

Recuperación parcial 2 - Prácticas - PRG - ETSInf - Curso 2012/13.
17 de junio de 2013. Duración: 50 minutos.

NOMBRE Y GRUPO DE PRÁCTICAS:

1. 4 puntos Al intentar leer un valor entero mediante el método `leerInt` que figura a continuación, puede darse una excepción `NumberFormatException` si el número que se intenta leer no se ha escrito correctamente.

```
public static int leerInt(Scanner t, String mensaje) {
    System.out.print(mensaje);
    String linea = t.nextLine();
    int res = Integer.parseInt(linea);
    return res;
}

public static void main(String[] args) {
    Scanner tec = new Scanner(System.in);
    int valorInt = leerInt(tec, "Introduce un valor: ");
    System.out.println("El valor leído es: " + valorInt);
}
```

Se pide:

1. [1 punto]: Modifica el método `leerInt` para que **propague explícitamente** esa excepción.
2. [3 puntos]: Modifica también el método `main` para que **capture** dicha excepción y vuelva a pedir el valor, ejecutando `leerInt` todas las veces necesarias, hasta que la lectura sea válida.

Solución:

```
public static int leerInt(Scanner t, String mensaje) throws NumberFormatException {
    System.out.print(mensaje);
    String linea = t.nextLine();
    int res = Integer.parseInt(linea);
    return res;
}

public static void main (String[] args) {
    Scanner tec = new Scanner(System.in);
    int valorInt = 0;
    boolean hayError = true;
    do {
        try {
            valorInt = leerInt(tec, "Introduce un valor: ");
            System.out.println("El valor leído es: " + valorInt);
            hayError = false;
        } catch (NumberFormatException e) {
            System.out.println("¡Asegúrate de introducir un int válido!.");
        }
    } while (hayError);
}
```

2. 1 punto El siguiente código se corresponde con las primeras líneas de la clase `Cuenta` vista en prácticas:

```
import java.io.Serializable;
/** La clase Cuenta permite representar una cuenta bancaria. */
public class Cuenta implements Serializable {
    private double saldo;
    private int numCuenta;
```

Se pide: Justifica brevemente cuál es la utilidad de que se diga que la clase implementa `Serializable`.

Solución: El propósito de dicha declaración es poder almacenar y recuperar los objetos de dicha clase en un fichero binario (mediante `ObjectInputStream` y `ObjectOutputStream`).

3. 5 puntos Se desea incluir una nueva operación en la clase `Concordancia` que dada cierta palabra `pal`, devuelva una `String` con los números de líneas en los que dicha palabra aparece en el texto o el mensaje “No se encuentra” en el caso de que `pal` no esté.

Para ello, recuerda que los atributos de la clase `Concordancia` son los siguientes:

```
private NodoCnc prim;
private int talla;
private boolean esOrd;
private String separadores;
```

que los de la clase `NodoCnc` son, a su vez:

```
String pal;
ColaIntEnla numLins;
NodoCnc siguiente;
```

y, finalmente, recuerda que en la clase `ColaIntEnla` está definida la operación `toString()` que devuelve una `String` con los valores encolados en la `ColaIntEnla`.

Se pide: Resuelve el problema planteado añadiendo un método a la clase `Concordancia` con el perfil:

```
public String numsLinea(String pal)
```

Solución:

```
public String numsLinea(String pal) {
    NodoCnc p = prim;
    while (p!=null && !pal.equals(p.pal)) p = p.siguiente;
    if (p!=null) return p.numLins.toString(); else return "No se encuentra";
}
```