

Parcial de Java

1. Sea una aplicación con dos clases, Jugador en el paquete juego.modelo.personajes y Punto en juego.modelo.geometria. ¿En qué carpetas se organiza el código de la aplicación?
2. Indica los posibles errores en el siguiente código

<pre>package paquete1; public class A { static int at=35; } </pre>	<pre>package paquete2; public class B extends A { public static void main (String[] args){ Object obj = at; System.out.println("at = " + obj); } } </pre>
---	---

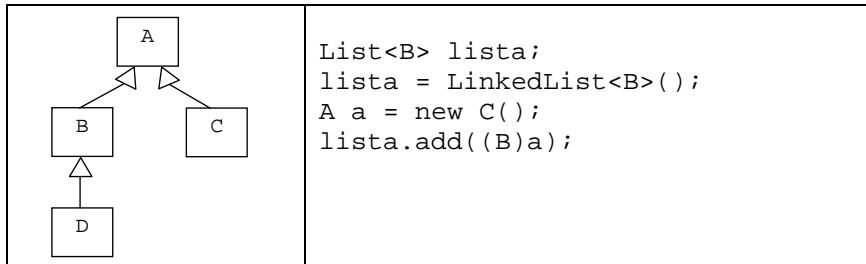
3. Indica si el siguiente enunciado es verdadero o falso, justificando la respuesta: “Un atributo de tipo enumerado que no es inicializado en la declaración ni en el constructor, toma como valor por defecto el primer valor del enumerado”.
4. En la implementación de una clase, ¿puede un método de instancia llamar a un método de clase de la propia clase? ¿Y al revés? Justifica la respuesta.
5. ¿Cuáles son los valores de oa1.at, oa2.at y de v después de la ejecución del método met?

<pre>public class A{ private int at; public A(){ at=0; } public void met(A p1, int p2){ p1 = new A(); p1.at=5; at = ++p2; } } </pre>	<pre>A oa1 = new A(); A oa2 = new A(); int v = 5; oa1.met(oa2,v); </pre>
--	---

6. ¿Dos métodos declarados en la misma clase con el mismo nombre, la misma lista de parámetros y distinto valor de retorno están sobrecargados? Justifica la respuesta.
7. Señala los posibles errores del siguiente código y corrígelo para que compile.

<pre>package prueba; public class A { int at1; public A(int v){ at1 = v; } } </pre>	<pre>public class B extends A { private int at2; public B(int v){ at1 = v; at2 = v; } } </pre>
--	--

8. ¿En la redefinición de un método podemos cambiar el tipo de los parámetros o del valor de retorno? Justifica la respuesta.
9. Dada la jerarquía de herencia de la figura. ¿Qué problemas tiene el código siguiente?



10. ¿Tiene sentido que una clase final declare algún método que sea abstracto? Justifica la respuesta.
11. Identifica los posibles errores del siguiente fragmento de código:

```

class A {
    private final int at = 4;
    public void setAt(int at){
        this.at = at;
    }
}
            
```

12. Indica si el siguiente enunciado es verdadero o falso, justificando la respuesta: “Después de clonar un objeto, el objeto original siempre es igual (equals) al objeto clonado, es decir, se cumple que `objOriginal.equals(objClonado) == true`”.
13. Dada una variable `var`, si `(var instanceof A)` es igual a `true`, ¿podemos asegurar que el tipo dinámico de `var` es `A`? Justifica la respuesta.
14. ¿Qué habría que hacer para que el programa muestre el siguiente mensaje tras su ejecución: “Exception in thread "main" java.lang.AssertionError: programa incompleto”

```

1. public class Examen{
2.     public static void main (String[] args){
3.
4.     }
}
            
```

15. Indica si el siguiente enunciado es verdadero o falso, justificando la respuesta: “Sea la clase genérica `Contenedor<T>`, sobre las entidades de tipo genérico (tipo `T`) dentro del código de la clase `Contenedor` no es posible aplicar ningún método”.

16. ¿Es correcto el siguiente código? Justifica la respuesta.

```

Object obj = 5;
int i = (int)obj;
            
```

17. Sea la interfaz `Dibujable`, ¿puedo declarar y construir un array cuyo tipo sea el de la interfaz? Justifica la respuesta.

```

Dibujable[] array = new Dibujable[10];
            
```

18. ¿Cómo podemos recorrer la colección de valores de un `Map`?
19. Dado el siguiente código, si `Movil` es una interfaz ¿el tipo `Flota` debe ser una clase o una interfaz? Justifica la respuesta

```
Barco barco = new Barco();
Movil m = barco;
Flota f = barco;
f= m;
```

20. Durante la ejecución de una clase de pruebas `JUnit` ¿cuántas veces se ejecuta el método `setUp`?
21. Sea la clase `Jefe` subtipo de `Empleado` y dada la declaración del método `listar` ¿Podríamos pasar como parámetro un objeto de tipo estático `List<Jefe>`? ¿Cómo se podría arreglar? Justifica la respuesta.

```
public void listar (List<Empleado> empleados){
    for (Empleado e: empleados)
        System.out.println(e);
}
```

22. Sea una variable que referencia a una colección cuyo tipo estático es `Collection<String>`. ¿La aplicación del método `add` añade siempre el elemento a la colección? Justifica la respuesta.
23. ¿Una aplicación Java compilada en Windows puede ejecutarse en un ordenador con un sistema operativo diferente? Justifica la respuesta.
24. Identifica los posibles errores de compilación del siguiente código:

```
1. public class Test implements Cloneable{
2.     private Demo d;
3.     private int at;
4.     ...
5.     public Test clone(){
6.         return super.clone();
7.     }
8. }
```

25. Dadas las excepciones `Exc0` que hereda de `Exception` y `Exc1` que hereda de `Exc0` ¿Cuál es el resultado de la ejecución del siguiente código? Justifica la respuesta.

```
public class Examen {
    public static void main (String[] args){
        try{
            throw new Exc1();

        }catch (Exc0 e1){
            System.out.println("Capturada Exc0");

        }catch(Exception e2){
            System.out.println("Capturada Exception");
        }
    }
}
```