

Examen de Ejemplo

Programación Orientada a Objetos (Grado en Ingeniería Informática y Grado en Ingeniería en Tecnologías de la Información - UNED)

Parte Teórica (2,5 puntos). La respuesta correcta se indica en negrita. Las respuestas correctas suman un ítem (+1). Las respuestas erróneas restan un cuarto de ítem (-0,25). La nota final de la parte teórica se calcula dividiendo el valor máximo (2,5 puntos) multiplicado por el número de ítems conseguidos entre el número total de preguntas del test. Para superar la parte teórica será necesario obtener al menos 1,25 puntos.

1. El tipo de los siguientes literales: `"333"` `333` `3.33` `true` es:
- a. String String int boolean
 - b. String int double boolean**
 - c. String int double String
 - d. String int double string
2. ¿Cuál de los siguientes fragmentos de código compilará sin errores? :

1 <pre>int i=0; if(i) { System.out.println("Hello"); }</pre>	2 <pre>boolean b=true; boolean b2=true; if(b==b2) { System.out.println("So true"); }</pre>
3 <pre>int i=1; int j=2; if(i==1 j==2) System.out.println("OK");</pre>	4 <pre>int i=1; int j=2; if(i==1 & j==2) System.out.println("OK");</pre>

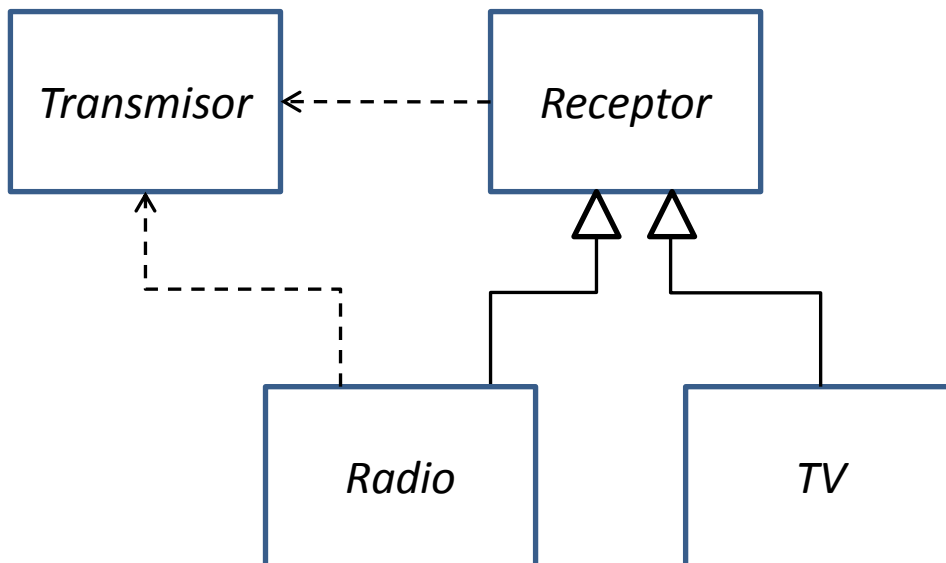
- a. 1 y 2
 - b. 1, 2 y 3
 - c. 2 y 4
 - d. 2 y 3**
3. Dado el siguiente código:

```
public class Pass{
static int j=20;
public static void main(String[] args){
int i=10;
Pass p = new Pass();
p.ametodo(i);
System.out.println(i);
System.out.println(j);
}

public void ametodo(int x){
x=x*2;
j=j*2;
}
}
```

- ¿Que valores imprimirá por pantalla al ser ejecutado?
- a. Error: el parámetro de *ametodo* no coincide con la variable
 - b. 20 y 40
 - c. 10 y 40**

- d. 10 y 20
4. ¿Cuál es el resultado de $(n\%6)$ para cualquier n entero?
- 0, 1, 2, 3, 4, 5
 - 5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5**
 - 36
 - 6, 0, -6
5. ¿Puede una clase contar cuántas instancias han sido creadas de dicha clase?
- Depende de la clase.
 - No.
 - Sí, usando un método en la librería System.
 - Sí, modificando el constructor de la clase.**
6. Indique la respuesta correcta. Acerca del concepto de sobrescritura en orientación a objetos:
- Una subclase no puede sobrescribir la implementación de un método.
 - El método sobrescrito no tiene precedencia cuando se invoca sobre objetos de la subclase.
 - La subclase puede declarar un método con la misma signatura que la superclase pero con un cuerpo diferente.**
 - Ninguna de las respuestas anteriores es correcta.
7. Dado el siguiente diagrama:



Indique cual de las siguientes declaraciones correspondientes a Radio y TV son correctas:

- public class Radio implements Receptor implements Transmisor { // ... }
public class TV extends Receptor { // ... }
- public class Radio extends Receptor implements Transmisor { // ... }
public class TV implements Receptor { // ... }
- public class Radio implements Receptor extends Transmisor { // ... }
public class TV extends Receptor { // ... }

d. **public class Radio extends Receptor implements Transmisor { //... }
public class TV extends Receptor { // ... }**

8. ¿Cuáles son los tres tipos estándar de diálogos que posee la clase de Swing JOptionPane?

- a. **Diálogo de mensaje, diálogo de confirmación y diálogo de entrada.**
- b. Diálogo de mensaje, dialogo de control y diálogo de entrada.
- c. Diálogo de confirmación, dialogo de control y diálogo de entrada.
- d. Diálogo de confirmación, diálogo de entrada y diálogo de salida.

9. En el siguiente fragmento de código hemos definido la ejecución de cinco bloques. Estos bloques se ejecutarán dependiendo de las excepciones que se produzcan en cada caso.

```
// Bloque 1  
try {  
    // Bloque 2  
} catch (Exception error) {  
    // Bloque 3  
} catch (ArithmeticException ae) {  
    // Bloque 4  
}  
// Bloque 5
```

Indique en qué caso se ejecutaría el bloque 4:

- a. Si se produce un acceso a un objeto nulo (null).
- b. **En ninguno de los casos anteriores.**
- c. Si no se produce ninguna excepción.
- d. Si se produce una excepción de tipo aritmético.

11. Indique la respuesta correcta. Una excepción...

- a. ... Es lanzada para indicar que ha ocurrido un fallo o que el programa ha finalizado.
- b. **... Es un objeto que representa los detalles de un fallo de un programa.**
- c. ... Permite que la aplicación termine normalmente a pesar de que el cliente fracase al manejarla.
- d. En una excepción el mecanismo de la excepción depende del valor de retorno de un método.

12. Si la clase Y es una subclase de la clase X. ¿Cuál de los siguientes códigos no compilará?

- a. X objeto=new X();
- b. **Y objeto=new X();**
- c. Y objeto= new Y();
- d. X objeto=new Y();

13. Si una variable calificada como final es una referencia a un objeto. ¿Cuál de las siguientes afirmaciones es falsa?

- a. Es posible modificar los miembros del objeto mientras el programa se está ejecutando.
- b. **Es posible modificar el objeto al que apunta dicha variable mientras el programa se está ejecutando.**

- c. No es posible modificar el objeto al que apunta dicha variable mientras el programa se está ejecutando.
 - d. Al contrario de lo que ocurre con las constantes, el valor de una variable final no tiene porque conocerse en tiempo de compilación.
14. ¿Cómo articula la interfaz Set la duplicidad de elementos?
- a. Lanza una excepción cuando se inserta un elemento duplicado.
 - b. El método add devuelve false cuando se inserta un elemento duplicado.**
 - c. Los elementos duplicados causan errores en tiempo de compilación.
 - d. De ninguna manera. La interfaz Set no impone restricciones respecto a la duplicidad de elementos.
15. ¿De qué modo es posible crear la lista {"a", "b", "c"}?
- a. `List<String> list = new ArrayList<String>(new String[]{"a", "b", "c"});`
 - b. `List<String> list = Arrays.asList(new String[]);`
 - c. `List<String> list = Collections.addAll(new ArrayList<String>(), "a", "b", "c");`
 - d. `List<String> list = Arrays.asList("a", "b", "c");`**

Parte Práctica (6,5 puntos)

El juego del Master-Mind consiste en que un jugador genera, de manera arbitraria, una combinación de cuatro colores en un orden determinado y que el otro jugador tiene que adivinar.

El número de colores en principio está fijo a seis (rojo, amarillo, verde, azul, morado y negro). En cada turno, el jugador que tiene que adivinar la combinación secreta propondrá una combinación posible de colores, y el jugador que ha generado la combinación secreta indicará, para cada color colocado por el primer jugador, si ese color está en la combinación secreta, y si lo está, si está colocado en su posición correcta o está en otra posición dentro de la combinación.

Supondremos que se dispone de 20 intentos para adivinar la combinación secreta y que en una combinación pueden aparecer colores repetidos.

Una posible implementación de este juego pasa por la definición de una clase `Combinacion`, que contiene los siguientes atributos:

```
final int NumColoresCombinacion = 4;      /* Número de colores de una combinación */
final int NumColoresJuego = 6;          /* Número total de posibles colores del juego */

public class Combinacion
{
    char[] Combi; /* Tabla de NumColoresCombinacion elementos que representa la combinación */
    char[] TablaColores={'R', 'A', 'V', 'Z', 'M', 'B'}; /* Posibles colores dentro del juego */

    /* ... Definiciones de Métodos ... */
}
```

1. (2,5 puntos) Implemente un método para la clase `Combinacion` que tenga el siguiente prototipo:

```
public void EvaluacionCombinacion (char []Combi, char []Resultado);
```

A este método se le pasa como parámetro una determinada combinación, y el método comprobará la combinación facilitada con la combinación secreta, devolviendo en `Resultado` la evaluación correspondientes de la combinación siguiendo el criterio expuesto en el enunciado.

2. (2,5 puntos) Proporcione un método *main* válido que muestre la lógica del juego del Máster-Mind, teniendo en cuenta que uno de los contrincantes será un jugador humano (introducimos sus datos y sus preferencias por el teclado) y el otro contrincante (que será el que genere la combinación secreta) será el propio ordenador (luego todos sus procesos tienen que hacerse automáticamente sin la intervención del usuario).
3. (1,5 puntos) Indiqué qué modificaciones son necesarias introducir en la aplicación para permitir que en lugar de combinaciones de 4 colores sean de 5 colores, se aumente el número de colores a 10 y se suprima la posibilidad de tener colores repetidos en la combinación.