

## Prueba 2 de Evaluación Continua

### Análisis y Diseño de Software (2014/2015)

Contesta AL DORSO o en las áreas sombreadas, usa hojas adicionales si fuese necesario

Apellidos:

Nombre:

#### Ejercicio 1: *Diseño y Programación Orientada a Objetos* (3,5 puntos)

En una aplicación orientada a objetos para un negocio de numismática (monedas y sellos), todos los objetos numismáticos tienen un valor nominal (impreso o grabado en el sello o moneda) y un factor de antigüedad; ambos participan en el cálculo del precio de mercado del objeto. En general, este precio es la suma del valor nominal más un valor de antigüedad que varía según se trate de un sello o una moneda. En el caso de una moneda el valor de antigüedad es el producto del factor de antigüedad por la mitad del valor nominal; y en el caso de un sello el valor de antigüedad es el producto del factor de antigüedad por el doble del valor nominal. En el caso de los sellos su factor de antigüedad es siempre 0.125.

Se pide diseñar adecuadamente las clases Java necesarias para modelar los objetos descritos arriba de forma que al ejecutar la siguiente clase de prueba se produzca la salida mostrada abajo. No escribas métodos innecesarios para esta ejecución ni repitas código innecesariamente, pero aplica correctamente los fundamentales de diseño a las clases necesarias así como a las relaciones entre ellas y entre sus métodos.

```
public class Ej1 {
    public static void main(String[] args) {
        Numismatico x1 = new Moneda(8.0, 0.75); // nominal 8, coeficiente antigüedad 0.75
        Numismatico x2 = new Sello(2.0); // nominal 2, sin coeficiente conservación

        System.out.println("    Precio\tNominal\tValorAntigüedad");
        System.out.println("moneda1: "+x1.precio()+"\t"+ x1.nominal()+"\t"+ x1.valorAntigüedad());
        System.out.println("sello 2: "+x2.precio()+"\t"+ x2.nominal()+"\t"+ x2.valorAntigüedad());
    }
}
```

Salida esperada:

Precio	Nominal	ValorAntigüedad
moneda1: 11.0	8.0	3.0
sello 2: 2.5	2.0	0.5

```
public abstract class Numismatico {
    private double valorNominal;
    private double factorAntigüedad;

    public Numismatico(double valorNominal, double coefAntigüedad){
        this.valorNominal = valorNominal;
        this.factorAntigüedad = coefAntigüedad;
    }

    public double nominal() { return this.valorNominal; }

    public double factorAntigüedad(){ return this.factorAntigüedad; }

    public double precio(){ return valorNominal+valorAntigüedad(); } // es igual en monedas y en sellos

    public abstract double valorAntigüedad(); // se calcula distinto en monedas y en sellos
}

public class Moneda extends Numismatico {
    public Moneda(double valorNominal, double coefAntigüedad) { super(valorNominal, coefAntigüedad); }

    public double valorAntigüedad() { return this.nominal() * 0.5 * this.factorAntigüedad(); }
}

public class Sello extends Numismatico {
    private static final double COEF_ANTIGUEDAD_SELLOS = 0.125;

    public Sello(double valorNominal) { super(valorNominal, COEF_ANTIGUEDAD_SELLOS); }

    public double valorAntigüedad() { return this.nominal() * 2 * this.factorAntigüedad(); }
}
```