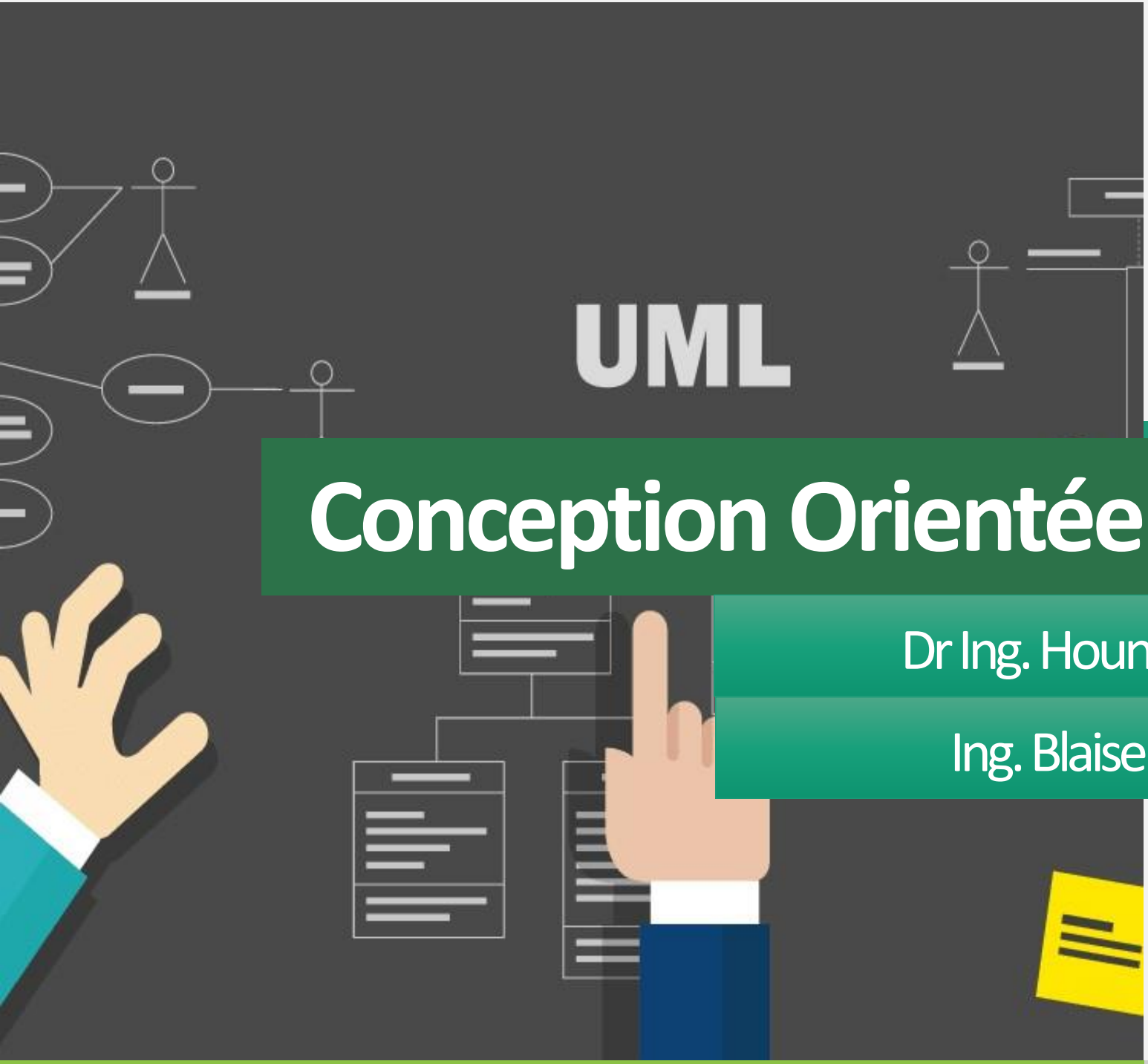


Conception Orientée Objet

Dr Ing. Houndji V. Ratheil

Ing. Blaise Arius Zogba

Plan du cours

- **CHAPITRE 1** : Le paradigme Orienté Objet
- **CHAPITRE 2** : Le diagramme de cas d'utilisation
- **CHAPITRE 3** : Le diagramme de classe
- **CHAPITRE 4** : Le diagramme d'objet
- **CHAPITRE 5** : Le diagramme de séquence
- **CHAPITRE 6** : Cas pratique : POO et ORM

CHAPITRE 6 :
POO & ORM

CHAPITRE 6 : POO & ORM

- ❑ Quelques rappels
- ❑ Les principes de la POO
- ❑ La PPO sans ORM
- ❑ Pourquoi les ORM ?
- ❑ Les exemples d’ORM

Quelques rappels

- La POO c’est simplement un moyen d'autoriser la syntaxe OO (par opposition au procédural), mais pas réellement un moyen de programmer efficacement avec des objets.
- La POO est un modèle de programmation qui repose sur le concept de classes et d'objets
- Elle vise à mettre en œuvre des entités du monde réel comme l'héritage ou le polymorphisme en programmation.
- L'objectif de la programmation orientée objet est de produire un code réutilisable bien conçu

Les principes de la POO

❖ La POO repose sur 4 principes :

- Le principe d'encapsulation
- Le principe d'abstraction
- Le principe d'héritage
- Le principe de polymorphisme

✓ **Le principe d'encapsulation**

Cette propriété garantit que les informations d'un objet sont cachées au monde extérieur en regroupant dans une classe les caractéristiques ou attributs qui ont un accès privé, et les comportements ou méthodes qui ont un accès public

Les principes de la POO

- ✓ **Le principe d'abstraction**

L'abstraction est basée sur l'utilisation de choses simples pour représenter la complexité. Les objets et les classes représentent le code sous-jacent, en cachant les détails complexes à l'utilisateur.

- ✓ **Le principe d'héritage**

L'héritage définit des relations hiérarchiques entre les classes, de sorte que les attributs et méthodes communs peuvent être réutilisés.

- ✓ **Le principe de polymorphisme**

Le polymorphisme consiste à concevoir des objets qui partagent des comportements, ce qui permet de traiter les objets de différentes manières. Il s'agit de la capacité de présenter la même interface pour différents moyens ou types de données sous-jacents.

Les fondamentaux de la PPO sans ORM

Voici les différentes étapes de la création d'un modèle de base de données (BDD) sans l'utilisation d'un Object-Relational Mapping (ORM) :

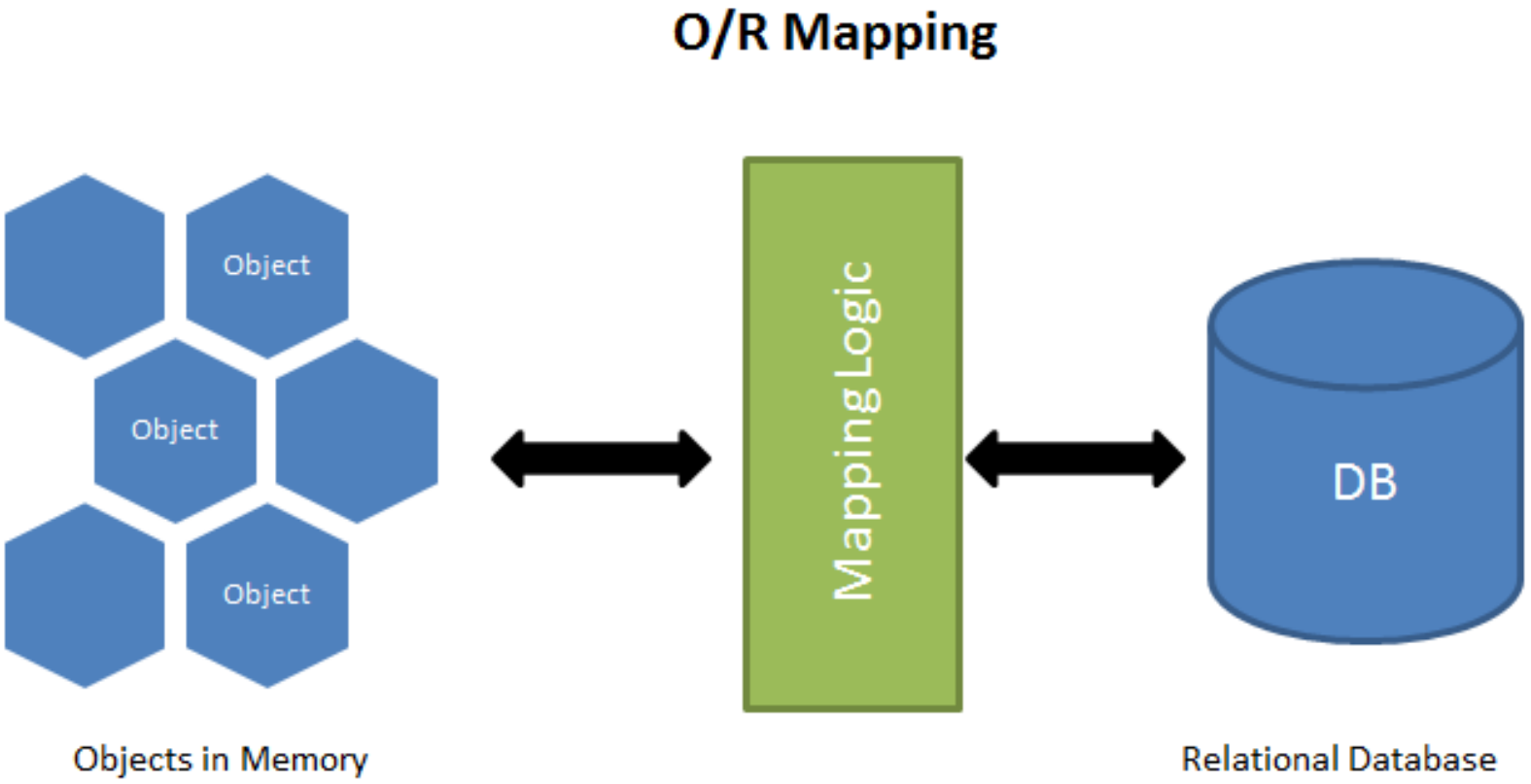
- Établir une connexion à la base de données
- Création des classes pour représenter les tables
- Implémenter les méthodes CRUD
- Gérer les relations entre les tables
- Sécurité et optimisation des requêtes
- Tests et débogage.

Pourquoi les ORM ?

L'un des défis de l'utilisation des langages et des bases de données de programmation orientée objet (POO) est la complexité de l'alignement du code de programmation avec les structures de la base de données.

- Un **ORM** (Object-Relational Mapper) **fournit une couche** orientée objet entre les bases de données relationnelles et les langages de programmation orientés objet **sans avoir à écrire de requêtes SQL**. Il standardise les interfaces en réduisant le passe-partout et **en accélérant le temps de développement.**

Pourquoi les ORM ?



Pourquoi les ORM ?

- Le mappage décrit la relation entre un objet et les données sans savoir comment les données sont structurées. Le modèle peut ensuite être utilisé pour connecter l'application au code SQL nécessaire à la gestion des activités de données
- Il fait abstraction du système de base de données afin que le passage de MySQL à PostgreSQL, ou à la version que vous préférez, soit facile
- En fonction de l'ORM, il existe des fonctionnalités avancées prêtes à l'emploi, telles que la prise en charge des transactions, la mise en commun des connexions, les migrations, les graines, les flux et toutes sortes d'autres avantages.

Les exemples d'ORM

- Java
  - Java Persistence API
  - Hibernate
  - Spring Data
  - Java Data Objects
- NodeJS
  - Prisma
  - Mongoose
- PHP
  - Doctrine
  - Eloquent
- Python
  - SQLAlchemy
  - Peewee
  - SQLObject
  - Django

Merci pour votre attention

Bibliographie

- Conception Orientée Objet, Amosse Edouard, UNICE, Faculte des sciences, http://miageprojet2.unice.fr/@api/deki/files/2677/Cours_Magistral_Seance_1.pdf
- Bases de la conception orientée objet, Concepts Objet – Diagrammes de classes, Petru Valicov, Institut Universitaire de Technologie, 2017-2018
- Qu'est-ce qu'un ORM et pourquoi devriez-vous l'utiliser, WETIC consulté le 30/01/24

https://wetic.be/quest-ce-quun-orm-et-pourquoi-devriez-vous-lutiliser

- Présentation des 14 types et exemples de diagrammes UML, GitMind, consulté le 30/01/24

https://gitmind.com/fr/types-diagrammes-uml.html