

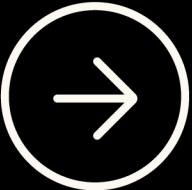
# PROGRAMMATION ORIENTÉE OBJET (POO)

Présenté par

Mohammed-Ali Rostom &  
M'hamed Ouhanane



# SOMMAIRE



1 Introduction

2 Qu'est-ce que la POO ?

3 POO vs Procedural

4 Concepts principaux de la POO

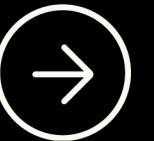
5 Classe vs objet

6 Exemple en PHP

7 Les Accesseurs (Access Modifiers) en POO

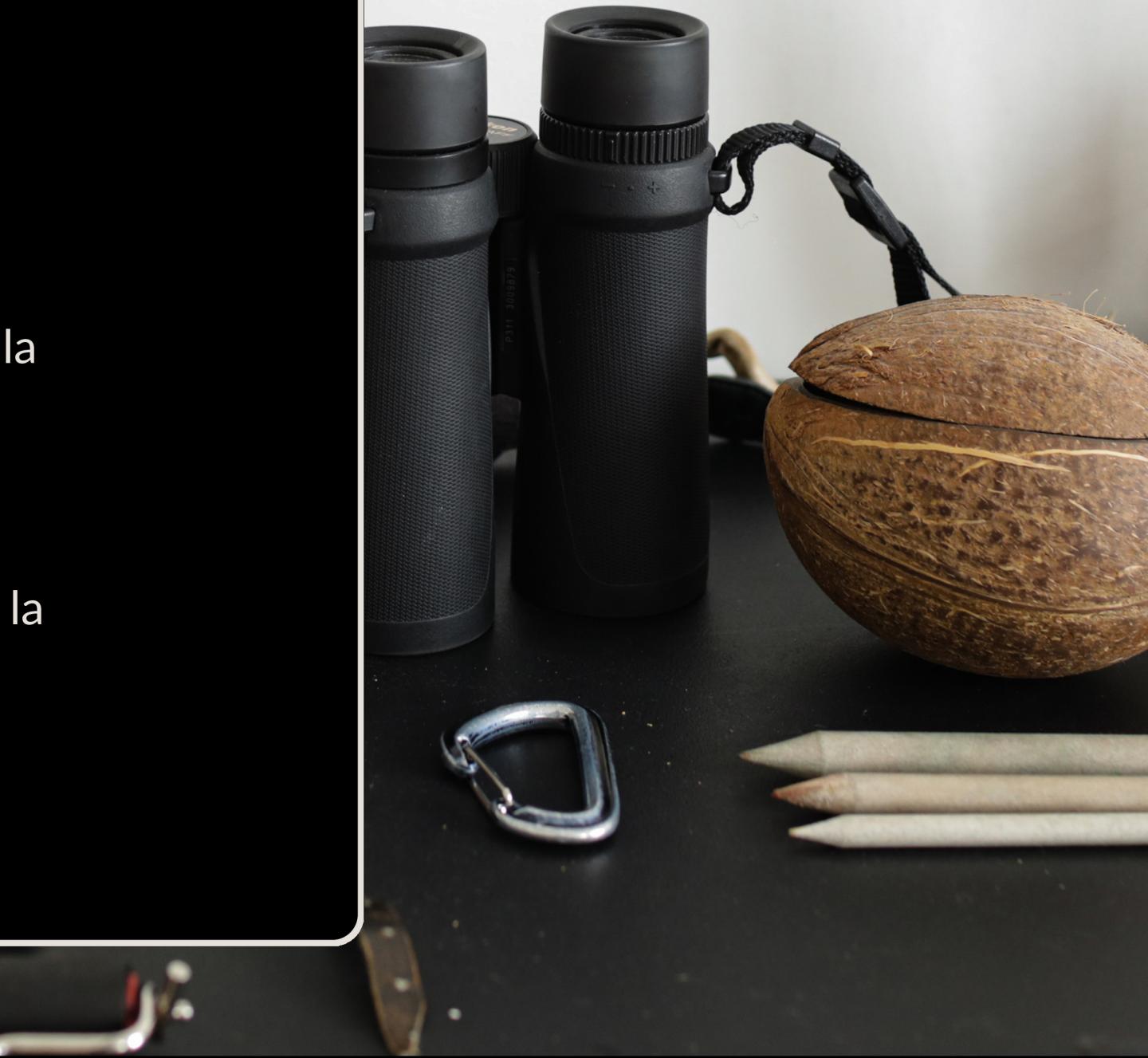
8 Les Avantages du POO

9 Conclusion



# INTRODUCTION

La Programmation Orientée Objet (POO) est une manière d'organiser le code dans les logiciels. Au lieu d'écrire des instructions qui se suivent les unes après les autres, comme dans la programmation traditionnelle (procédurale), la POO consiste à organiser le code en objets. Un objet est une sorte de "boîte" qui contient des données (par exemple, la couleur, le modèle d'une voiture) et des actions qu'on peut lui faire faire (comme démarrer la voiture, freiner, etc.).



# QU'EST-CE QUE LA POO ?

La POO est une méthode de programmation qui permet de modéliser des concepts du monde réel à travers des objets. Un objet est une instance d'une classe, et une classe définit les caractéristiques et les comportements des objets. Les objets peuvent avoir des attributs (ou propriétés) et des méthodes (ou fonctions) qui déterminent comment ils se comportent et interagissent entre eux.



# POO VS PROGRAMMATION PROCÉDURALE

## PROGRAMMATION PROCÉDURALE

Elle est axée sur l'exécution de fonctions ou de procédures qui manipulent des données. Le code est généralement linéaire, et les données sont manipulées de manière globale.

## PROGRAMMATION ORIENTÉE OBJET

Elle met l'accent sur la création d'objets et la définition de leur comportement et de leur état. Elle est plus flexible et modulaire, permettant de mieux organiser et maintenir le code.

# CONCEPTS PRINCIPAUX DE LA POO

## ENCAPSULATION

Ce principe consiste à regrouper les données (attributs) et les méthodes qui les manipulent dans une même classe, tout en contrôlant leur accès grâce aux modificateurs d'accès (private, protected, public).

## HÉRITAGE

Il permet à une classe d'hériter des caractéristiques et des comportements d'une autre classe. Cela favorise la réutilisation du code.

## POLYMORPHISME

C'est la capacité d'un objet à prendre plusieurs formes, permettant d'utiliser une méthode de manière générique pour différentes classes.

## ABSTRACTION

L'abstraction permet de ne se concentrer que sur l'essentiel d'un objet, en cachant les détails d'implémentation complexes.

# CLASSE VS OBJET



## Classe

Il s'agit d'un plan ou d'une structure qui définit les propriétés (attributs) et les comportements (méthodes) des objets. Une classe est une sorte de modèle ou de schéma.

## Objet

Un objet est une instance concrète d'une classe. Par exemple, si la classe représente une voiture, un objet serait une voiture spécifique avec des caractéristiques propres (couleur, marque, etc.).



# LES ACCESSEURS (ACCESS MODIFIERS) EN POO

Les modificateurs d'accès (ou access modifiers) sont des mécanismes utilisés en POO pour définir la visibilité et l'accessibilité des attributs et des méthodes d'une classe. Ils permettent de contrôler qui peut accéder à quelles parties d'un objet, ce qui améliore l'encapsulation et la sécurité du code.

## Public (public)

Les membres publics d'une classe (attributs ou méthodes) sont accessibles de n'importe où, que ce soit à l'intérieur de la classe, dans d'autres classes ou à l'extérieur du programme.

## Privé (private)

Les membres privés sont accessibles uniquement à l'intérieur de la classe où ils sont définis. Ils ne peuvent pas être accédés ou modifiés depuis l'extérieur de la classe, même par des classes dérivées.

## Protégé (protected)

Les membres protégés sont accessibles dans la classe où ils sont définis ainsi que dans les classes qui en héritent (les sous-classes), mais ne sont pas accessibles depuis l'extérieur.



# AVANTAGES DU POO

---

## Modularité

Le code est divisé en objets indépendants, ce qui facilite la gestion.

## Extensibilité

Il est facile d'ajouter de nouvelles fonctionnalités sans perturber le code existant.

## Réutilisation du code

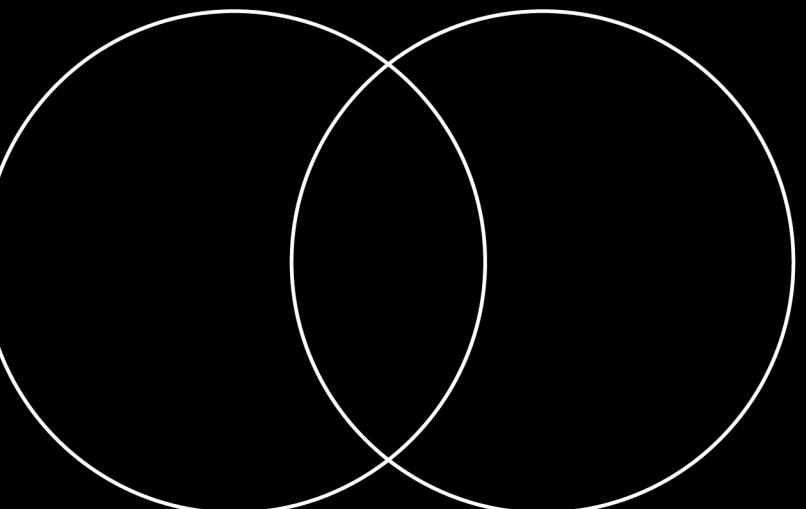
L'héritage permet de réutiliser le code sans le répéter.

## Maintenance

Le code est plus facile à comprendre et à corriger.

## Sécurité

Les détails internes des objets sont cachés, protégeant ainsi les données.



# CONCLUSION

---

La Programmation Orientée Objet (POO) est une méthode de programmation qui organise le code autour d'objets, rendant les applications plus faciles à comprendre, maintenir et étendre. En utilisant des concepts comme l'encapsulation, l'héritage, le polymorphisme et l'abstraction, la POO aide à structurer le code de manière plus claire et modulaire. Cela permet de réutiliser du code, d'assurer la sécurité des données et de rendre les applications plus flexibles et évolutives.

**MERCI**  
pour votre temps et votre attention

