

Programación de Sistemas  
Grado en Ingeniería de Sistemas Audiovisuales

1.- Indica qué devuelve el siguiente método recursivo para a=3.

```
public static int metodo(int a){
    int resultado = 0;
    if(a<0){
        resultado = -1;
    }else if(a>10){
        resultado = 1;
    }else{
        resultado = 2 + metodo(a+2);
    }
    return resultado;
}
```

- (a) \*\*\* 9
- (b) 1
- (c) La llamada recursiva se ejecuta indefinidamente.
- (d) 3

2.- Indica cuál de las siguientes afirmaciones sobre recursión es *correcta*.

- (a) La recursión en cascada implica dos métodos que se llaman entre sí.
- (b) \*\*\* La recursión no por la cola requiere realizar operaciones pendientes después de la última llamada recursiva.
- (c) La recursión anidada también se puede llamar lineal por la cola.
- (d) La recursión se utiliza porque normalmente es más eficiente que un bucle.

3.- Dado el siguiente código. ¿Que se imprimiría en pantalla al ejecutar el método main?.

```
public class ClaseA{
    private String a;
    public ClaseA(String a){this.a=a;}
    public String toString(){return a;}
}
public class ClaseB extends ClaseA{
    private String b;
    public ClaseB(String a, String b){
        super(a);
        this.b = b;
    }
    public String toString(){return b;}
}
public class TestPolimorfismo{
    public static void main(String[] args){
        ClaseA[] datos= {new ClaseA("hola"), new ClaseB("bye","adios")};
        for(int i=0; i<datos.length;i++){
```

```

        System.out.print(datos[i]+" ");
    }
}

```

- (a) adios adios
- (b) \*\*\* hola adios
- (c) bye adios
- (d) hola hola

4.- Dado el siguiente código. Cual será la salida tras ejecutar el método main.

```

public class TestIncremento{
    private static int a =2;
    private static int b =2;
    public static void main(String[] args){
        int a1 = 2 * a++;
        int b1 = 2 * ++b;
        System.out.println(a1 + " " + b1);
    }
}

```

- (a) Ninguna opción es correcta.
- (b) 6 6
- (c) 4 4
- (d) \*\*\* 4 6

5.- Dado el siguiente código y las variables `array = {'a', 'b', 'c'}` y `b=3`. ¿Qué valor tienen las variables `array` y `b` después de llamar al método `m(array, b)`.

```

public static void m(char[] array, int b){
    for(int i=0; i<array.length; i++){
        array[i] = 'x';
        b = 0;
    }
}

```

- (a) \*\*\* `array = {'x', 'x', 'x'}` `b= 3`
- (b) `array = {'x', 'x', 'x'}` `b = 0`
- (c) `array = {'a', 'b', 'c'}` `b = 0`
- (d) `array = {'a', 'b', 'c'}` `b = 3`

6.- Dado el siguiente código, indique cuál de las siguientes afirmaciones es *incorrecta*.

```

public class ClaseA{
    private String a;
    public ClaseA(int a){ this.a=a;}
    public ClaseA(){ this(null);}
}
public class ClaseB extends ClaseA{
    private String b;
    public ClaseB(){
        super();
        this.b = null;
    }
}

```

- (a) El constructor de `ClaseA` está sobrecargado.
- (b) La llamada a `super()` en `ClaseB` asigna el valor `null` al atributo `a`
- (c) \*\*\* El constructor de `ClaseB` está sobreescrito.
- (d) `this(null)` es una llamada a un constructor de la misma clase.

7.- Indique cual de las siguientes afirmaciones no es aplicable a pruebas de caja negra.

- (a) Las pruebas de caja negra también se pueden llamar pruebas de entrada / salida
- (b) Las clases de equivalencia definen subconjuntos disjuntos de posibles datos de entrada para pruebas de caja negra.
- (c) Las pruebas de caja negra permiten probar la funcionalidad del código.
- (d) \*\*\* La cobertura de ramas es un tipo de prueba de caja negra.

8.- Dadas las clases `Billete` y `BilleteDeTren` que hereda de la anterior y las interfaces `Vendible` y `VendiblePorInternet` que hereda de la anterior. Indique cual de las siguientes sentencias es *incorrecta*.

- (a) \*\*\* `BilleteDeTren b = (BilleteDeTren)new Billete();`
- (b) `BilleteDeTren b = new BilleteDeTren();`
- (c) `public class Billete implements Vendible, VendiblePorInternet{}`
- (d) `Billete b = new BilleteDeTren();`

9.- Indique cual de las siguientes afirmaciones sobre modificadores es *incorrecta*:

- (a) \*\*\* `private` aplicado a un paquete quiere decir que ese paquete sólo es accesible a la carpeta donde se encuentra la clase con el método `main`
- (b) `private` aplicado a un método de una clase quiere decir que sólo es accesible dentro de la propia clase
- (c) Si un atributo no tiene modificador sólo es accesible desde las clases del mismo paquete.
- (d) `static` aplicado a un atributo de una clase quiere decir que su valor es el mismo para todos los objetos de la clase.

10.- Dado el siguiente código indique cual de las siguientes sentencias de prueba para acceder al método `metodoA()` desde el `metodoB()` es incorrecta.

```
public class ClaseA{
    public void metodoA(){...}
}
public class ClaseB extends ClaseA{
    public void metodoB(){
        //sentencias de prueba
    }
}
```

- (a) `*** ClaseA.metodoA();`
- (b) `this.metodoA();`
- (c) `super.metodoA();`
- (d) `metodoA();`