```python
# ----------------------------
# Méthode 1 : Héritage
# ----------------------------
class A:
    def __init__(self, x):
        self.x = x  # Attribut de A

    def show(self):
        print(f"[Héritage] Valeur de x dans A : {self.x}")

class B(A):  # B hérite de A
    def __init__(self, x, y):
        super().__init__(x)  # Appel du constructeur de A
        self.y = y           # Attribut spécifique à B

    def display(self):
        print(f"[Héritage] Valeur de y dans B : {self.y}")
        self.show()  # Utilisation de la méthode de A

b = B(10, 20)
b.display()


# ----------------------------
# Méthode 2 : Composition
# ----------------------------
class C:
    def __init__(self, x):
        self.x = x

    def show(self):
        print(f"[Composition] Valeur de x dans C : {self.x}")

class D:
    def __init__(self, c_instance, y):
        self.c = c_instance  # Stocke une instance de C
        self.y = y

    def display(self):
        print(f"[Composition] Valeur de y dans D : {self.y}")
        self.c.show()  # Utilise la méthode de l'instance de C

c = C(30)
d = D(c, 40)
```

```python
d.display()


# ----------------------------
# Méthode 3 : Passage direct d'une instance
# ----------------------------
class E:
    def __init__(self, x, y):
        self.x = x
        self.y = y

    def display(self, c_instance):
        print(f"[Passage direct] Valeur de y dans E : {self.y}")
        c_instance.show()  # Utilise l'instance de C passée en argument

e = E(50, 60)
e.display(c)  # On passe directement l'instance de C
```