

Examen Julio 2014

*Las estructuras de datos y los algoritmos funcionan bien juntos
porque ninguno sabe nada del otro*
Alex Stepanov

No se permite el uso de material en clase. Duración de 1hr 40 mins.

Pregunta 1 (15 ptos)

```
class Jugador {
    public Nombre {get; set;}
    public void Jugar() { Console.WriteLine("Pasando la pelota"); }
}
class Golero : Jugador {
    public void Jugar() { Console.WriteLine("Atajando la pelota"); }
}
```

Realice las modificaciones que crea conveniente al código anterior para dar un ejemplo del uso de encadenamiento estático y otro dinámico. Ejemplifique con un programa principal.

Pregunta 2 (30 ptos)

Una tienda de música tiene una serie de productos (CDs, DVDs, Libros, etc) los cuales comercializa. Para ello, la empresa recibe dichos productos, los pone en stock para luego venderlos.

1. Realiza un modelo de clases capaz de representar este escenario, indicando las relaciones entre las entidades encontradas. Ten en cuenta que el modelo debe ser genérico para cualquier tipo de producto que pueda ingresar a la tienda.
2. Desarrolla el código fuente de las entidades encontradas, capaz de poder ingresar productos al stock, venderlos (la venta de un producto lo saca del stock) y consultar la cantidad de ventas realizadas hasta el momento.
3. Desarrolla un programa principal capaz de probar lo desarrollado en el punto anterior.

Pregunta 3 (20 ptos)

Sea el siguiente código

```
public class MSOffice {
    public void EjecutarExcel();
}
public class OpenOffice {
    public void EjecutarCalc();
}
public class SistemaOperativo {
    public void EjecutarPlanillaElectronica (Object algo, int tipo) {
        if (tipo == 1) { (MSOffice) algo.EjecutarExcel(); }
        if (tipo == 2) { (OpenOffice) algo.EjecutarCalc(); }
    }
}
```

1. Crítica el código anterior en base a los conceptos vistos en clase, enfatizando en los principios S.O.L.I.D. Justifica cada una de las críticas.
2. Modifica el código anterior según las críticas del punto anterior.

Pregunta 4 (20 ptos)

Sea el siguiente código

```
public class CantantePop {
    String nombre;
    String apellido;
    void Cantar() { /* */;
    void SolicitarManicura() { /* */ }
}

public class CantanteRock {
    String nombre;
    String apellido;
    void Cantar() { /* */;
    void SaltarAlPublico() { /* */;
}
```

1. Critica el código anterior en términos de buenas prácticas en programación orientada a objetos.
2. Realiza los cambios en el código pertinentes a tus críticas en el punto anterior (**no es posible usar interfaces**).

Pregunta 5 (15 ptos)

Sea el siguiente código

```
public class Animal {
    void Comer();
}

public class Perro : Animal {
    public void Ladrar() { /* */ }
}

public class Gato : Animal {
    public void Maullar() { /* */ }
}
```

1. Indica y **justifica** cuáles líneas son **incorrectas**.

```
/*1*/ Animal a1 = new Animal();
/*2*/ Animal a2 = new Perro();
/*3*/ Animal a3 = new Gato();
/*4*/ a2 = a3;
/*5*/ Gato b = new Gato();
/*6*/ b = a3;
/*7*/ a2 = b;
/*8*/ Perro p = new Animal();
```

2. Modifica el código anterior para que los perros y gatos puedan comer de forma diferente.