

Ayudantía Pre-Certamen 1

Diseño y Programación Orientado a Objetos

Matías Soto S. (Basada en la ayudantia 2021 de F. Llorens)

14 de Abril 2022

1. Preguntas teóricas

1. ¿Qué se entiende por POO?
2. ¿Qué diferencia existe entre Programación imperativa y Programación declarativa?
3. ¿Qué es una clase?
4. Defina los conceptos de Herencia y Polimorfismo.
5. ¿Qué se entiende por abstracción en POO?
6. ¿Qué beneficios tiene Java por sobre C++?
7. ¿Qué es la JVM?
8. ¿Qué diferencia existe entre compilar y ejecutar?
9. ¿Qué función tiene la keyword static? ¿Y la keyword final? ¿Cuándo podemos utilizarlas?
10. ¿Qué es un constructor? ¿Es necesario implementarlo siempre? ¿Por Qué?
11. ¿Cuál es la utilidad del método finalize()?
12. ¿Qué es una referencia? ¿Existen referencias en tipos primitivos de datos?
13. Defina cada uno de los modificadores de acceso. Indique la restricción que aplican en base a: clases, paquetes y subclases.
14. Defina override y en qué caso o casos es posible utilizarlo.
15. ¿Qué se entiende por ligado dinámico?
16. ¿Qué es una clase abstracta? Nombre sus principales características.
17. ¿Qué es una interfaz? Nombre sus principales características.
18. ¿Qué es una clase anidada? ¿Cuáles son sus tipos?

19. ¿Para Qué se utiliza `super()`?
20. ¿Qué es la copia profunda (deep clone) y la copia baja (shallow copy)?
21. ¿Qué es una expresión lambda? ¿Cual es su utilidad?
22. ¿A Qué nos referimos con manejo de excepciones?

2. Verdadero o Falso

1. Si una clase es abstracta entonces puede ser instanciada.
2. Si en una clase no define un constructor, se asume uno por defecto, sin argumentos.
3. Un método publico de una superclase sólo se puede declarar como tal sino protegido en la subclase.
4. El modificador final se puede usar sólo para las clases.
5. En Java, es posible definir relaciones de herencia, tanto para clases como para interfaces.
6. En Java, una clase que es abstracta y final a la vez, debiera ser representada correctamente por una interface.
7. Un constructor, por tratarse de un método, debe retornar un valor, que corresponde al tipo de la referencia en cuestión.
8. En Java, todos los parámetros (tanto tipos primitivos como referencias) se pasan por valor.
9. La referencia this, usada dentro de un método estático, refiere al objeto sobre el cual se invocó el método
10. El ligado dinámico permite crear variables de un tipo que pueden referenciar objetos de cualquier clase de la jerarquía de herencia.
11. Las variables miembros que se comparten entre todos los objetos de una clase deben estar definidas con el modificador final.
12. Para pasar una referencia del objeto actualmente en uso como parámetro a otro método, se debe utilizar self.
13. Una interfaz puede tener constantes, las cuales implícitamente son públicas, no estáticas y finales.
14. Dos métodos predefinidos de la clase Object son equals() y clone().
15. Si a pesar de ejecutar el recolector de basura, a´un no existe memoria suficiente, se lanza una excepción llamada MemoryNothing.
16. . En Java, es necesario el uso del operador de desreferenciación.
17. El recolector de basura en Java libera los punteros que contengan referencias.