

**Prueba 2 de Evaluación Continua****Análisis y Diseño de Software****Contesta en esta misma hoja****Apellidos:****Nombre:****Apartado 1 (3 puntos)**

Dada la definición de las clases A, B y C, señala los errores de compilación del método *main* en el siguiente fragmento de código Java (si los hubiera). Para cada error explica en una línea por qué se produce. Después de eliminar las líneas que los provocan, indica la salida del programa:

```

class A {
    protected int n = 10;
    public void f(int x) { System.out.println("A:" + x + ":" + n); }
}

class B extends A {
    public B(double m) { n = (int) m; }
    public void f(double x) { System.out.println("B:" + x + ":" + n); }
}

class C extends B {
    protected double n = 20;
    public C(int m) {
        super(m);
        n = 2 * m;
    }
    public void f(int x) { System.out.println("C:" + x + ":" + n); }
    public void f(double x) { System.out.println("D:" + x + ":" + n); }
}

public class Main{
    public static void main(String[] args) {
        A aa = new A();           //Linea 1
        A ab = new B(20);         //Linea 2
        B bb = new B(30);         //Linea 3
        A ac = (B) new C(40);     //Linea 4
        B bc = new C();           //Linea 5

        aa.f(6.0);                //Linea 6
        bb.f(7);                  //Linea 7
        ab.f(8);                  //Linea 8
        ab.f(9.0);                //Linea 9
        ((B)ab).f(10.0);          //Linea 10
        ac.f(11);                 //Linea 11
    }
}

```

Nº Línea, motivo del error o salida del programa

(puedes seguir al dorso)



## Prueba 2 de Evaluación Continua

### Análisis y Diseño de Software

Contesta en esta misma hoja

Apellidos:

Nombre:

#### Apartado 2 (3 puntos)

A continuación se muestra el código de una aplicación para facturación de mensajería que incluye dos clases auxiliares, Carta y Paquete, completamente definidas en el mismo paquete, y que utiliza otras dos clases CodigoPostal (dada también en este enunciado, sin incluir detalles) y Envio, ambas ubicadas en paquetes diferentes. Escribe y comenta el código Java completo de la clase Envio para que se pueda ejecutar la prueba que figura en el método main de la aplicación.

Archivo: aplicacionMensajeria/Aplicacion.java

```
package aplicacionMensajeria;
import ubicaciones.CodigoPostal;
import envios.Envio;

class Carta extends Envio {
    public boolean urgente;
    public Carta(CodigoPostal origen, CodigoPostal destino, boolean urgente) {
        super(origen, destino);
        this.urgente = urgente;
    }
    public double costeDistancia() {
        return (destino.distancia(origen) > Envio.LARGA_DISTANCIA) ? 3.5 : 0.75;
    }
    public double costeContenido() { return (urgente) ? 1.0 : 0.0; }
}

class Paquete extends Envio {
    public int peso; // en gramos
    public Paquete(CodigoPostal origen, CodigoPostal destino, int peso) {
        super(origen, destino);
        this.peso = peso;
    }
    public double costeDistancia() {
        return 20.0 * (1 + (int) (destino.distancia(origen) / Envio.LARGA_DISTANCIA)) ;
    }
    public double costeContenido() {
        return 3.5 * (1 + (peso / 250));
    }
}

public class Aplicacion {
    public static void main(String[] args) {
        CodigoPostal c = new CodigoPostal("E28007");
        Envio e = new Carta(c, c, true);
        System.out.println(e.costeTotal()); // debe imprimir 1.75 (0.75 más 1 por urgente)
        // implementacion sin terminar
    }
}
```

Archivo: ubicaciones/CodigoPostal.java

```
package ubicaciones;

public class CodigoPostal {
    private String id;
    public CodigoPostal(String id) {this.id = id;}
    public String getId() {return id;}
    public double distancia(CodigoPostal x){
        double d = 0.0 ;
        // cálculo complejo de la distancia no se muestra aquí
        return d;
    }
}
```



## Prueba 2 de Evaluación Continua

### Análisis y Diseño de Software

Contesta en esta misma hoja

Apellidos:

Nombre:

#### Apartado 3 (4 puntos)

Se quiere construir una aplicación que controle las reservas de espacios en un complejo hotelero, sin emplear fechas. Los espacios reservables pueden ser salas de reuniones y de videoconferencia, las primeras se pueden reservar en modo exclusivo o modo compartido, mientras que las de videoconferencia sólo en modo exclusivo. Ningún espacio puede estar reservado en modo exclusivo por dos o más clientes a la vez, y un mismo cliente no puede reservar un espacio que ya está reservado por el mismo, aunque sea con un modo diferente.

Escribe las clases necesarias para que la salida resultante sea la que se muestra más abajo.

```
public class Hotel {
    public static void main(String[] args) {
        String cli1 = "Luis";
        String cli2 = "Pedro";

        SalaReuniones sr = new SalaReuniones ("SR-01");
        VideoConferencia vc = new VideoConferencia ("VC-03");

        sr.reservar(cli1, Modo.COMPARTIDO); //cli1 reserva sr en modo compartido
        vc.reservar(cli1, Modo.COMPARTIDO); //cli1 intenta reservar vc en modo compartido

        vc.reservar(cli2); //cli2 reserva vc en modo exclusivo
        sr.reservar(cli2, Modo.COMPARTIDO); //cli2 se añade a la reserva de sr

        //Nota: getReservas() devuelve la lista de clientes(List<String>),
        //y se llama automáticamente a toString() de la lista al concatenar las cadenas,
        //produciendo una cadena de la forma esperada: [elemento1, elemento2, ...]
        System.out.println("Reservas de " + sr + " = "+ sr.getReservas());
        System.out.println("Reservas de " + vc + " = "+ vc.getReservas());

        sr.liberar(cli1); //libera la reserva de cli1
        sr.liberar(cli2); //libera la reserva de cli2
        vc.liberar(cli2); //libera la reserva de cli2
    }
}
```

Salida:

Reservas de SR-01 = [Luis, Pedro]  
Reservas de VC-03 = [Pedro]

#### Notas:

- Aunque en el programa no se ha usado, *reservar* debe devolver **true** o **false** dependiendo de si al cliente se le ha otorgado o no la reserva del espacio solicitado
- Se puede emplear el método "boolean contains(object)" de las colecciones de java como List, etc., para saber si la colección contiene ya el objeto indicado.

(puedes seguir al dorso)

