

Programación orientada a objetos. Segundo parcial 2004.

Viernes 3 de junio de 2006.

Al igual que las demás instancias de evaluación este examen será calificado de acuerdo a la escala de letras vigente: D, R, B, MB, BMB, S o D, R, B, MB, S según la generación de cada alumno. Por eso las preguntas no tienen punto: todas valen lo mismos

Pongan el nombre sólo en la primera hoja. Las respuestas deben aparecer en el orden en que están formuladas las preguntas. Escriban sólo del anverso de la hoja y eviten doblarlas. Gracias y mucha suerte.

Sean las siguientes clases:

```
public interface ICoordinable
{
    // Código no mostrado
}

public class Coordinador
{
    public void Coordinar(ICoordinable objeto)
    {
        // Código no mostrado
    }
}

public class Proceso : ICoordinable
{
    public void Iniciar()
    {
        // Código no mostrado
    }
    public void Terminar()
    {
        // Código no mostrado
    }
    public void Dormir()
    {
        // Código no mostrado
    }
}

public class ProcesoCoordinado : Proceso
{
    public void Coordinar()
    {
        // Código no mostrado
    }
}
```

1. Programen nuevamente la clase `ProcesoCoordinado` -debe tener el mismo tipo que la clase provista- usando composición exclusivamente. No pueden completar los métodos poniendo “Código no mostrado” a diferencia del código provisto.
2. Programen nuevamente la clase `ProcesoCoordinado` -debe tener el mismo tipo que la clase provista- usando herencia y composición. No pueden completar los métodos poniendo “Código no mostrado” a diferencia del código provisto.

3. En el **encadenamiento estático** el método que se ejecuta como consecuencia de la recepción de un mensaje por un objeto se determina:
 - a. A partir la **variable o referencia** al objeto en **tiempo de compilación**.
 - b. A partir de la **clase** del objeto en **tiempo de compilación**.
 - c. A partir la **variable o referencia** al objeto en **tiempo de ejecución**.
 - d. A partir de la **clase** del objeto en **tiempo de ejecución**.
4. En el **encadenamiento dinámico** el método que se ejecuta como consecuencia de la recepción de un mensaje por un objeto se determina:
 - a. A partir la **variable o referencia** al objeto en **tiempo de compilación**.
 - b. A partir de la **clase** del objeto en **tiempo de compilación**.
 - c. A partir la **variable o referencia** al objeto en **tiempo de ejecución**.
 - d. A partir de la **clase** del objeto en **tiempo de ejecución**.
5. ¿Puedo prescindir del **encadenamiento estático** en un lenguaje de programación orientada a objetos? ¿Y del **encadenamiento dinámico**? ¿Qué se pierde en cada caso? Respondan en forma breve y concreta.
6. ¿Puede haber **encadenamiento estático** cuando el tipo de una variable está declarado en una interfaz? Justifiquen la respuesta en forma breve y concreta.
7. Sean las siguientes clases:

```
public class Ancestro
{
    public void Método()
    {
        Console.WriteLine("Yo soy el ancestro");
    }
}

public class Sucesor:Ancestro
{
    public void Método()
    {
        Console.WriteLine("El sucesor soy yo");
    }
}

class Programa
{
    static void Main()
    {
        Ancestro b = new Sucesor();
        b.Método();
    }
}
```

El programa compila y es correcto. Al ejecutar la clase `Programa` aparece “Yo soy el ancestro” en la consola. Hagan la menor cantidad de cambios en la menor cantidad de clases para que aparezca “El sucesor soy yo”. ¿Qué cambiaría si no pudieran modificar la clase `Programa`?

8. 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 en forma breve y concreta.