

Programación orientada a objetos. Primer parcial 2º semestre 2009.
Martes 8 de setiembre de 2009.

1. Sean las siguientes clases e interfaces, y las siguientes asignaciones:

```
public interface IInstrumento{
    void Tocar();
}
public interface IMusical: IInstrumento {
    void Producir();
}
public interface IMedicion: IInstrumento {
    void Medir();
}
public interface IViento:IMusical {
    String TipoViento {get;set;}
}
public interface ICuerda:IMusical {
    String TipoCuerda {get;set;}
}
public interface IMaterial{}
public interface IMadera: IMaterial {
    String Color {get;set;}
    Int32 Grosor {get;set;}
}
public interface IMetal: IMaterial {
}
public interface ICristal: IMaterial {
}
public class Flauta: IViento, IMetal {
}
public class Arpa: ICuerda, IMadera {}
public class Guitarra: ICuerda, IMadera {
    Int32 CantidadCuerdas {get;set;}
}
public class Balanza: IMedicion{
    Int32 PesoMax {get;set;}
}
public class Termometro: IMedicion, ICristal, IMadera{
    Int32 TempMax {get;set;}
}
class Program {
    static void Main(string[] args) {
        /*1*/ ICuerda g = new Guitarra();
        /*2*/ IMusical m = new IMusical();
        /*3*/ Flauta f = new IViento();
        /*4*/ Arpa a = g;
        /*5*/ IMaterial b = new Balanza();
        /*6*/ IMedicion u = b;
        /*7*/ ICristal t = new Termometro();
        /*8*/ IMadera i = t;
    }
}
```

1.1 Indiquen las líneas de código incorrectas y justifiquen (**muy brevemente**) su decisión.

2. Basándote exclusivamente en las líneas correctas del ejercicio anterior, responde las siguientes preguntas:

2.1 ¿Qué tipos tienen los objetos referenciados en las variables definidas en las líneas correctas? Para cada variable, indica los tipos.

2.2 ¿Qué mensajes puede recibir un objeto cómo el creado en la línea 1? ¿Cómo lo sabes?

2.3 ¿Qué mensajes puede recibir un objeto bajo los términos de la variable declarada en la línea 6? ¿Cómo lo sabes?

- 3.1. ¿Qué es un objeto inmutable? ¿Cómo puedes asegurarte, al programar una nueva clase, que los objetos creados por la misma sean inmutables?
- 3.2. Programa un fragmento de código que genere por lo menos 2 objetos y por lo menos 3 variables que los referencien.
- 3.3 Indica qué variables referencian al mismo objeto. Si no existe el caso, généralo agregando nuevas líneas de código. Indica cuáles son.
- 3.4. ¿Hay objetos iguales al final de tu código? Si no los encuentras, agrega las líneas necesarias para que encuentres por lo menos 2 objetos iguales.

4. 1 Provee un ejemplo en C# donde utilices polimorfismo.

4.2 En un lenguaje en el que no puedas definir el tipo de las variables ni existan las interfaces, ¿puedes utilizar polimorfismo?. Justifica tu respuesta.

5 ¿Qué te permiten hacer las interfaces que no te permiten hacer las clases? ¿Qué te permiten hacer las clases que no te permiten hacer las interfaces?

6. Sea el siguiente código:

```
public class Reloj {
    private Int32 id;
    public Int32 IdReloj {
        get { return id; }
        set { id = value; }
    }
    public void Registrar(Empleado e) {
        Incidencia i = new Incidencia();
        i.ProcesarIncidencia(e, this);
    }
}

public class Tarjetero {
    private Int32 id;
    public Int32 IdTarjetero {
        get { return id; }
        set { id = value; }
    }
    public void Marcar(Empleado e) {
        Incidencia i = new Incidencia();
        i.ProcesarIncidencia(e, this);
    }
}

public class Empleado {
    private readonly Int32 idEmpleado;
    public Int32 IdEmpleado { get { return idEmpleado; } }
    private readonly Int32 idUnidad;
    public Int32 IdUnidad { get { return idUnidad; } }
    private readonly Int32 idSeccion;
    public Int32 IdSeccion { get { return idEmpleado; } }
    public Empleado(Int32 idEmp, Int32 idUni, Int32 idSec) {
        idEmpleado = idEmp;
        idUnidad = idUni;
        idSeccion = idSec;
    }
}

public class Incidencia {
    private String clave;
    public String Clave { get { return clave; } }
    public void EnviarIncidencia(String claveIncidencia) {
        /* Envía la incidencia nueva al la base de datos */
    }
    public void ProcesarIncidencia(Empleado e, Object medidor) {
```

```
        if (medidor is Reloj) {
            PorReloj(e, (Reloj)medidor);
        } else {
            PorTarjeta(e, (Tarjetero)medidor);
        }
    }
    private void PorReloj(Empleado e, Reloj r) {
        clave = "" + e.IdEmpleado + e.IdUnidad + e.IdSeccion +
            r.IdReloj;
        EnviarIncidencia(clave);
    }
    private void PorTarjeta(Empleado e, Tarjetero t) {
        clave = "" + e.IdEmpleado + e.IdUnidad + e.IdSeccion +
            t.IdTarjetero;
        EnviarIncidencia(clave);
    }
}
```

Crítica el código en base a:

6.1 Patrón experto

6.2 OCP

6.3 SRP

7. Modifica el código anterior para solucionar los problemas encontrados.