

Rúbrica Analítica para Prueba de Programación en Java: Herencia y Orientación a Objetos

Rúbrica Analítica | Tecnología e Informática | Informática | 4 niveles

Descripción

Esta rúbrica evalúa la implementación de una prueba de programación orientada a objetos en Java, enfocada en herencia, utilizando dos paquetes con tres clases, constructores, métodos getter/setter, y métodos para calcular y determinar la aprobación de estudiantes. Cada criterio se califica en cuatro niveles para identificar fortalezas y áreas de mejora.

Rúbrica

Rúbrica Analítica para Prueba de Programación en Java: Herencia y Orientación a Objetos

Esta rúbrica evalúa la implementación de una prueba de programación orientada a objetos en Java, enfocada en herencia, utilizando dos paquetes con tres clases, constructores, métodos getter/setter, y métodos para calcular y determinar la aprobación de estudiantes. Cada criterio se califica en cuatro niveles para identificar fortalezas y áreas de mejora.

Criterios / Niveles	Excelente (4)	Bueno (3)	Aceptable (2)	Bajo (1)
Definición de los cinco atributos en las clases	Los cinco atributos están correctamente definidos con tipos adecuados y encapsulación en todas las clases.	Faltan uno o dos atributos o hay pequeños errores en la definición pero no afectan la funcionalidad.	Se definen menos de tres atributos o existen errores que dificultan el uso correcto de los atributos.	No se definen los atributos requeridos o están mal declarados, afectando gravemente el diseño.
Implementación de dos paquetes con tres clases	Los dos paquetes están correctamente organizados y contienen tres clases con estructura clara y coherente.	Los paquetes y clases están presentes, pero la organización o coherencia es menor a lo esperado.	Faltan uno o más paquetes o clases, o la organización es confusa y afecta la comprensión.	No se implementan los paquetes ni las clases requeridas o están mal estructurados.

Criterios / Niveles	Excelente (4)	Bueno (3)	Aceptable (2)	Bajo (1)
Uso correcto del método constructor en las clases	Todos los constructores inicializan correctamente los atributos y permiten la creación adecuada de objetos.	Los constructores están presentes pero algunos no inicializan todos los atributos o tienen errores menores.	Constructores incompletos o ausentes en la mayoría de las clases, dificultando la creación de objetos.	No se utilizan constructores o están mal implementados, impidiendo la correcta instanciación.
Implementación de métodos getter y setter	Se implementan getters y setters para todos los atributos, respetando la encapsulación y buenas prácticas.	Faltan algunos getters o setters o tienen errores menores que no afectan gravemente el acceso a atributos.	Se implementan pocos métodos o con errores que limitan el acceso controlado a los atributos.	No se usan getters ni setters o están mal implementados, rompiendo la encapsulación.
Método para determinar si el estudiante aprobó	El método evalúa correctamente la condición de aprobación con lógica clara y resultados precisos.	El método funciona pero la lógica es poco clara o tiene pequeños errores que no afectan el resultado final.	El método existe pero tiene errores importantes que pueden afectar la determinación correcta.	No se implementa el método o la lógica es incorrecta, impidiendo determinar la aprobación.
Método para retorno y cálculo del promedio de notas	El método calcula el promedio correctamente y retorna el valor esperado con precisión.	El cálculo del promedio es correcto pero la implementación puede ser poco eficiente o clara.	El método existe pero el cálculo es incorrecto o incompleto, afectando el resultado final.	No se implementa el método o el cálculo es erróneo, impidiendo obtener el promedio.
Uso del método main para instanciar objetos y mostrar resultados	El método main instancia correctamente objetos, llama a los métodos adecuados y muestra resultados claros.	El método main funciona pero la presentación o llamadas a métodos podrían mejorarse.	El método main existe pero tiene errores que dificultan la ejecución o visualización de resultados.	No se implementa método main o no permite probar correctamente el programa.
Aplicación correcta de herencia y orientación a objetos	La herencia está bien aplicada, con clases derivadas que reutilizan y extienden funcionalidad correctamente.	Se evidencia herencia pero con algunos errores o usos poco claros en la estructura del programa.	La herencia está implementada de forma limitada o incorrecta, con poca reutilización o errores.	No se aplica herencia o el diseño orientado a objetos es deficiente o inexistente.