

1.- En el polimorfismo:

- (a) Es el programador el que decide el método correcto al que se debe llamar.
- (b) Es el compilador el que decide el método correcto al que se debe llamar.
- (c) Ninguna de las respuestas anteriores es correcta.
- (d) *** Es en tiempo de ejecución cuando se decide al método correcto al que se debe llamar.

2.- Cual de las siguientes opciones es la correcta:

- (a) *** Las pruebas de caja blanca comprueban la estructura del programa y se realizan mediante JUnit y EclEmma.
- (b) Las pruebas de caja blanca comprueban la estructura del programa y se realizan mediante EclEmma.
- (c) Las pruebas de caja blanca comprueban la funcionalidad del programa y se realizan mediante EclEmma.
- (d) Las pruebas de caja blanca comprueban la funcionalidad del programa y se realizan mediante JUnit y EclEmma.

3.- Dado el siguiente código, ¿qué deberías programar en (1) para probar que funciona correctamente el método de objeto *imprimir()*?:

```
public class Parcial1{
    public void imprimir(){
        System.out.println("Prueba de programa");
    }
    public static void main (String [] args){
        (1) ...
    }
}
```

- (a) *** En (1) instanciar un objeto de la clase Parcial1 y llamar al método: *Parcial1 p = new Parcial1(); p.imprimir();*
- (b) Cambiamos la visibilidad de *imprimir()* de public a private y llamamos directamente al método *imprimir()*; en (1)
- (c) Programamos en (1) la misma línea que tiene el método *imprimir()* en *main()*: *System.out.println("Prueba de programa");*
- (d) Añadir static al método *imprimir()*: *public static void imprimir()* y llamarlo desde (1) como: *imprimir()*;

4.- ¿Cuál de la siguientes opciones es la correcta?:

- (a) La sobrecarga sólo se da en los constructores puesto que permite inicializar el estado del objeto de forma diferente.
- (b) La sobrecarga en la orientación a objetos permite eliminar redundancias en el uso de memoria RAM.

- (c) La sobrecarga sólo puede darse en métodos puesto que en los constructores al llamarse igual que el nombre de la clase no hay problema de conflicto.
- (d) *** La sobrecarga se da tanto en constructores y métodos. Mismo nombre de método/constructor con diferente tipo y/o número de parámetros.

5.- Dada la siguiente estructura de código, sabiendo que la clase Triángulo hereda de la clase Figura y que Triangulo especializa e implementa el método *dibujar()* ¿Qué sentencia hay que incorporar en (1) para que funcione el código?

```
Figura f = new Triangulo();
if (f instanceof Triangulo)
{
    (1)
    t.dibujar();
}
```

- (a) No hay que indicar nada.
- (b) `f = (Triangulo) t;`
- (c) ***

`Triangulo t = (Triangulo) f;`

- (d) `Triangulo t = f;`

6.- Una clase ¿podría implementar un constructor privado?:

- (a) No; el compilador de Java da un aviso indicando que no se puede hacer.
- (b) No; porque no se pueden crear instancias con un constructor privado.
- (c) *** Sí; no hay nada en Java que lo limite.
- (d) Sí; si es estático.

7.- Dada la clase *Triangulo* que hereda de la clase *Figura* y el siguiente código: *Figura f = new Triangulo();*, si el método *toString()* está redefinido en la clase *Triangulo*. ¿Cuál de las siguientes sentencias es la correcta?:

- (a) *** Se ejecutará el método *toString()* proporcionado por la clase *Triangulo*.
- (b) No se llamará a ningún método puesto que el compilador de Java da error.
- (c) Se ejecutará el método *toString()* proporcionado por la clase *Figura*.
- (d) Dependiendo del momento de ejecución se llamará al *toString()* de la clase *Figura* o al de la clase *Triangulo*.

8.- Un caso de uso de modificador final es:

- (a) *** Definición de constantes.
- (b) Terminar la ejecución del programa.

- (c) Terminar la implementación de una clase.
- (d) Terminar la implementación de un método.

9.- Para llamar de un constructor a otro dentro de la misma clase (seleccione la respuesta correcta):

- (a) Debe utilizarse *super(...)* implementado en cualquier lugar del constructor.
- (b) *** Debe utilizarse *this(...)* como primera sentencia del constructor.
- (c) Debe utilizarse *super(...)* como primera sentencia del constructor.
- (d) Debe utilizarse *this(...)* implementado en cualquier lugar del constructor.

10.- Indique la afirmación correcta. De una clase abstracta:

- (a) No se pueden heredar otras clases.
- (b) Se puede tener una única instancia de objeto en la aplicación.
- (c) *** En Java debe contener al menos un método abstracto.
- (d) Se pueden crear instancias de objetos.