

Programación Orientada a Objetos

Instrucciones del test:

Tiempo de realización: 20 minutos

Si la frase es verdadera debéis marcar **A**, y si es falsa **B**.

¿Verdadero (A) o Falso (B)?

1. Un mensaje tiene cero o más argumentos. Por el contrario, una llamada a procedimiento/función tiene uno o más argumentos.
2. La interpretación de un mismo mensaje puede variar en función del receptor del mismo y/o del tipo de información adicional que lo acompaña.
3. En una agregación, la existencia de un objeto *parte* depende de la existencia del objeto *todo* que lo contiene.
4. Una forma de mejorar el diseño de un sistema es reducir su acoplamiento y aumentar su cohesión.
5. Los inicializadores en C++ tienen el formato *nombreAtributo(valor)* y se colocan entre la lista de argumentos y el cuerpo de cualquier método, permitiendo asignar valores a los atributos de instancia de la clase en la que se define dicho método.
6. Un atributo estático ocupa una zona de memoria que es compartida por todos los objetos de la clase en la que se define, aunque no por los objetos de cualquier clase derivada de ella.
7. Un atributo de clase debe tener visibilidad pública para poder ser accedido por los objetos de la clase.
8. La herencia múltiple se produce cuando de una misma clase base heredan varias clases derivadas.
9. Un atributo protegido en la clase base es también protegido en cualquier clase que derive de dicha clase base, independientemente del tipo de herencia utilizado.
10. Cuando se crea un objeto de una clase D que deriva de una clase B, el orden de ejecución de los constructores es siempre *B()* *D()*.
11. Una clase abstracta es una clase que no permite definir instancias de ella.
12. Un interfaz es una clase abstracta con al menos un método de instancia abstracto.
13. Un método de clase (estático) no puede tener enlace dinámico.
14. Un método virtual en C++ siempre tiene enlace estático.
15. El principio de sustitución implica una coerción entre tipos de una misma jerarquía de clases.
16. La llamada a un método sobrescrito se resuelve en tiempo de compilación.
17. El *downcasting* estático siempre es seguro.
18. Los métodos definidos en una clase genérica son a su vez genéricos.
19. No se puede definir un bloque *catch* sin su correspondiente bloque *try*.
20. La instrucción *throw* (en C++) sólo permite lanzar objetos de clase *exception* o de clases derivadas de ella.