

PARTE TEÓRICA - TEST [2,5 PUNTOS]:

Sólo una de las respuestas es válida. Las respuestas correctas se puntuarán con +1.0, mientras que las respondidas de manera incorrecta se puntuarán con -0.25. Las no contestadas no tendrán influencia ni positiva ni negativa en la nota.

Pregunta 1 ¿Cuál de las siguientes inicializaciones para un array es la correcta?

- a) `Array nombreArray3 = new int[20];`
- b) `int [] nombreArray4 = new int[20];`
- c) `int [] nombreArray1 = new Array(20);`
- d) `Array nombreArray2 = new Array(20);`

Pregunta 2 ¿Qué pasaría a continuación si se crea un objeto sin parámetros con la sentencia `A objA = new A();` siendo la clase A la siguiente?

```
public class A
{
    int nA;
    public A(int nA) {this.nA=nA;}
}
```

- a) Se crearía un objeto objA ya que se llamaría al constructor por defecto de la clase A
- b) Se crearía un objeto objA, aunque al no tener parámetros, se pasaría un valor nulo al constructor `publicA(int nA){this.nA = nA;}`
- c) Daría un error de compilación debido a que si se declara un constructor en una clase el constructor por defecto deja de ser accesible
- d) Daría un error de compilación por poner la palabra reservada new delante de la expresión A()

Pregunta 3 ¿Qué es necesario si queremos utilizar el método de una clase predefinida como por ejemplo `Integer.parseInt("String");`?

- a) Solamente se crea una instancia de la clase predefinida.
En este caso: `Integer integ = new Integer(); integ.parseInt("String");`
- b) No es necesario importar la clase en cuestión ni crear una instancia de la clase predefinida
- c) Importar la clase en cuestión. En este caso: `import java.lang.Integer;` y crear una instancia de la clase predefinida. En este caso: `Integer integ = new Integer(); integ.parseInt("String");`
- d) Solamente se importa la clase en cuestión. En este caso: `import java.lang.Integer;`

Pregunta 4 Un método de clase static, ¿se podría llamar sin instanciarse un objeto de la clase?

- a) No, siempre hay que declarar el objeto y luego llamar al método
- b) No, siempre hay que declarar el objeto, crearlo y a continuación llamar al método
- c) Sí, se podría llamar al método desde la misma clase
- d) Si, se podría llamar al método pero sólo si la clase es abstracta

Pregunta 5 Suponiendo que, en una estructura de herencia, cada subclase de la clase abstracta "figura" tiene su propio método "calcula_area", y dado un objeto "f" declarado como "figura", ¿cómo tendrían que estar declarados los métodos para poder ejecutar "f.calcula_area()"?

- a) "calcula_area" sólo en las subclases
- b) "calcula_area" en "figura" y en sus subclases
- c) "calcula_area" en figura y no en sus subclases
- d) No puede ser ejecutado de ninguna manera

Pregunta 6 Sea "c1" una clase con 6 atributos, y sea "c2" otra clase con 3 atributos, 3 de los atributos de "c1" son los de la clase "c2". Tienen los mismos métodos aunque algunos se comportan de manera diferente. ¿Qué relación de herencia es la correcta y por qué?

- a) No se puede establecer herencia dado que los mismos métodos se comportan de manera diferente
- b) No se puede establecer herencia dado que los atributos no son exactamente los mismos
- c) "c2" es hija de "c1" dado que todos sus atributos están incluidos en "c2"
- d) "c1" es hija de "c2" dado que incluye sus atributos y añade nuevos

Pregunta 7 ¿Cuál de las siguientes declaraciones nunca podría generar dos objetos de la misma clase con los mismo valores?

- a) `Triangulo objCir1=new Triangulo(5,8);`
`Triangulo objCir2=new Triangulo(objCir1);`
- b) `Triangulo objCir1=new Triangulo(5,8);`
`Triangulo objCir2=objCir1;`
- c) `Triangulo objCir1=new Triangulo(5,8);`
`Triangulo objCir2=new Triangulo(5,8);`
- d) `Triangulo objCir1=new Triangulo(5);`
`Triangulo objCir2=new Triangulo(5);`

Pregunta 8 Una diferencia entre HashSet y TreeSetes ...

- a) HashSet implementa la interfaz Set y TreeSet no
- b) En TreeSet se permiten elementos repetidos y en HashSet no
- c) TreeSet mantiene todos los elementos ordenados en su orden natural o de acuerdo a como indique el Comparator que se indica en el constructor, y HashSet no hace esto
- d) HashSet se puede recorrer mediante un Iterator, mientras que TreeSet no

Pregunta 9 ¿Qué tipo de acceso se puede emplear para sobrecargar los métodos que implementan una interfaz?

- a) Pueden ser métodos con acceso public y private, pero no protected
- b) Sólo pueden ser con acceso public
- c) Sólo pueden ser con acceso private
- d) Pueden emplear cualquier tipo de acceso de Java

Pregunta 10 ¿Qué se entiende por cohesión en lo que a una unidad de código se refiere?

- a) Que la unidad de código es responsable de una y sólo una tarea
- b) Que la unidad de código no se descompone en otras unidades de programación (métodos)
- c) Que todos los métodos que forman parte de la unidad de código se encuentran en la misma clase
- d) Todas las anteriores son correctas

Pregunta 11 Respecto a las variables polimórficas en Java...

- a) Es aquella que puede almacenar varios objetos de diferentes tipos
- b) Ese concepto, al igual que la herencia múltiple, no se implementa en Java
- c) Cada variable objeto en Java es potencialmente polimórfica
- d) Se declaran siempre como public o private, pero nunca protected

Pregunta 12 Respecto a la declaración de las clases internas y su relación con su clase envolvente...

- a) La clase interna puede acceder tanto a los métodos privados como a los públicos y protegidos de la clase envolvente
- b) La clase interna puede acceder tanto a los métodos públicos como a los métodos protegidos de la clase envolvente, pero no a los métodos privados de la misma
- c) La clase interna puede acceder a los métodos públicos de la clase envolvente, pero no a los métodos privados ni a los protegidos de la misma
- d) La clase interna no puede acceder ni a los métodos públicos ni a los privados ni a los protegidos de la clase envolvente

Pregunta 13 En lo que se refiere a las clases internas anónimas, se puede afirmar...

- a) Suelen emplearse en los lugares en los que se requiere la implementación de una sola instancia
- b) Siempre se hará referencia la instancia mediante su supertipo
- c) Permiten definir una clase y crear una instancia de ella, todo en un solo paso
- d) Todas las anteriores son correctas

Pregunta 14 ¿Cuál de las siguientes afirmaciones es cierta para los interfaces en Java?

- a) Todos los métodos de la interfaz son abstractos, pero se permiten métodos con cuerpos. No es necesaria la palabra clave *abstract*
- b) Todos los métodos de la interfaz son abstractos, luego no se permiten métodos on cuerpos. Es necesaria la palabra clave *abstract*
- c) Todos los métodos de la interfaz son abstractos, luego no se permiten métodos con cuerpos. No es necesaria la palabra clave *abstract*
- d) Ninguna de las afirmaciones anteriores es cierta

Pregunta 15 ¿Qué instrucción permite cargar en la variable “a” el tamaño del array?

- a) `int a; int [] b = new int[10]; a=b.size;`
- b) `int a; int [] b = new int[10]; a=b.size();`
- c) `int a; int [] b = new int[10]; a=b.length;`
- d) `int a; int [] b = new int[10]; a=b.length();`

PARTE PRÁCTICA [6,5 PUNTOS]:

Un banco desea enviar a sus clientes una carta, mensaje de correo electrónico o mensaje al móvil (según los datos y preferencias de cada cliente) de agradecimiento por cada uno de los productos financieros que ha contratado el cliente. Cada cliente puede haber contratado varios productos (tarjetas de crédito, débito, plan de pensiones, seguro, fondos de inversión, etc.) y cada tipo de producto requiere una carta/mensaje diferente. Se quiere usar una jerarquía de clases para representar los diferentes tipos de producto y otra para los diferentes tipos de mensajes. Para el programa hay que usar un ArrayList del tipo más adecuado más un iterador para gestionar el envío de mensajes conjuntamente.

- a) **[2 puntos]** Identificar la estructura y las relaciones de herencia y de uso de las clases necesarias para la aplicación que realice el trabajo descrito.
- b) **[2 puntos]** Dibujar un esquema de la organización de estas clases en el diseño global.
- c) **[2 puntos]** Implementar el método “main” del programa mostrando cómo se gestiona el envío de los mensajes.
- d) **[0,5 puntos]** ¿Qué cambios serían necesarios en el diseño y programa para mandar otro tipo de mensaje, algo de publicidad sobre nuevos productos, conjuntamente con el mensaje de agradecimiento?