Ejemplo Aplicación

August 30, 2023

Supongamos que estás construyendo un sistema para calcular el área y el perímetro de diferentes tipos de figuras geométricas. Utilizarás clases abstractas para definir una estructura base y luego implementarás clases concretas para diferentes tipos de figuras. Finalmente, utilizarás el polimorfismo para tratar todas las figuras de manera uniforme.

[ ]:
```python
from abc import ABC, abstractmethod

class FiguraGeometrica(ABC):
    def __init__(self):
        pass

    @abstractmethod
    def calcular_area(self):
        pass

    @abstractmethod
    def calcular_perimetro(self):
        pass

class Cuadrado(FiguraGeometrica):
    def __init__(self, lado):
        self.lado = lado

    def calcular_area(self):
        return self.lado ** 2

    def calcular_perimetro(self):
        return 4 * self.lado

class Circulo(FiguraGeometrica):
    def __init__(self, radio):
        self.radio = radio

    def calcular_area(self):
        return 3.14159 * self.radio ** 2

    def calcular_perimetro(self):
        return 2 * 3.14159 * self.radio
```

```python

# Crear instancias de las clases concretas
cuadrado = Cuadrado(5)
circulo = Circulo(3)

# Función para imprimir detalles de una figura
def imprimir_detalles(figura):
    print(f"Tipo:\t\t{type(figura).__name__}")
    print(f"Área:\t\t{figura.calcular_area()}")
    print(f"Perímetro:\t{figura.calcular_perimetro():.2f}")
    print()

# Utilizar polimorfismo para tratar diferentes figuras de manera uniforme
imprimir_detalles(cuadrado)
imprimir_detalles(circulo)
```

```
Tipo:           Cuadrado
Área:           25
Perímetro:      20.00

Tipo:           Circulo
Área:           28.27431
Perímetro:      18.85
```