

## **Module 3**

### **Les 4 Piliers de la POO**

---

Objectif: comprendre pourquoi la POO structure le code au lieu de seulement l'organiser visuellement.

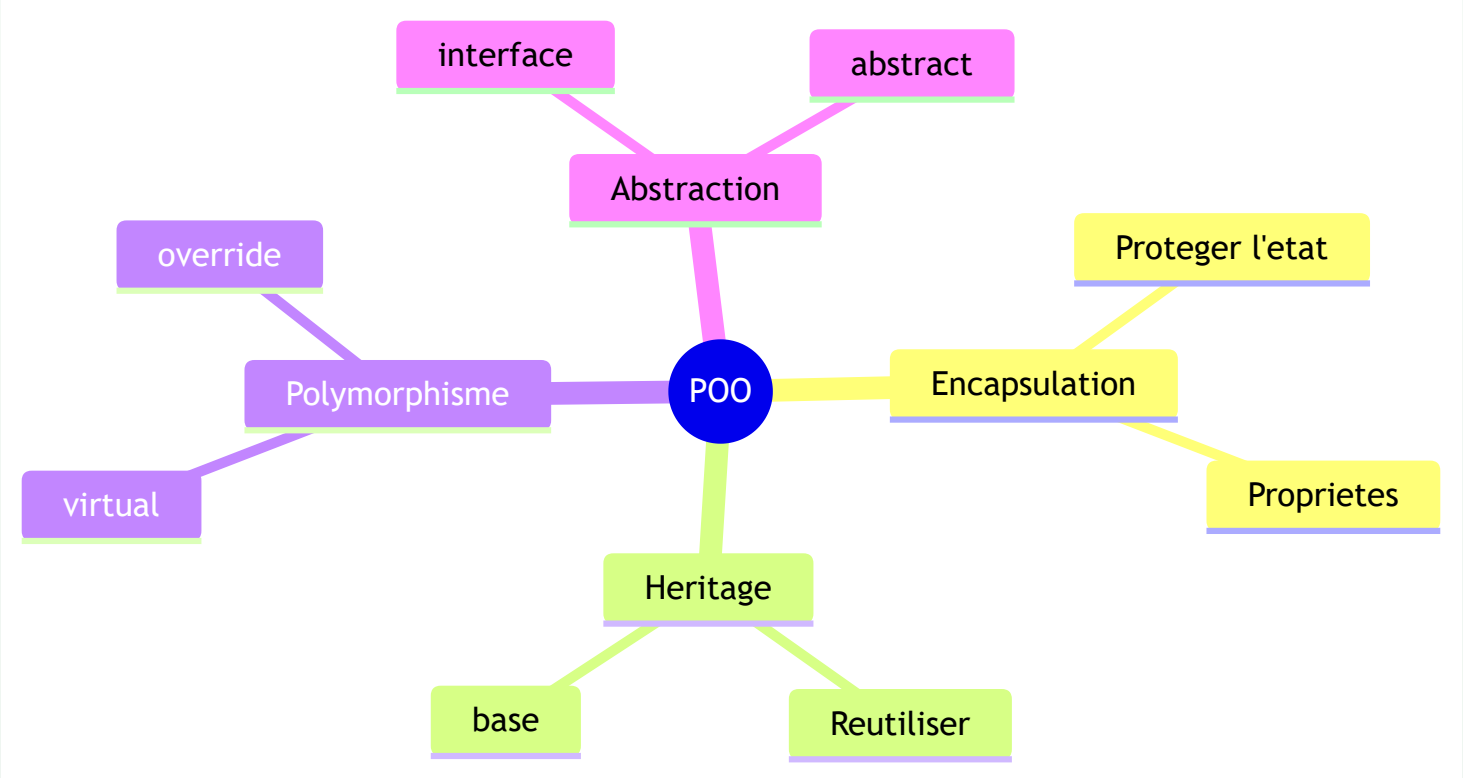


## Changelog – V0.0.2

---

- Deck dédié a encapsulation, héritage, polymorphisme et abstraction.
- Ajout d'un schema de synthese.

# Vue d'ensemble



## Encapsulation

---

```
public class CompteBancaire
{
    public decimal Solde { get; private set; }

    public void Deposer(decimal montant)
    {
        if (montant <= 0) return;
        Solde += montant;
    }
}
```

- On protège l'état.
- On force des règles métier dans la classe.

## Héritage

---

```
public class ProduitPerissable : Produit
{
    public DateTime DateLimite { get; }

    public ProduitPerissable(string nom, DateTime dateLimite) : base(nom)
    {
        DateLimite = dateLimite;
    }
}
```

- `:` déclare l'héritage.
- `base(...)` appelle le parent.

## Polymorphisme

---

```
public class Produit
{
    public virtual string AfficherType() => "Produit standard";
}

public class ProduitElectronique : Produit
{
    public override string AfficherType() => "Produit electronique";
}
```

- Une reference parent manipule plusieurs objets enfants.

## Abstraction

---

```
public interface IRemisable
{
    double CalculerRemise();
}

public abstract class Produit
{
    public abstract string AfficherType();
}
```

- Interface: ce qu'un objet sait faire.
- Classe abstraite: base commune incomplète.

## Checklist de sortie

---

- Savoir reconnaître les 4 piliers dans du code concret.
- Savoir choisir entre interface, classe et classe abstraite.