

Examen Diciembre 2013

Programación Orientada a Objetos

No se permite el uso de material en clase. Duración 2 horas.

Pregunta 1 (15 ptos) - Múltiple Opción: Pueden haber más de 1 opción correcta y siempre existe por lo menos 1 también incorrecta. Indique solamente las opciones que considere correctas.

1.1 En lo que concierne al concepto de encapsulación:

- A. Es un mecanismo que permite integrar los comportamientos y estado de los objetos, haciéndolos solo accesibles a través de mensajes.
- B. Es un mecanismo esencial para la inmutabilidad de objetos.
- C. Es un mecanismo que permite a un objeto reutilizar los comportamientos de las superclases.
- D. Es un mecanismo que es parte de los principios S.O.L.I.D.

1.2 En lo que concierne a los conceptos de igualdad e identidad:

- A. Si dos objetos son idénticos, también son iguales. Pero si son iguales, no significa que sean idénticos.
- B. Si dos objetos son idénticos, no necesariamente tienen que ser iguales, dado que no comparten la misma dirección de memoria.
- C. Si dos objetos son iguales, entonces son idénticos, dado que tienen el mismo estado.
- D. No existe una relación entre objetos idénticos e iguales, dado que son conceptos diferentes.

1.3 En lo que concierne a métodos de clase e instancia:

- A. La ejecución de un método de instancia se realiza dentro del contexto de una instancia de una clase.
- B. La ejecución de un método de instancia siempre se realiza al momento instanciar una clase.
- C. La ejecución de un método de clase se ejecuta dentro de una clase, dado que accede directamente al estado de la misma.
- D. Un ejemplo de método de clase es el constructor de la clase.

1.4 En lo que concierne al concepto de Patrón Experto, qué beneficios nos brinda?

- A. Se mantiene el encapsulamiento, los objetos utilizan su propia información para llevar a cabo sus tareas.
- B. Se distribuye el comportamiento entre las clases que contienen la información requerida.
- C. Las clases son más fáciles de entender y mantener.
- D. Nuestro sistema queda más claro, dado que tenemos que escribir menos código.

1.5 En lo que concierne a los conceptos de pre y pos condiciones:

- A. Una precondición es una obligación para el objeto que usa la operación y un derecho para el objeto que la implementa.
- B. Una precondición es un derecho para el objeto que usa la operación y una obligación para el objeto que la implementa.
- C. Una poscondición es un derecho para el objeto que usa la operación y una obligación para el objeto que la implementa.
- D. Una poscondición es una obligación para el objeto que usa la operación y un derecho para el objeto que la implementa.

Pregunta 2 (10 ptos)

Sea el siguiente código

```
public class ClaseA {  
    private String atributo1;  
    public ClaseA (String a) {  
        atributo1 = a;  
    }  
    public void Metodo () {  
        atributo1 = "Un valor";  
    }  
}
```

Si quisiera hacer inmutable una instancia de la ClaseA, qué cambios le haría a la definición de la misma?. Realice dichos cambios.

Pregunta 3 (15 ptos)

Sea el siguiente código:

```
public class Living {  
    int cantSillas;  
    string sillon;  
    string marcaTV;  
    string dimensionesTV;  
    void encenderTV() {}  
}  
public class Comedor {  
    int cantSillas;  
    string marcaTV;  
    string dimensionesTV;  
    void encenderTV() {}  
}
```

Crítica el código anterior y realiza los cambios pertinentes en términos de buenas prácticas en programación orientada a objetos.

Pregunta 4 (30 pts)

Un extraño cliente nos pide realizar un programa que represente un árbol de navidad. Si bien pudo habernos dado las especificaciones del mismo, no tuvo idea mejor que darnos una imagen del mismo:



El programa deberá poder encender / apagar el árbol y agregarle componentes decorativos. Tener en cuenta que algunos componentes decorativos pueden encenderse y obviamente, como cualquier componente de decoración, tiene un color asociado.

Es necesario realizar un programa principal (Program) capaz de mostrar el funcionamiento del árbol.

Pregunta 5 (30 pts)

Papa Noel, cansado ya de entregar personalmente los regalos, tuvo que tercerizar el servicio de entrega. Además, no tuvo mejor idea que dedicarse a programar en sus ratos libres para controlar su servicio, por lo que nos envía el siguiente código fuente para que lo critiquemos:

```
public class Regalo {
    public String Continente;
}

public class DHL {
    public boolean AutorizarProducto(Regalo r) {
        /* Indica si se puede enviar este producto o no */
    }
    public void Enviar (Regalo r) { /* Se envia el producto */ }
}

public class TNT {
    public void Llevar (Regalo r) { /* Se envia el producto */ }
}

public class FedEx {
    public void Send (Regalo r) { /* Se envia el producto */ }
}

public class Itinerario {
    public ArrayList regalos;
    public DHL dhl;
    public TNT tnt;
    public FedEx fedex;
    public void Agregar(Regalo r) {
        regalos.Add(r);
    }
    public void enviarPorDHL(Regalo r) {
        dhl.AutorizarProducto(r);
        dhl.Enviar(r);
    }
    public void enviarPorTNT(Regalo r) {
        tnt.Llevar(r);
    }
    /*Si no se pudo enviar algun regalo, debe retornar falso*/
    public boolean ComenzarProduccion() {
        foreach (Regalo r in regalos) {
            if (r.Continente == "America") {
                enviarPorDHL(r);
            }
            if ((r.Continente == "Africa") || (r.Continente == "Europa")) {
                fedex.Send(r);
            }
            if ((r.Continente == "Oceania") || (r.Continente == "Asia")) {
                enviarPorTNT(r);
            }
        }
        return true;
    }
}
```

5.1) Critica el código anterior en base a los principios S.O.L.I.D vistos en clase, encapsulación y buenas prácticas de programación.

5.2) Realiza todos los cambios indicados en tus críticas para ayudar a Papa Noel.