

Programación orientada a objetos. Primer parcial 1º semestre 2007.
Sábado 21 de abril 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 puntos.

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.

Sea el siguiente programa:

```
interface InterfazUno
{
    String Propiedad { get; set; }
    String MetodoUno(String parametro);
}

class ClaseUno: InterfazUno
{
    private String p;
    public String Propiedad { get { return p; } set { p = value; } }
    public ClaseUno( String propiedad) { p = propiedad; }
    public ClaseUno() {}
    public Int32 MetodoUno(Int32 parametro) {}
}

class Program
{
    static void Main(string[] args)
    {
        /*1*/ ClaseUno a = new ClaseUno("X");
        /*2*/ ClaseUno b = new ClaseUno();
        /*3*/ ClaseUno c = new ClaseUno();
        /*4*/ b.Propiedad = "X";
        /*5*/ InterfazUno d = b;
    }
}
```

- 1.1 ¿Cuándo dos objetos son iguales?
- 1.2 En el código anterior ¿qué variables contienen o referencian objetos iguales al final del programa? Pongan el nombre de las variables y la línea donde son creados los objetos.
- 1.3 En el código anterior ¿qué variables contienen o referencian al mismo objeto? Pongan el nombre de las variables y la línea donde son creados los objetos.
- 1.4 Supongan que se agrega la siguiente línea al final del método `static void Main(string[] args)` anterior:

```
/*6*/ d.Propiedad = "Y";
```

Luego de ejecutar esta operación, ¿qué variables contienen o referencian a objetos iguales?

Sean las clases e interfaces del programa anterior más las siguientes con el nuevo programa dado:

```
interface InterfazDos: InterfazUno
{
    String MetodoDos(String parámetro);
}

class ClaseDos: InterfazDos
{
    private String p;
    public String Propiedad { get { return p; } set { p = value; } }
    public ClaseDos(String propiedad) { p = propiedad; }
    public ClaseDos() {}
    public String Metodo1(String parametro) { /* Código no mostrado */ }
    public String Metodo2(String parametro) { /* Código no mostrado */ }
}

class Program
{
    static void Main(string[] args)
    {
        /*1*/ ClaseUno a = new ClaseUno();
        /*2*/ InterfazUno b = a;
        /*3*/ InterfazDos c = a;
        /*4*/ ClaseDos d = b;
        /*5*/ ClaseDos e = new ClaseUno();
        /*6*/ ClaseDos f = new ClaseDos("prueba");
        /*7*/ InterfazUno g = f;
        /*8*/ InterfazDos h = f;
        /*9*/ InterfazUno i = h;
        /*10*/ InterfazDos j = i;
        /*11*/ InterfazUno k = new InterfazUno();
    }
}
```

2. Indiquen las líneas de código incorrectas y justifiquen la decisión. Deben validar línea por línea. Si una asignación es incorrecta, deben asumir que la variable de todas formas fue definida.
- 3.1 ¿Cuáles tipos tiene cada uno de los objetos contenidos o referenciados en las variables definidas en las líneas correctas? Indiquen los nombres de los tipos para cada variable. Consideren solo los tipos definidos en esta pregunta. Si más de una variable hace referencia al mismo objeto, pueden escribir "variables P,Q,R : tipos X,Y,Z".
- 3.2 ¿Que variables contienen o referencian a objetos con tipos en común? Para cada tipo en esta situación indiquen las variables que contienen o referencian. Consideren solo los tipos definidos en esta pregunta y las variables definidas en las líneas correctas.
- 3.3 ¿Qué variables contienen o referencian a un objeto que tiene más de un tipo? Para cada variable en esta situación indiquen los tipos. Consideren solo los tipos definidos en esta pregunta y las variables definidas en las líneas correctas.
- 4 ¿Qué mensajes pueden ser enviados desde el código del método `void Main(string[] args)` a los objetos contenidos o referenciados en las variables `g` y `h` asumiendo que esas líneas fueran correctas? ¿Cómo lo saben? Escriban la firma de los mensajes. Consideren solo los tipos definidos en esta pregunta.
5. ¿Es correcto el siguiente programa? Si lo es, ¿qué imprime por consola al ejecutarse?

```
interface InterfazUno
{
    String P { get; }
    String Metodo(InterfazUno uno);
}
```

```

class ClaseUno: InterfazUno
{
    public String P { get { return "1"; } }
    public String Metodo(InterfazUno uno) { return uno.P + "1"}
}
class ClaseDos: InterfazUno
{
    public String P { get { return "2"; } }
    public String Metodo(InterfazUno uno) { return uno.P + "2"; }
}
class Program
{
    public static void Main(String[] args)
    {
        ClaseUno uno = new ClaseUno();
        ClaseDos dos = new ClaseDos();
        Console.WriteLine(uno.Metodo(dos));
        Console.WriteLine(uno.Metodo(uno));
        Console.WriteLine(dos.Metodo(uno));
    }
}

```

El siguiente código compila, ejecuta y funciona bien. Fue desarrollada para resolver el problema de crear cuadrados y rectángulos y calcular su área.

```

class Figura {
    private Int32 lados;
    public Int32 Lados { get { return lados; } }
    private Int32[] largos;
    public Int32[] Largos { get { return largos; } }
    public Figura(Int32[] largos, Int32 lados) {
        if (largos.Length != lados) {Console.WriteLine("ERROR");}
        this.largos = largos;
        this.lados = lados;
    }
}
class CalculadorArea {
    public Double CalcularArea(Figura f) {
        // Cuadrado, area=lado*lado, todos los lados iguales
        if (f.Lados == 4) {
            return f.Largos[0] * f.Largos[1];
        }
        // Triangulo, area=base*altura/2, 0 es altura, 1 es base
        else if (f.Lados == 3) {
            return (f.Largos[0] * f.Largos[1]) / 2;
        }
        // Solo soporto triangulos y cuadrados, informar con -1
        else { return -1; }
    }
}

```

6. Critiquen en forma breve el código anterior. Elijan aquellos puntos que consideren más graves. Basen parte de su respuesta en el patrón Experto y mencionen como programarían la solución basándose en el Experto.
7. Es posible proveer una solución al problema de calcular el área mejor que la anterior usando polimorfismo (teniendo en cuenta que solo se deben soportar cuadrados y triángulos). Programen una solución que utilice polimorfismo y no utilice para nada el atributo Lado. Pueden escribir el código de cero si lo desean.