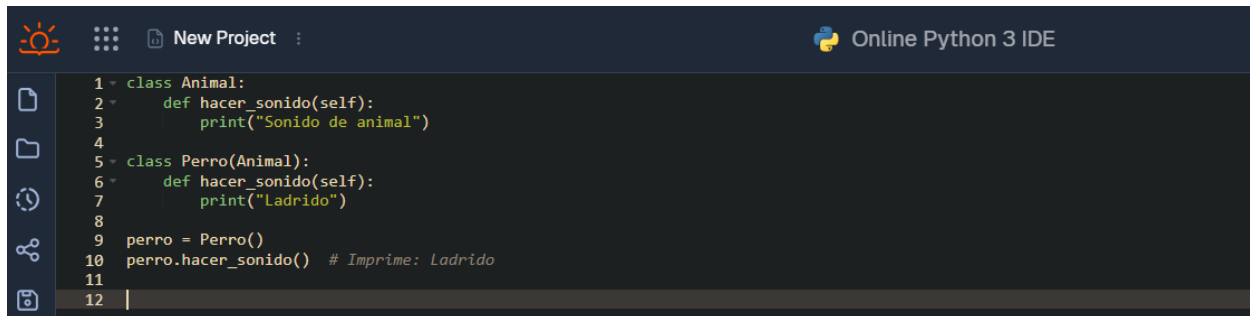


Formas de Herencia en Programación Orientada a Objetos (OOP)

1. Herencia Simple

En la herencia simple, una clase hija hereda los atributos y métodos de una sola clase base o superclase. Este es el tipo de herencia más común en muchos lenguajes de programación orientados a objetos como Java, C++, y Python. La clase hija puede usar, modificar o extender los comportamientos heredados de la clase base.

Ejemplo en Python:



```
1 class Animal:
2     def hacer_sonido(self):
3         print("Sonido de animal")
4
5 class Perro(Animal):
6     def hacer_sonido(self):
7         print("Ladrado")
8
9 perro = Perro()
10 perro.hacer_sonido() # Imprime: Ladrado
11
12
```

2. Herencia Múltiple

La herencia múltiple ocurre cuando una clase hija hereda de más de una clase base. Aunque algunos lenguajes como C++ permiten este tipo de herencia, otros como Java no lo soportan directamente (aunque se puede simular mediante interfaces). La herencia múltiple puede ser compleja, ya que puede generar problemas de ambigüedad si dos clases base tienen métodos con el mismo nombre.

Ejemplo en Python:

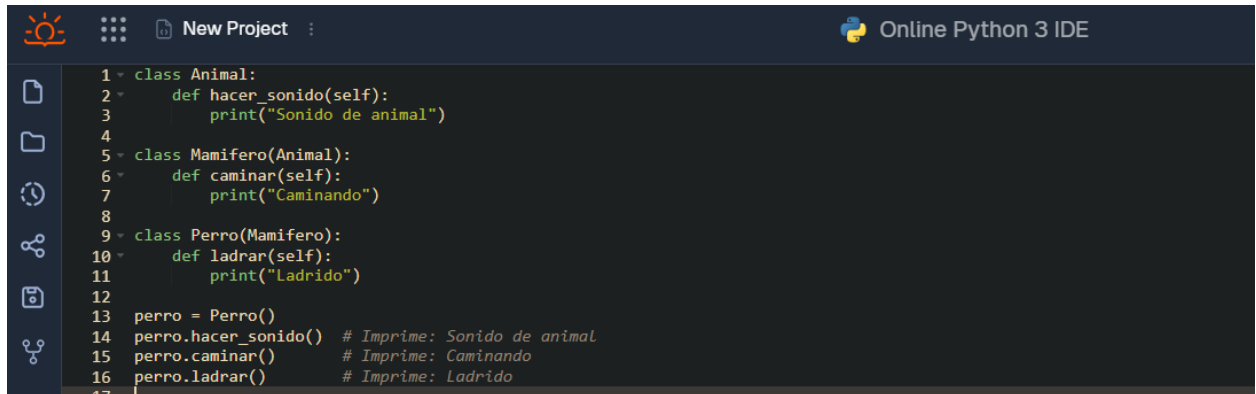


```
1 class Animal:
2     def comer(self):
3         print("Comiendo")
4
5 class Vehiculo:
6     def conducir(self):
7         print("Conduciendo")
8
9 class Auto(Vehiculo, Animal):
10     def mostrar_info(self):
11         print("Soy un auto")
12
13 auto = Auto()
14 auto.comer() # Imprime: Comiendo
15 auto.conducir() # Imprime: Conduciendo
16
```

3. Herencia Multinivel

En la herencia multinivel, una clase hija hereda de una clase base, y esa clase base a su vez hereda de otra clase. Este patrón crea una cadena de herencia en varios niveles.

Ejemplo en Python:

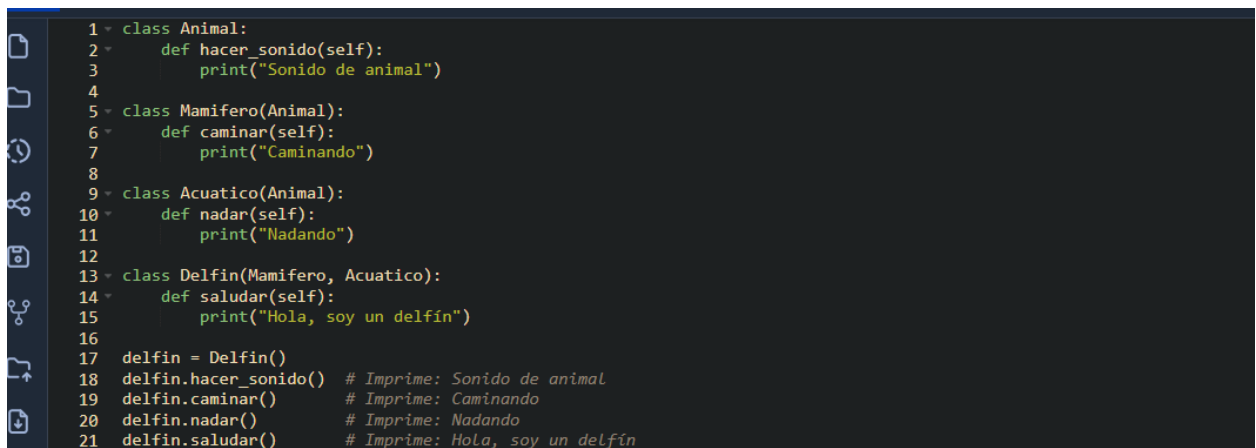


```
1 class Animal:
2     def hacer_sonido(self):
3         print("Sonido de animal")
4
5 class Mamifero(Animal):
6     def caminar(self):
7         print("Caminando")
8
9 class Perro(Mamifero):
10    def ladrar(self):
11        print("Ladrado")
12
13 perro = Perro()
14 perro.hacer_sonido() # Imprime: Sonido de animal
15 perro.caminar()     # Imprime: Caminando
16 perro.ladrar()      # Imprime: Ladrado
17
```

4. Herencia Híbrida

La herencia híbrida es una combinación de herencia múltiple y multinivel. Es un tipo de herencia compleja en la que una clase hereda de varias clases base, pero alguna de esas clases también puede ser derivada de otras clases. Algunos lenguajes, como Python, permiten este tipo de herencia.

Ejemplo en Python:



```
1 class Animal:
2     def hacer_sonido(self):
3         print("Sonido de animal")
4
5 class Mamifero(Animal):
6     def caminar(self):
7         print("Caminando")
8
9 class Acuatico(Animal):
10    def nadar(self):
11        print("Nadando")
12
13 class Delfin(Mamifero, Acuatico):
14    def saludar(self):
15        print("Hola, soy un delfín")
16
17 delfin = Delfin()
18 delfin.hacer_sonido() # Imprime: Sonido de animal
19 delfin.caminar()     # Imprime: Caminando
20 delfin.nadar()       # Imprime: Nadando
21 delfin.saludar()     # Imprime: Hola, soy un delfín

```

5. Herencia Virtual (solo en C++)

La herencia virtual se utiliza en C++ para resolver el problema de la ambigüedad en la herencia múltiple. Si dos clases base tienen un método común, la clase derivada podría heredar dos versiones de ese método. La herencia virtual resuelve este conflicto, garantizando que la clase derivada reciba solo una versión del método.

6. Herencia de Interfaces

En algunos lenguajes como Java y C#, las clases pueden implementar interfaces en lugar de heredar directamente de otras clases. Las interfaces definen un conjunto de métodos que deben ser implementados por la clase que las implementa, pero no

proporcionan una implementación concreta. Esto permite una forma de "herencia" sin la necesidad de usar clases base tradicionales.

Ejemplo en Java:

```
interface Animal {  
    void hacerSonido();  
}  
  
class Perro implements Animal {  
    public void hacerSonido() {  
        System.out.println("Ladrido");  
    }  
}  
  
public class Main {  
    public static void main(String[] args) {  
        Perro perro = new Perro();  
        perro.hacerSonido(); // Imprime: Ladrido  
    }  
}
```