

Guía de Estudio: Herencia

1. ¿Qué nombre recibe la clase que hereda y qué nombre recibe la clase que es heredada?

Hereda: Clase base o padre

Heredada: Clase derivadas, subclases o hija.

2. ¿Qué significa que la herencia es transitiva?

Que si A hereda de B y B hereda de C entonces A es derivada de C.

3. ¿Se heredan los constructores?

No

4. ¿Se heredan los miembros private de la clase base?

Si, se heredan pero no se puede acceder a ellos.

5. ¿Qué es herencia múltiple? ¿Es posible en C#?

Se define herencia multiple cuando una clase derivada puede heredar de una o más clases base. En C# no, no lo soporta.

6. ¿Que nos permite la herencia? ¿Cuál es su propósito?

La herencia nos permite crear clases nuevas a partir de otras ya existentes, su proposito es organizar major las clases que componen una misma realidad, agrupandolas en base a sus atributos y a su comportamiento.

7. ¿Una clase pública puede heredar de una clase privada?

Una subclase no puede ser mas accessible que su clase base.

8. ¿Qué es una clase sellada (sealed)?

Un tipo de clase que no puede ser heredada por otra clase.

9. ¿Una clase sellada puede heredar de otras clases? (Ser clase derivada)

Si puede.

10. ¿Cómo actúa el modificador `protected` en los miembros de la clase base para una clase derivada y cómo para una clase no-derivada?

Para una clase derivada el `protected` equivale al `public` y para una NO derivada equivale al `private`

11. ¿Qué pasa si la clase derivada no hace una llamada explícita a un constructor de la clase base? En esta situación, ¿qué pasa si la clase base declaró explícitamente un constructor con parámetros de entrada?

Si la clase derivada no tiene un constructor declarado el compilador de C# proporciona automáticamente una llamada al constructor sin parámetros o predeterminado de la clase base.

Si la clase base tiene declarado un constructor con parametros la clase derivada esta obligada a declarar un constructor que utilice los mismos parametros.