

# Examen Febrero 2014

## Programación Orientada a Objetos

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

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

### Pregunta 1.1

- A. El polimorfismo permite que una misma definición pueda ser usada con diferentes tipos.
- B. El polimorfismo permite que una misma clase pueda ejecutar varias funcionalidades al mismo tiempo.
- C. El polimorfismo permite indicar a una clase que su estado queda intacto en el ciclo de vida de una instancia de la misma.

### Pregunta 1.2

- A. La herencia es una forma de delegación de comportamientos, dado que puedo colaborar con clases compuestas y delegar la responsabilidad a ellas.
- B. La herencia es un mecanismo de definición de contratos con otros tipos.
- C. La herencia es una forma de reutilización de código pues permite crear nuevas clases a partir de clases existentes.

### Pregunta 1.3

- A. El encadenamiento dinámico es necesario para soportar polimorfismo.
- B. El encadenamiento dinámico no es necesario para soportar polimorfismo, dado que se define en tiempo de compilación.
- C. En C#, no es posible indicar al compilador qué tipo de encadenamiento queremos utilizar en cada caso.

### Pregunta 1.4

- A. Una clase abstracta tiene que tener todos sus métodos abstractos, así no permito que se puedan crear instancia de dicha clase.
- B. Una clase que tiene un método abstracto es una clase abstracta.
- C. Una clase abstracta me permite definir métodos comunes que serán luego implementados por clases sucesoras.

### Pregunta 1.5

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

## Pregunta 2 (30 ptos)

Un taller mecánico tiene una serie de vehículos (automóviles y motos), los cuales debe reparar. Para ello, el taller recibe dichos vehículos, los repara y los devuelve.

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 el tipo de vehículos que ingresan en el taller (más adelante pueden reparar buses).
2. Desarrolla el código fuente de las entidades encontradas, capaz de poder ingresar vehículos al taller, repararlos y luego devolverlos.
3. Desarrolla un programa principal capaz de probar lo desarrollado en el punto anterior.

## Pregunta 3 (30 pts)

Sea el siguiente código

```
public class Libro {
    public void Leer();
}

public class Revista {
    public void Ojear();
}

public class Biblioteca {
    private ArrayList libros;
    private ArrayList revistas;

    public Object Buscar (Object algo, int tipo) {
        if (tipo == 1) {
            /* Se busca en el ArrayList de libros */
        }
        if (tipo == 2) {
            /* Se busca en el ArrayList de revistas */
        }
    }

    public void Leer (Object obj, int tipo) {
        if (tipo == 1) {
            (Libro) obj.Leer();
        }
        if (tipo == 2) {
            (Revista) obj.Ojear();
        }
    }
}
```

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 (15 ptos)

Sea el siguiente código

```
public class JugadorFutbol {
    String nombre;
    String apellido;
    void HacerCaños() { /* */;
}

public class JugadorTenis {
    String nombre;
    String apellido;
    void RomperRaqueta() { /* */;
}
```

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.

## Pregunta 5 (10 ptos)

Sea el siguiente código

```
public interface IMusico {
    void Tocar();
}

public class Guitarrista : IMusico {
    public void Tocar() { /* */ }
}

public class Baterista : IMusico {
    public void Tocar() { /* */ }
}
```

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

```
/*1*/ IMusico m1 = new IMusico();
/*2*/ IMusico m2 = new Guitarrista();
/*3*/ IMusico m3 = new Baterista();
/*4*/ m2 = m3;
/*5*/ Baterista b = new Baterista();
/*6*/ b = m3;
/*7*/ m2 = b;
```