

Presentation Timeline

1. Introducción

- Presentación del tema: Herencia Múltiple, Representación de Objetos y Conceptos de POO en C++, Python, C#.
- Importancia.

2. Herencia Múltiple

- Definición.
- Lenguajes que la soportan (C++, Python) vs. C# (interfaces).
- Beneficios.

3. Ambigüedad en Herencia Múltiple

- Problema: Miembros con el mismo nombre en clases base.
- Ejemplos: Python (`saludo()`), C++ (`imprimir()`).
- Problema del Diamante: Explicación y ejemplo.

4. Soluciones para la Ambigüedad

- Cambiar orden de clases base (Python MRO).
- Uso de `super()` (Python).
- Uso de nombre explícito de la clase (C++, Python).
- Redefinición del método en clase derivada.
- Herencia Virtual (C++) - para problema del diamante.
- Renombrar métodos/atributos.

5. MRO en Python

- Qué es el MRO (orden de búsqueda de miembros).
- Algoritmo C3 (Python 2.3+): Lineal y predecible.
- Regla de Fusión del C3: Proceso y detección de conflictos (`TypeError`).

6. Solución de C#

- Por qué C# no tiene herencia múltiple directa de clases.
- Uso de Interfaces: Definición de contratos.
- Manejo de Colisiones de Interfaces:

- Implementación Implícita.
- Implementación Explícita (ej: `IInterface.Metodo()`), acceso por *casting*.

7. Representación de Objetos en Memoria

- Varía según el lenguaje (gestión de memoria).
- Representación general en:
 - C++ (Pila/Heap, datos, vtables, gestión manual).
 - C# (Heap para objetos, Stack para referencias/tipos de valor, GC).
 - Python (Heap, referencias, diccionario interno, conteo de referencias/GC).
- Ejemplos (representación simplificada) para:
 - Objeto sin herencia.
 - Objeto con herencia simple.
 - Objeto con herencia múltiple (C++/Python) / Múltiples interfaces (C#).

8. Análisis de Conceptos Clave Adicionales

- Visibilidad (Modificadores/Convenciones en C++, C#, Python).
- Redefinición de Miembros (Anulación) (`virtual / override` , redefinición directa).
- Polimorfismo (Concepto). Cómo se logra en C++ / C# (`virtual/abstract`) y Python (tipado pato).
- Conversión (Casting) (Operadores/funciones principales en C++, C#, Python).

9. Conclusiones

- Resumen de puntos clave.
- Diferentes enfoques por lenguaje.
- Preguntas.