Polymorphism

Pankaj Chouhan
www.codeswithpankaj.com

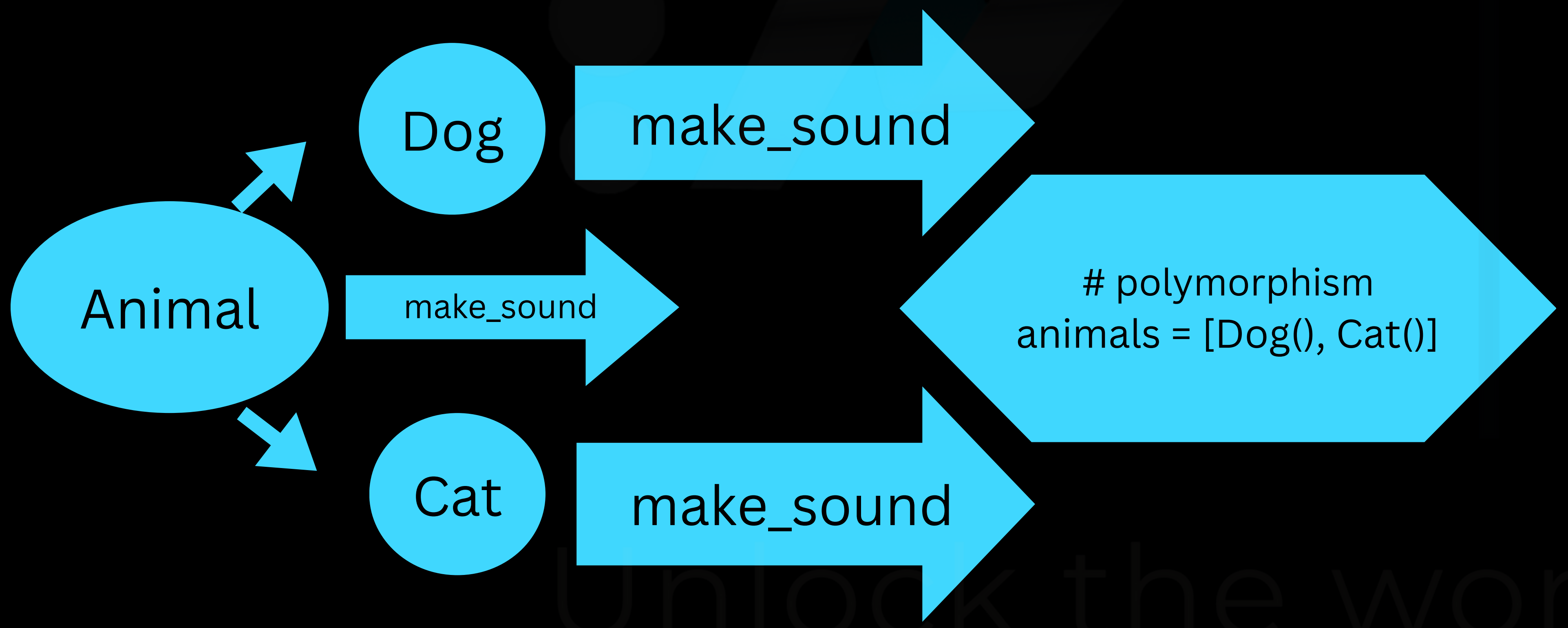
Polymorphism means "many forms" in Greek. In Python, polymorphism allows objects of different classes to be treated as objects of a common class. It helps in writing flexible and reusable code.

Types of Polymorphism in Python

1. Method Overriding (Runtime Polymorphism)
2. Method Overloading (Python does not support true method overloading but can be achieved using default arguments)
3. Operator Overloading

Method Overriding (Runtime Polymorphism)

When a child class provides a specific implementation of a method that is already defined in its parent class.



Example

```
class Animal:
    def make_sound(self):
        print("Animal makes a sound")

class Dog(Animal):
    def make_sound(self):
        # Overriding parent method
        print("Dog barks")

class Cat(Animal):
    def make_sound(self):
        # Overriding parent method
        print("Cat meows")

# Using polymorphism
animals = [Dog(), Cat()]
for animal in animals:
    animal.make_sound()

# Output:
# Dog barks
# Cat meows
```

Pankaj Chouhan

www.codeswithpankaj.com

Method Overloading

(Not Directly Supported in Python)

Python does not support method overloading like Java/C++, but it can be done using default arguments.

Example

```
class MathOperations:
    def add(self, a, b, c=0):
        # Default argument c
        return a + b + c

obj = MathOperations()
print(obj.add(2, 3))
# Output: 5
print(obj.add(2, 3, 4))
# Output: 9
```

Pankaj Chouhan

Operator Overloading

Python allows operators like +, -, * to work differently for different data types by defining special methods like __add__(), __sub__(), etc.

Example

```
class Number:
    def __init__(self, value):
        self.value = value

    def __add__(self, other):
        # Overloading '+' operator
        return Number(self.value +
other.value)

num1 = Number(5)
num2 = Number(10)
result = num1 + num2
 # Calls __add__() method
print(result.value)
# Output: 15
```

Pankaj Chouhan
www.codeswithpankaj.com

✅ Polymorphism

allows the same method name to have different behaviors.

✅ Method overriding

lets child classes redefine a parent class method.

✅ Method overloading

can be simulated using default arguments.

✅ Operator overloading

lets us use operators with custom classes.

READ MORE