

Programación orientada a objetos. Examen diciembre 2005.
Viernes 9 de diciembre de 2005.

1. ¿Puede un objeto tener un **tipo** que tenga más operaciones que el tipo definido por la **clase** de ese objeto? Justifiquen la respuesta. Programen un pequeño ejemplo en C# si fuera posible.
2. ¿Puede tener más de un tipo un objeto de una clase que no implementa una interfaz? Justifiquen la respuesta. Programen un pequeño ejemplo en C# si fuera posible.
3. Expliquen para qué son utilizadas las **clases abstractas** y hagan, además, un ejemplo simple en C# usando clases abstractas para mostrarlo. ¿Tiene sentido definir un método abstracto que sea privado? Justifiquen la respuesta.
4. Expliquen en forma clara y precisa cómo funciona el **encadenamiento estático** y el **encadenamiento dinámico** en esos lenguajes. ¿El concepto es aplicable a los lenguajes en los que no se declara el tipo de las variables y argumentos?

Muestren las diferencias entre encadenamiento estático y dinámico programando un pequeño ejemplo en C#.
5. Un objeto puede tener más de un **tipo** y varios objetos de diferentes clases pueden tener el mismo **tipo**.

Muestren con un pequeño programa en C#, usando las clases provistas, un ejemplo de un objeto con más de un tipo y otro con varios objetos de diferentes clases y el mismo tipo.
6. Dos de las tres formas más comunes de reutilización en programación orientada a objetos son **herencia** y **composición-delegación**. Ignorando la semántica de las relaciones de generalización-especialización y exclusivamente desde el punto de vista de la reutilización de código ¿puedo hacer lo mismo con composición-delegación que con herencia? Programen un pequeño ejemplo en C# si fuera posible.
7. Un objeto se manda un mensaje a sí mismo. ¿El método que se ejecuta como consecuencia tiene que estar implementado en la clase de ese objeto? Justifiquen la respuesta de forma breve y concreta.
8. Enuncie el **principio de sustitución de Liskov**. ¿Hay alguna relación entre el principio de sustitución de Liskov y el diseño por contrato?
9. Cuando un objeto cliente necesita la colaboración de un objeto servidor con cierto tipo... ¿Cuáles son las obligaciones del cliente respecto del tipo? ¿Y las del servidor? ¿Cuáles son los derechos del cliente? ¿Y los del servidor? Formulen las respuestas usando los términos mensaje, operación, método.

10. Sean las siguientes clases:

```
public class CompactDisc {
    public void playAll() {
        /* ... */
    }
}

public class VideoDisc {
    public void playMovie() {
        /* ... */
    }
}
```

Los métodos `CompactDisc.playAll()` y `VideoDisc.playMovie()` son semánticamente iguales pero tienen nombres diferentes porque han sido programados en distinto momento y por diferentes personas.

Un fragmento del programa que usa estas clases aparece a continuación:

```
IList s = new ArrayList();
s.Add(new CompactDisc());
/* Otros discos compactos son creados y agregados aquí... */

s.Add(new VideoDisc());
/* Otros videodiscos son creados y agregados aquí... */

foreach (Object o in s) {
    if (o is CompactDisc) {
        ((CompactDisc)o).playAll();
    }
    else if (o is VideoDisc) {
        ((VideoDisc)o).playMovie();
    }
}
```

Ahora su jefe le pone a usted a cargo de este programa y le pregunta si el fragmento está escrito usando buenas prácticas de programación orientada a objetos; en caso de que no sea así le pedirá que lo escriba de nuevo -el jefe no sabe programación orientada a objetos y confía en su respuesta o se hace el que no sabe para evaluar si usted sabe-.

- a. ¿Las clases y el fragmento están bien escritos o no? Justifique su respuesta.
- b. En caso que no lo esté escriba la versión correcta. Justifique sus decisiones.