

P1C4 - Organisez votre code avec les bases de la programmation orientée objet

Code final :

```
public class Dish {
    private String name;
    private double price;
    private boolean available;

    public Dish(String name, double price, boolean available) {
        this.name = name;
        this.price = price;
        this.available = available;
    }

    public String describe() {
        if (available) {
            return name + " (" + price + " €) - disponible";
        } else {
            return name + " (" + price + " €) - indisponible";
        }
    }
}

public class Main {
    public static void main(String[] args) {
        Dish pizza = new Dish("Pizza", 12.0, true);
        Dish tiramisu = new Dish("Tiramisu", 6.0, false);

        System.out.println(pizza.describe());
        System.out.println(tiramisu.describe());
    }
}
```

Sortie attendue :

```
Pizza (12.0 €) - disponible
Tiramisu (6.0 €) - indisponible
```

Les attributs `private` sont accessibles à l'intérieur de la classe — la méthode `describe()` peut donc lire `name`, `price` et `available` directement, sans passer par un getter. Le constructeur utilise `this.` pour lever l'ambiguïté entre paramètre et attribut. Chaque objet a ses propres valeurs, mais partage la même structure et les mêmes méthodes.



Dans le projet, chaque entité métier (utilisateur, transaction, catégorie...) sera modélisée par une classe similaire. Le trio attributs privés + constructeur + méthode qui utilise les attributs est le geste fondateur de toute POO Java.