

Programación Orientada a Objetos

Primer Parcial 1er Semestre 2015

Viernes 17 de Abril del 2015

El parcial tiene una duración de 1 hora 40 mins. Terminado el parcial, se realiza un recreo de 10 minutos y se procederá a dar la solución en clase. **No se permite el uso de material.**

Pregunta 1 (20 ptos)

Sean las siguientes interfaces

```
public interface IProducto {
    int ObtenerPeso();
}

public interface IDulce {
    void DegustarDulce();
}

public interface ILacteo {
    int PorcentajeAgua();
}

public interface IQueso {
    int ObtenerCuajo();
}

public class Membrillo : IDulce, IProducto { /*Codigo*/ }
public class Queso : IQueso, IProducto { /*Codigo*/ }
public class QuesoDulce : IDulce, IQueso, IProducto { /*Codigo*/ }

public static void Main(String[] args) {
    /*1*/ IDulce d = new Queso();
    /*2*/ IProducto p = new Queso();
    /*3*/ IProducto m = new Membrillo;
    /*4*/ p = m;
    /*5*/ ILacteo l = new Queso();
    /*6*/ QuesoDulce qd = new IQueso();
    /*7*/ QuesoDulce h = new QuesoDulce();
    /*8*/ IQueso dambo = h;
}
```

- 1.1) Indique las líneas de código incorrectas y justifique brevemente.
- 1.2) ¿Qué tipos tienen los objetos creados en las líneas 2 y 7?
- 1.3) ¿Qué mensajes puede recibir un objeto como el creado en la línea 7?. ¿Cómo lo sabes?
- 1.4) ¿Qué mensajes puede recibir un objeto bajo los términos de la variable declarada en la línea 8? ¿Cómo lo sabes?
- 1.5) ¿Que contrato firmaron las variables declaradas en las líneas 2 y 7?

Pregunta 2 (10 ptos)

Teniendo en cuenta el ejercicio anterior, ahora suponga que tenemos un almacén. Para ello, nos fabricamos la siguiente clase:

```
public class Almacen {
    /*1*/ private ArrayList productos;
    public Almacen () {
    /*2*/         productos = new ArrayList();
    }

    public void AgregarProducto (IProducto prod) {
    /*3*/         productos.Add(prod);
    }

    public int GetPesoTotal() {
    /*4*/         int i = 0;
    /*5*/         foreach (IProducto p in productos) {
    /*6*/             i = i + p.ObtenerPeso();
    /*7*/             String str = new String();
    /*8*/             str = "Suma parcial: " + i;
    /*9*/             Console.WriteLine (str);
    }
    /*10*/        return i;
    }
}
```

2.1) ¿Qué entiende por polimorfismo de tipos?

2.2) ¿Indique en cuál/es línea/s del código se ejecuta polimórficamente un método? ¿Cómo lo/s detectó?

Pregunta 3 (15 ptos)

Dado el siguiente código

```
public class SmartWatch{
    private String marca;
    private String modelo;
    public String DarHora() { return new Date().ToString(); }
}

public class Program {
    public static void Main (String[] args) {
        // Código a implementar por el alumno.
    }
}
```

3.1) Modifica la clase SmartWatch y escribe en el método Main del Program.cs un programa que genere variables que referencian a objetos del tipo SmartWatch , que sean iguales e idénticos.

3.2) Realiza los cambios necesarios para que la clase SmartWatch quede inmutable.

Pregunta 4 (15 ptos)

Dada la siguiente clase

```
public class Celular {  
    private String numero;  
    private String color;  
    public Celular (String num) {  
        numero = num;  
        color = "Negro";  
    }  
    public static Int32 GetSIM() { return 1234; }  
    public void Llamar() {  
        Console.WriteLine("Llamando al numero " + numero);  
    }  
  
    public String GetNumero() {  
        return numero;  
    }  
}
```

4.1) Indica cuales son métodos de instancia y cuales son métodos de clase.

4.2) ¿Cuál es la diferencia entre ambos conceptos?

4.3) ¿Cómo se define la sobrecarga de métodos?. ¿Qué particularidad debe tener?.

4.4) En base a la clase anterior, ejemplifica la sobrecarga del método constructor y realiza una sobrecarga al método "llamar".

Pregunta 5 (40 ptos)

Realiza un programa capaz de almacenar empleados dentro de una empresa. Los empleados deberán tener un nombre, apellido, edad y cédula y la empresa deberá poder almacenarlos, imprimirlos y calcular la edad promedio de los mismos mediante métodos propios.

Recuerda encapsular las clases, **brindar las interfaces correctas** y un programa principal que muestre lo realizado.