

P1C4 - Implémentez la programmation orientée objet en Python

Code final :

```
class Dish:
    def __init__(self, name, price, available):
        self.name = name
        self.price = price
        self.available = available

    def describe(self):
        if self.available:
            return f"{self.name} ({self.price} €) - disponible"
        else:
            return f"{self.name} ({self.price} €) - indisponible"

pizza = Dish("Pizza", 12.0, True)
tiramisu = Dish("Tiramisu", 6.0, False)

print(pizza.describe())
print(tiramisu.describe())
```

Sortie attendue :

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

La méthode `describe()` lit les attributs de l'objet avec `self.name`, `self.price` et `self.available`. Chaque objet a ses propres valeurs mais partage la même structure et les mêmes méthodes. La condition `if self.available:` utilise le booléen directement : pas besoin d'écrire `== True`.



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