

Programación Orientada a Objetos

Primer Parcial 2do Semestre 2014

Viernes 5 de Setiembre del 2014

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 IPersona {
    void Pensar();
}

public interface IExtranjero {
    void AprenderEspañol();
}

public interface IDeportista : IPersona {
    void Ejercitarse();
}

public interface IFutbolista : IDeportista {
    void HacerGoles();
}

public interface ITenista : IDeportista {
    void RomperRaqueta();
}

public interface IFutbolistaExtranjero : IFutbolista, IExtranjero {
    void TomarMate();
}

public class Cuevas : ITenista { /*Codigo*/ }
public class Federer : ITenista, IExtranjero { /*Codigo*/ }
public class Forlan : IFutbolista { /*Codigo*/ }
public class Henry : IFutbolistaExtranjero { /*Codigo*/ }

public static void Main(String[] args) {
    /*1*/ IDeportista federer = new Federer();
    /*2*/ IExtranjero nadal = federer;
    /*3*/ IPersona yo = federer;
    /*4*/ IFutbolista forlan = new Forlan();
    /*5*/ IFutbolista henry = new Henry();
    /*6*/ IFutbolistaExtranjero otroHenry = henry;
    /*7*/ Cuevas cuenvas = new ITenista();
    /*8*/ IFutbolista otroCuevas = new Cuevas();
}
```

- 1.1) Indique las líneas de código incorrectas y justifique brevemente.
- 1.2) ¿Qué tipos tienen los objetos creados en las líneas 1 y 5?
- 1.3) ¿Qué mensajes puede recibir un objeto como el creado en la línea 5?. ¿Cómo lo sabes?
- 1.4) ¿Qué mensajes puede recibir un objeto bajo los términos de la variable declarada en la línea 1? ¿Cómo lo sabes?
- 1.5) ¿Que contrato firmaron las variables declaradas en las líneas 3 y 4?

Pregunta 2 (10 ptos)

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

```
public class Liverpool {
/*1*/  private ArrayList jugadores;
    public Liverpool () {
/*2*/      jugadores = new ArrayList();
    }
    public void AgregarJugador (IFutbolista jugador) {
/*3*/      jugadores.Add(jugador);
    }
    public int HacerGoles() {
/*4*/      int i = 0;
/*5*/      foreach (IFutbolista j in jugadores) {
/*6*/          j.HacerGoles();
/*7*/          i++;
      }
/*8*/      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 Celular{
    private String marca;
    private String modelo;
}

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

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

3.2) Realiza los cambios necesarios para que la clase Celular 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 alumnos dentro de un salón. Los alumnos deberán tener un nombre, apellido y cédula y el salón deberá poder almacenarlos e imprimirlos, mediante un método propio de la clase, en consola. Recuerda encapsular las clases, brindar las interfaces correctas y un programa principal que muestre lo realizado.