

# Correction - Héritage

Avec la bibliothèque math :

```
import math

class Forme:
    def __init__(self, couleur):
        self.couleur = couleur

    def afficher_info(self):
        print(f"Couleur de la forme : {self.couleur}")

class Rectangle(Forme):
    def __init__(self, couleur, longueur, largeur):
        super().__init__(couleur)
        self.longueur = longueur
        self.largeur = largeur

    def calculer_surface(self):
        return self.longueur * self.largeur

class Cercle(Forme):
    def __init__(self, couleur, rayon):
        super().__init__(couleur)
        self.rayon = rayon

    def calculer_surface(self):
        return math.pi * self.rayon ** 2
```

```

# Création d'instances
rectangle = Rectangle("rouge", 5, 3)
cercle = Cercle("bleu", 4)

# Affichage des informations et calcul de surface
rectangle.afficher_info()
print("Surface du rectangle :", rectangle.calculer_surface())

cercle.afficher_info()
print("Surface du cercle :", cercle.calculer_surface())

```

Sans la bibliothèque Math :

```

class Forme:
    def __init__(self, couleur):
        self.couleur = couleur

    def afficher_info(self):
        print(f"Couleur de la forme : {self.couleur}")

class Rectangle(Forme):
    def __init__(self, couleur, longueur, largeur):
        super().__init__(couleur)
        self.longueur = longueur
        self.largeur = largeur

    def calculer_surface(self):
        return self.longueur * self.largeur

class Cercle(Forme):

```

```

def __init__(self, couleur, rayon):
    super().__init__(couleur)
    self.rayon = rayon

def calculer_surface(self):
    return 3.1416 * self.rayon ** 2 # Approximation de pi

# Création d'instances
rectangle = Rectangle("rouge", 5, 3)
cercle = Cercle("bleu", 4)

# Affichage des informations et calcul de surface
rectangle.afficher_info()
print("Surface du rectangle :", rectangle.calculer_surface())

cercle.afficher_info()
print("Surface du cercle :", cercle.calculer_surface())

```