

Polimorfismo

Letícia Vidal

Definimos Polimorfismo

O Polimorfismo tem como princípio a partir do qual as classes derivadas de uma única classe base são capazes de invocar os métodos que, embora apresentem a mesma assinatura, comportam-se de maneira diferente para cada uma das classes derivadas.

Explicando a definição

Explicando a sua definição

Quando você diz:

“métodos com a mesma assinatura, mas comportamentos diferentes”

Significa que:

- O **nome do método é o mesmo**
- Os **parâmetros são iguais**
- Mas o **resultado (comportamento) muda** conforme a classe

```
class Forma:
    def calcular_area(self):
        pass

class Quadrado(Forma):
    def __init__(self, lado):
        self.lado = lado

    def calcular_area(self):
        return self.lado * self.lado
```

```
class Circulo(Forma):
    def __init__(self, raio):
        self.raio = raio

    def calcular_area(self):
        return 3.14 * self.raio ** 2

# Lista de objetos diferentes
formas = [Quadrado(4), Circulo(3)]

for forma in formas:
    print(forma.calcular_area())
```

for forma in formas:

```
print(forma.calcular_area())
```

Aqui está o ponto principal:

- O código chama **o mesmo método**
- Mas o Python decide qual executar:
 - Se for **Quadrado** → usa a fórmula do quadrado
 - Se for **Circulo** → usa a fórmula do círculo

Isso é **polimorfismo**

Polimorfismo =

“Um mesmo método pode ter vários comportamentos dependendo do objeto”

```
from abc import ABC, abstractmethod
```

```
class Forma(ABC):
```

```
    @abstractmethod
```

```
    def calcular_area(self):
```

```
        pass
```

```
class Quadrado(Forma):  
    def __init__(self, lado):  
        self.lado = lado  
  
    def calcular_area(self):  
        return self.lado ** 2
```

```
class Circulo(Forma):  
    def __init__(self, raio):  
        self.raio = raio  
  
    def calcular_area(self):  
        return 3.14 * self.raio ** 2  
  
# Testando  
formas = [Quadrado(5), Circulo(2.5)]  
  
for forma in formas:  
    print(forma.calcular_area())
```