

UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA – ESCUELA TÉCNICA SUPERIOR DE  
INGENIERÍA INFORMÁTICA  
71901072 – PROGRAMACIÓN ORIENTADA A OBJETOS (GRADO EN INGENIERÍA INFORMÁTICA /  
TECNOLOGÍAS DE LA INFORMACIÓN)  
SEPTIEMBRE 2018 – MODELO A – **NO ESTÁ PERMITIDO EL USO DE MATERIAL ADICIONAL**

**PARTE TEÓRICA - TEST [2,5 PUNTOS]:**

Solo 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.

Las preguntas de reserva sólo tendrán utilidad en el caso de que alguna de las 14 preguntas iniciales del test sea anulada por cualquier circunstancia. Caso de ocurrir este hecho, si se produjera la anulación de alguna de las 14 preguntas iniciales, la primera pregunta de reserva sustituiría a la pregunta anulada. Caso de que una segunda pregunta de las 14 iniciales fuese anulada, entonces la segunda pregunta de reserva sustituiría a esta segunda pregunta anulada. En aquellos hipotéticos casos en los que se produjese la anulación de una tercera o sucesivas preguntas de las 14 iniciales, entonces sólo en ese caso, las preguntas tercera y sucesivas anuladas se considerarían como correctas (al no existir más preguntas de reserva que las sustituyan).

**Pregunta 1:** Dado el siguiente fragmento de código:

```
float A = 3.3F;  
char B = 'z';  
int C = 3;  
  
System.out.println(A + B < 12);  
System.out.println(A >= 8 || C != 'w');  
System.out.println((C == 'c') && ((A + B) == 12));
```

Indica cual será la salida por pantalla (cada valor en una línea diferente):

- a. false true false
- b. true false true
- c. true false false
- d. false false false

**Pregunta 2:** ¿Qué significa el siguiente fragmento de código Java?: **String saludar() { return “Hola”; }**

- a. Hay un método “saludar” que no recibe ningún parámetro de entrada y devuelve una cadena cuyo valor es “Hola”.
- b. Hay un método “String saludar” que no recibe ningún parámetro de entrada y devuelve el valor “Hola”.
- c. Hay una variable “String” cuyo valor es “saludar() { return “Hola”; }”
- d. El fragmento no representa un fragmento de código legal en Java.

**Pregunta 3:** ¿Cuál de las siguientes sentencias se ejecuta de manera correcta?

- a. String coches [] = new String {"BMW" "AUDI" "SEAT"};
- b. String coches = {"BMW", "AUDI", "SEAT"};
- c. String coches [] = { "BMW" "AUDI" "SEAT"};
- d. String coches [] = {"BMW", "AUDI", "SEAT"};

**Pregunta 4:** Según el libro de la asignatura, la duplicación de código es un síntoma de:

- a. Buena cohesión.
- b. La solución inevitable de un problema complejo.
- c. Mala cohesión.
- d. Mal encapsulamiento.

**Pregunta 5:** Queremos compilar el siguiente código que se puede encontrar en el texto base de la asignatura y que ha sido modificado convenientemente. ¿Cuál es el resultado que obtenemos al compilar?

```
public class Prueba {
    public static void main (String args[]) {
        String cadena1 = new String("ejemPLO");
        String cadena2 = new String("ejemplo");
        cadena1 = cadena1.toLowerCase();
        if (cadena1.toString().equals(cadena2.toString())) {
            System.out.println("Son iguales");
        }
        else {
            System.out.println("Son diferentes");
        }
    }
}
```

- a. Se produce una excepción y la ejecución falla.
- b. Se imprime por pantalla el mensaje: Son diferentes.
- c. Se imprime por pantalla el mensaje: Son iguales.
- d. Ninguna de las anteriores.

**Pregunta 6:** Indica cual de las siguientes afirmaciones es correcta:

- a. A partir de una clase tan solo se puede crear un solo objeto.
- b. Los métodos pueden devolver información de algún objeto mediante un valor de retorno.
- c. Los métodos siempre tienen parámetros con los que obtener la información necesaria.
- d. El estado de los objetos se representa mediante los métodos implementados.

**Pregunta 7:** Dado el siguiente fragmento de código:

```
1. public class Suma {
2.     static int n;
3.     public static void main (String args []) {
4.         int j = 0;
5.         for (int j = 0; j++; j < 10) {
6.             if (n!=0) {
7.                 n = n + j;
8.                 System.out.println("El número es " + n);
9.             }
10.        }
11.    }
12. }
```

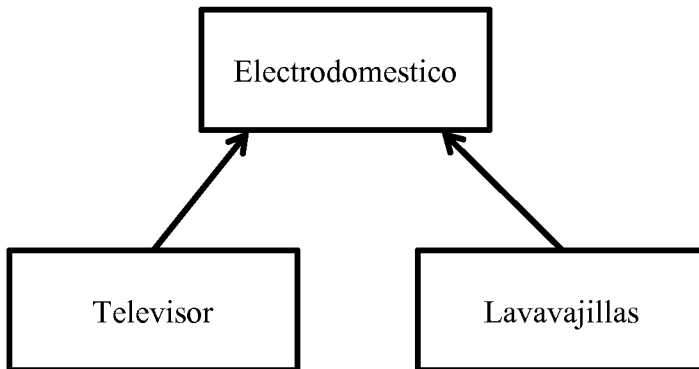
¿Cuál es la línea que provoca que el código produzca uno o varios errores de compilación?

- a. No se produce error de compilación
- b. En la línea 4
- c. En la línea 5
- d. En la línea 6

**Pregunta 8:** Indica cual de las siguientes afirmaciones es correcta:

- a. Un objeto de tipo String puede ser modificado una vez que está creado, por tanto no es un ejemplo de objeto inmutable
- b. Un objeto es inmutable si su contenido o su estado no puede ser cambiado una vez que se ha creado
- c. La clase String tiene un método de nombre trim que permite modificar caracteres en cualquier posición de una cadena
- d. Las cadenas de texto de tipo String solamente se pueden comparar mediante el operador “==”

**Pregunta 9:** Dada la siguiente jerarquía de herencia:



Indica cual de las siguientes asignaciones es correcta:

- a. Televisor t1 = new Electrodomestico();
- b. Lavavajillas l1 = new Electrodomestico();
- c. Electrodomestico e1 = new Lavavajillas();
- d. Lavavajillas l1 = new Televisor();

**Pregunta 10:** Un método de acceso o selector:

- a. Permite acceder al constructor de la clase que lo define.
- b. Habitualmente devuelve void.
- c. Devuelve siempre información sobre el estado de un objeto.
- d. Devuelve siempre un objeto de la clase Object.

**Pregunta 11:** Respecto a las clases internas ...

- a. Las instancias de la clase interna no están necesariamente asociadas a instancias de la clase circundante.
- b. Presentan un acoplamiento muy estrecho con la clase circundante.
- c. No se consideran una parte de la clase circundante.
- d. No pueden acceder a los métodos privados de la clase circundante.

**Pregunta 12:** Sea el siguiente fragmento de código modificado de la clase MailItem mostrada en el libro de texto:

```
1    public class MailItem {
2        static int num1 = 10;
3        public static void main (String args []) {
4            int num2 = 5;
5            new MailItem ();
6        }
7        public MailItem () {
8            int aux = this.num1;
9            if (aux > 1) {
10                System.out.println(aux);
11            }
12        }
13    }
```

¿Cuál es el resultado que produce?

- a. Se produce un error de compilación.
- b. Se produce un error de ejecución.
- c. No produce ningún error pero no muestra nada por pantalla.
- d. No se produce ningún error y muestra por pantalla el valor 10.

**Pregunta 13:** Si llamamos al método toUpperCase() de una instancia (nombre) de la clase String de la siguiente forma:

```
patata.toUpperCase();
```

¿Cuál sería el resultado de compilar y/o ejecutar el código?

- a. El código generaría un error de ejecución.
- b. El código convertiría la cadena nombre en mayúsculas.
- c. El código generaría un error de compilación.
- d. Ninguna de las anteriores.

**Pregunta 14:** Dado el siguiente fragmento de código:

```
1.  public class Suma {
2.      static int n;
3.      public static void main (String args []) {
4.          for (int j = 0; j < 10; j++) {
5.              if (n!=0) {
6.                  n = n + j;
7.                  System.out.println("El número es " + n);
8.              }
9.          }
10.     }
11. }
```

¿Cuál es la línea que provoca que el código produzca uno o varios errores de compilación?

- a. En la línea 2
- b. En la línea 5
- c. No se produce error de compilación
- d. En la línea 6

**RESERVA 1:** Según el texto de la bibliografía básica de la asignatura, ¿qué se entiende por abstracción?

- a. Capacidad de ignorar los detalles de las distintas partes, para centrar la atención en un nivel superior de un problema.
- b. Capacidad de dividir un todo en partes bien definidas ignorando los detalles de las distintas partes.
- c. Proceso de dividir un todo en partes bien definidas que puedan construirse y examinarse por separado y que interactúen de formas bien definidas.
- d. Ninguna de las anteriores.

**RESERVA 2:** Dado el siguiente fragmento de código que pretende mostrar un ejemplo de sobrescritura:

```
class Examen {
    private float pregunta = 1.0f ;
    protected float getNota () {return pregunta;}
}

class Test extends Examen {
    private float nota = 2.0f;
    //Insertar código aquí
}
```

Indique cual de las siguientes opciones completaría el código anterior para dar lugar a un ejemplo correcto de sobrescritura:

- a. public float getNota (float valor ) { return valor;}
- b. float getNota ( ) { return nota;}
- c. float double getNota ( ) { return nota;}
- d. public float getNota ( ) { return nota;}

## **PARTE PRÁCTICA [6,5 PUNTOS]:**

La Práctica del presente curso va a consistir en diseñar e implementar un sistema integrado de gestión de una tienda de electrodomésticos. Hoy en día las tiendas de electrodomésticos además de disponer de una gran cantidad de productos en sus tiendas, disponen de diferentes perfiles de empleados (técnicos, cajeros, financiación y postventa) para atender de la mejor manera posible a sus clientes. Además, los clientes disponen de un perfil que además de sus datos personales incluyen un historial de compras, generación de facturas, descarga de manuales, comprobación de estado de garantía y promociones. De esta forma, la práctica consiste en desarrollar un sistema de gestión que englobe todas estas características teniendo en cuenta un diseño orientado a objetos.

En general, las funciones que tienen un sistema de gestión de una tienda de electrodomésticos son varias:

- Venta de un electrodoméstico (cajero): cuando un cliente pasa por la línea de cajas es necesario generar una ficha de cliente en el caso de que no disponga de ella. El identificador principal es el DNI y los datos más importantes son el nombre, apellidos, dni, domicilio y número de teléfono. Esta ficha tendrá disponible un histórico de los productos comprados y su fecha de adquisición. En el caso de solicitar financiación, deberá constar en la ficha y el cliente debería pasar por la oficina de financiación para obtener el visto bueno.
- Financiación (financiación): El empleado de la oficina de financiación recibirá clientes que previamente hayan pasado por la línea de cajas para comprar productos y analizará la ficha de financiación. Solicitará la última nómina al cliente, dejando constancia de la cantidad en la ficha del cliente y en caso

de que el cargo mensual no supere el 15% de la nómina en un máximo de financiación de 60 meses, la financiación se aprobará.

- Reparación de Electrodomésticos (técnico): Los clientes podrán llevar sus productos comprados en la tienda a reparar. Las condiciones de reparación serán las siguientes: reparación gratuita en los dos primeros años. A partir de esa fecha, se cargará un importe al cliente dependiendo de la reparación efectuada.
- Devolución de electrodoméstico (postventa): Un cliente, presentando su DNI, podrá devolver uno o varios electrodomésticos en el caso de que el periodo de compra no supera los 3 meses.
- Gestión comercial (comercial): Este empleado generará una serie de comunicaciones con el cliente ofreciendo diferentes posibilidades de compra.
- Gestión de usuarios: altas, bajas, modificaciones de las personas que figuran en el sistema (empleados - técnicos, cajeros, financiación, postventa y comerciales -- y clientes). La primera vez que acude un cliente a la tienda hay que darle de alta en el sistema.
- Gestión de clientes: Cada uno de los empleados tendrán un tipo de relación con el cliente teniendo que dejar constancia en la ficha del cliente la operación realizada y sus detalles (cliente, empleado que atiende, tipo de operación, productos involucrados, fecha, etc.)

Se pide realizar las siguientes tareas:

- a) **[1,0 puntos]** Diseñar utilizando un paradigma orientado a objetos, los elementos necesarios para la aplicación explicada de la práctica durante el curso. Es necesario identificar la estructura y las relaciones de herencia (mediante el uso de un diagrama de clases) y de uso de las clases necesarias para almacenar y gestionar esta información. Debe hacerse uso de los mecanismos de herencia siempre que sea posible. Se valorará un buen diseño que favorezca la reutilización de código y facilite su mantenimiento.
- b) **[1,5 puntos]** Implementar los métodos que gestionan las altas, bajas y modificaciones de las personas que figuran en el sistema (empleados -- técnicos, cajeros, financiación, postventa, comerciales y clientes). La primera vez que acude un cliente a la tienda hay que darle de alta en el sistema.
- c) **[2,0 puntos]** Implementar un método (o métodos) que desarrolle la reparación de los electrodomésticos. El sistema tendrá que recuperar los datos de venta del aparato. Se almacenará información sobre la avería. Si se trata de algo pequeño el cliente puede llevarlo a la tienda para la reparación. Si es más grande tendrá que ir un técnico a su casa. Si el aparato está en garantía de dos años se hará la reparación de una forma gratuita. De otra manera, el cliente tendrá que pagar.
- d) **[2,0 puntos]** Implementar un método (o métodos) que permita(n) asignar a los usuarios clientes ofertas en función de las compras realizadas anteriormente. De este modo, los clientes que hayan gastado en sus últimas compras más de 999 euros recibirán un descuento de 100 euros en su próxima compra; los que hayan gastado más de 1.999 euros recibirán un descuento de 200 euros y los que hayan gastado más de 2.999 euros recibirán un descuento de 300 euros. Una vez aplicado el descuento, las compras tenidas en cuenta para la oferta no deberán volver a ser consideradas. Justifíquense las opciones y decisiones que se tomen.