

16/05/2025

Trabajo Práctico 5 – Herencia – Parte 1

1. Describa las formas en las que la herencia:

- Fomenta la re-utilización de software.
- Ahorra tiempo durante el desarrollo de los programas.
- Ayuda a prevenir errores.
- Promueve una estructura jerárquica más clara y comprensible del código, facilitando su mantenimiento y extensión

2. Conteste con verdadero o falso a cada una de las siguientes proposiciones; en caso de ser falso, explique por qué.

- a) Los constructores de la superclase son llamados automáticamente por los constructores de la subclase sin necesidad de ninguna palabra clave especial.
- b) Una relación “tiene un” se implementa mediante la herencia.
- c) Una clase Auto tiene una relación “es un” con las clases Volante, Dirección y Frenos .
- d) La herencia fomenta la re-utilización de software comprobado, de alta calidad.
- e) Cuando una subclase re-define al método de una superclase utilizando la misma firma, se dice que la subclase sobrecarga a ese método de la superclase.

3. Explique la herencia con un ejemplo. ¿Aplicaría herencia porque detecta que en un conjunto de clases hay atributos compartidos, o porque estas clases comparten comportamiento/? La herencia es la única forma de manejar atributos compartidos o existen alternativas? Justifique.

4. ¿Cuáles son las desventajas de la herencia? Explique cada una.

5. ¿Cómo modifica la visibilidad de atributos y métodos el modificador de visibilidad protected? ¿Qué opina de su uso en atributos?

Compare el modificador de visibilidad protected con private y public.

6. A través de la herencia se puede sobre-escribir un método de la superclase en la subclase ¿Qué beneficios supone esto para la creación de sus aplicaciones? Dé un ejemplo de como esto puede simplificar el código y hacerlo mas flexible.

7. ¿Cuál es el propósito de la palabra clave super en el contexto de la herencia? Describe al menos dos situaciones comunes donde se utiliza y proporciona un breve ejemplo de cada una.

8. Describe el orden en que se ejecutan los constructores cuando se crea un objeto de una subclase en una jerarquía de herencia. ¿Qué sucede si un constructor de la superclase requiere argumentos?

9. ¿Qué significa declarar una clase como final? ¿Y un método? ¿Con qué propósito se utilizan estas declaraciones en el contexto de la herencia?

10.

Ejercicio 8.17 Observe el siguiente código. Se tienen cuatro clases (O, X, T y M) y una variable de cada una de ellas.

```
O o;  
X x;  
T t;  
M m;
```

Las siguientes asignaciones son todas legales (asuma que todas compilan).

```
m = t;  
m = x;  
o = t;
```

Las siguientes asignaciones son todas ilegales (provocan error en la compilación):

```
o = m;  
o = x;  
x = o;
```

¿Qué puede decir sobre la relación entre estas clases?