

Programación Orientada a Objetos

Julio 2010

Instrucciones del test:

Tiempo de realización: 20 minutos

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

1. La herencia protegida permite a los métodos de la clase derivada acceder a las propiedades privadas de la clase base.
2. Una clase abstracta se caracteriza por no tener definido ningún constructor.
3. La siguiente clase en C++: `class S {public: virtual ~S()=0;};` define una interfaz.
4. Los métodos virtuales son métodos abstractos.
5. Los métodos abstractos siempre tienen enlace dinámico.
6. Los constructores siempre tienen enlace dinámico.
7. En C++, un atributo de clase debe declararse dentro de la clase con el modificador `static`.
8. Un método sobrecargado es aquel que recibe como argumento al menos una variable polimórfica.
9. Un método tiene polimorfismo puro cuando devuelve una variable polimórfica.
10. Un atributo declarado con visibilidad protegida en una clase A es accesible desde clases definidas en el mismo espacio de nombres donde se definió A.
11. Dada la siguiente definición de clase en C++:

```
class TClase {  
    public:  
        TClase(int dim);  
    private:    int var1;  
};
```

La instrucción `TClase c1;` no da error de compilación e invoca al constructor por defecto.

12. Una de las características básicas de un lenguaje orientado a objetos es que todos los objetos de la misma clase pueden recibir los mismos mensajes.
13. Hablamos de encapsulación cuando agrupamos datos junto con las operaciones que pueden realizarse sobre esos datos.
14. Una operación de clase sólo puede ser invocada mediante objetos constantes.
15. En C++ los constructores se pueden declarar como métodos virtuales.
16. En la misma clase, podemos definir constructores con distinta visibilidad.
17. Si no se captura una excepción lanzada por un método, el programa no advierte que ha ocurrido algún error y continua su ejecución normalmente.
18. En C++, es obligatorio especificar qué excepciones lanza una función mediante una cláusula `throw` tras la declaración de la función.
19. Dada una clase genérica, se pueden derivar de ella clases no genéricas.
20. Una clase interfaz no debe tener atributos de instancia. Una clase abstracta sí puede tenerlos.