

Test Programación Unidad 5

PREGUNTAS

Pregunta 1

¿Cuál es la principal diferencia entre la Programación Orientada a Objetos y los modelos procedimentales tradicionales?

- A)** La POO no utiliza funciones, solo variables globales
- B)** La POO modela el software como una colección de objetos que representan entidades del mundo real con estado y comportamiento
- C)** La POO es más lenta pero más fácil de aprender
- D)** La POO solo se puede usar en Java y no en otros lenguajes

Pregunta 2

¿Cuál es la relación correcta entre clase y objeto?

- A)** Ambos términos significan exactamente lo mismo
- B)** La clase es una plantilla o molde, mientras que el objeto es una instancia específica creada a partir de esa clase
- C)** Un objeto es una colección de clases
- D)** La clase solo existe en documentación, el objeto es lo que realmente se ejecuta

Pregunta 3

¿Cuál es la función principal de un constructor en una clase?

- A)** Destruir objetos cuando ya no se necesitan
- B)** Inicializar el estado del objeto y garantizar que comienza en condiciones válidas
- C)** Convertir un objeto a texto para imprimirla
- D)** Copiar los valores de un objeto a otro

Eliminado: <object>

PROMETEO

Pregunta 4

¿Por qué es importante declarar los atributos de una clase como private en lugar de public?

- A)** Porque ocupa menos memoria
- B)** Porque el compilador lo obliga
- C)** Para proteger el estado interno del objeto y permitir acceso controlado mediante getters y setters, evitando estados inválidos
- D)** Para que el código sea más rápido de ejecutar

Pregunta 5

¿Cuál de las siguientes afirmaciones describe correctamente la herencia?

- A)** Una subclase no puede tener sus propios atributos, solo hereda los de la superclase
- B)** Una clase puede heredar de múltiples clases simultáneamente en Java
- C)** La herencia permite crear nuevas clases basadas en otras existentes, reutilizando código y especializando comportamientos
- D)** La herencia solo es útil en aplicaciones grandes, no en proyectos pequeños

Pregunta 6

¿Cuál es el propósito de utilizar la palabra clave super en una subclase?

- A)** Acceder a atributos privados de la superclase directamente
- B)** Crear una instancia de la superclase desde la subclase
- C)** Invocar constructores y métodos de la superclase, reutilizando su lógica sin duplicar código
- D)** Impedir que la subclase modifique el comportamiento de la superclase

Eliminado: <object>

PROMETEO

Pregunta 7

¿Qué significa el término “polimorfismo” en el contexto de la Programación Orientada a Objetos?

- A)** Tener muchas clases heredando de una sola superclase
- B)** Una misma acción puede tener diferentes comportamientos según el tipo real del objeto que la ejecuta (binding dinámico)
- C)** La capacidad de una clase de implementar múltiples interfaces
- D)** La imposibilidad de crear objetos de una clase abstracta

Pregunta 8

¿Cuál es la principal diferencia entre una interfaz y una clase abstracta?

- A)** Las interfaces pueden tener atributos con valores, las clases abstractas no
- B)** Una clase puede heredar de múltiples interfaces, pero solo de una clase abstracta. Además, las interfaces definen qué debe hacer una clase, las abstractas definen qué es y cómo comportarse
- C)** Las clases abstractas son más modernas que las interfaces
- D)** Las interfaces no pueden tener métodos con implementación en ningún caso

Pregunta 9: Sobrescritura de Métodos (@Override)

¿Cuál es la importancia de usar la anotación @Override al redefinir un método en una subclase?

- A)** Es obligatorio usarla o el compilador no acepta el código
- B)** Ayuda al compilador a detectar errores tipográficos en la firma del método y asegura que realmente existe en la clase padre
- C)** Hace que el método sea más rápido
- D)** Permite que el método tenga un nombre diferente en la subclase

Eliminado: <object>

PROMETEO

Pregunta 10: Principio de Responsabilidad Única (SRP)

¿Qué establece el principio SRP (Single Responsibility Principle) en el diseño de clases?

- A)** Cada clase debe tener exactamente un atributo
- B)** Cada clase debe tener exactamente un método público
- C)** Cada clase debe tener una única responsabilidad principal, evitando que acumule demasiadas funcionalidades
- D)** Cada clase debe heredar de una única superclase

HOJA DE RESPUESTAS

Pregunta	Respuesta	Justificación
1	B	La POO modela el software como objetos con estado y comportamiento, diferente del enfoque procedimental tradicional
2	B	La clase es el molde o plantilla; el objeto es la instancia específica creada a partir de esa clase
3	B	El constructor inicializa el estado del objeto y garantiza condiciones válidas desde su creación
4	C	Declarar atributos como private protege el estado interno y permite acceso controlado mediante getters/setters
5	C	La herencia permite reutilizar código de una clase base y especializarlo en subclases
6		super permite invocar constructores y métodos de la superclase, reutilizando su lógica
7	B	Polimorfismo significa que la misma

Eliminado: <object>

PROMETEO

Pregunta	Respuesta	Justificación
		acción puede tener distintos comportamientos según el tipo real del objeto (binding dinámico)
8	B	Una clase implementa múltiples interfaces pero hereda de una sola clase abstracta. Interfaces definen el qué, abstractas el qué es
9	B	@Override ayuda al compilador a detectar errores en la firma del método y asegura su existencia en la clase padre
10	C	SRP establece que cada clase debe tener una única responsabilidad principal para mejorar mantenibilidad