

## Examen de Programación 3 - 2ª parte - Cuestiones

Dada la siguiente implementación parcial del UML adjunto, (se han omitido los import), responde a las cuestiones en los espacios correspondientes en el propio enunciado. Asume que el método clone() devuelve un objeto que es copia del objeto con el cual se invoca al método.

```
class Catalogo {
    private List<Item> items;
    public Catalogo() {
        items = new ArrayList<Item>();
    }
    // ...
}

class Ticket {
    private List<LineaTicket> lineas;
    public Ticket() {
        lineas = new ArrayList<LineaTicket>();
    }
    // ...
}

class Articulo implements Item {
    // ...
    public Articulo() { descripcion = "XX"; referencia="YY"; precio=0.0; }
    public String toString() {
        return this.descripcion + "("+ this.referencia + ") = " + this.precio;
    }
}

class Producto extends Articulo {
    private int stockActual;
    public Producto() { stockActual=0; }
    // ...
    public String toString() { return "Producto: " + super.toString() + ", stock="+this.stockActual; }
}

class Servicio extends Articulo {
    // ...
    public String toString() { return "Servicio: " + super.toString(); }
}
```



1. (0.25 puntos) Dado el UML de la figura, ¿indica cuál de las tres opciones ocurre al ejecutar una línea de código en Java como la siguiente?

```
Item p = new Producto();  
Item q = p;
```

1. Por este orden, se reserva sitio en la pila para un objeto de clase Producto, se invoca al constructor por defecto de la clase Artículo (que a su vez invocará al de Producto) y se almacena en la posición de la pila correspondiente a la variable 'p' la dirección de memoria del objeto recién creado. A continuación se crea una copia del objeto al que referencia 'p' en el heap y se almacena su dirección de memoria en la posición de la pila correspondiente a la variable 'q'.
2. Por este orden, se reserva sitio en el heap para un objeto de clase Producto, se invoca el constructor sin parámetros de Producto y se almacena en la posición de la pila correspondiente a la variable 'p' la dirección de memoria del objeto recién creado. A continuación se copia en la posición de la pila correspondiente a la variable 'q' la dirección de memoria almacenada en 'p'.
3. Por este orden, se invoca al constructor de la clase Producto, se reserva sitio en la pila para un objeto de la clase Producto y se almacena en la posición del heap correspondiente a la variable 'p' la dirección de memoria del objeto recién creado. A continuación se copia en la posición de la pila correspondiente a la variable 'q' la dirección de memoria almacenada en 'p'.

1. \_\_\_\_\_

2. (0.25 puntos) ¿Qué caracteres hay que poner en el espacio de la última línea para que el siguiente código compile y se ejecute sin errores sea cual sea el valor de 'x'?

```
Item item;  
double x = Math.random();  
if (x>0.5)  
    item = new Producto();  
else  
    item = new Servicio();  
Articulo articulo= _____ item;
```

3. (0.75 puntos) Implementa el constructor de copia de Ticket

4. (0.75 puntos) Implementa el constructor de copia de Catalogo.

5. (0.25 puntos) ¿Cual es el tipo en tiempo de compilación de la variable 'super' en Producto.toString()?

5. \_\_\_\_\_

6. (0.25 puntos) ¿Cual es el tipo en tiempo de compilación de la variable 'this' en Artículo.toString()?

6. \_\_\_\_\_

-O-O-O-

¿Cuál será el resultado de tratar de compilar y ejecutar los siguientes fragmentos de código? Si piensas que se produce algún error de compilación o de ejecución, indica en qué línea se produce el primer error (1 para la primera, 2 para la segunda, etc.) y porqué. Si crees que se compila y ejecuta sin errores, indica el resultado que aparecerá en la salida estándar.

7. (0.5 puntos) Código cliente 1

```
Producto p = new Producto();  
Articulo a = p;  
System.out.println( a.getStockActual() );
```

☐ Error de compilación    ☐ Error de ejecución    ☐ Compila y ejecuta correctamente

Línea del error:

---

Motivo del error o resultado de la ejecución correcta:

---

8. (0.5 puntos) Código cliente 2

```
Item item = new Producto();
Producto p = item;
Servicio s = item;
System.out.println( p.toString() );
System.out.println( s.toString() );
```

☐ Error de compilación    ☐ Error de ejecución    ☐ Compila y ejecuta correctamente

Línea del error:

---

Motivo del error o resultado de la ejecución correcta:

---

9. (0.5 puntos) Código cliente 3

```
Producto p = new Producto();
Servicio s = new Servicio();
Item item = p;
System.out.println( p.toString() );
System.out.println( s.toString() );
System.out.println( item.toString() );
```

☐ Error de compilación    ☐ Error de ejecución    ☐ Compila y ejecuta correctamente

Línea del error:

---

Motivo del error o resultado de la ejecución correcta:

---

10. (0.5 puntos) Código cliente 4

```
Item item = new Servicio();
if (item instanceof Producto) {
    Producto p = (Producto) item;
    System.out.println( p.getStockActual() );
}
```

☐ Error de compilación    ☐ Error de ejecución    ☐ Compila y ejecuta correctamente

Línea del error:

---

Motivo del error o resultado de la ejecución correcta:

---

11. (0.5 puntos) Código cliente 5

```
Articulo p = new Producto();  
p.setStockActual(5);  
p.vender(1);  
System.out.println(p.toString());
```

☐ Error de compilación    ☐ Error de ejecución    ☐ Compila y ejecuta correctamente

Línea del error:

---

Motivo del error o resultado de la ejecución correcta:

---



Figura 1: Diagrama de classes UML.

