

Programación

UT3 - TAREA08

(10 puntos) En esta tarea vas a practicar el uso del manejo de excepciones en Java.

Ejemplo: Antes de comenzar echa un vistazo a este ejemplo porque te puede servir de base:

```
import java.util.InputMismatchException;
import java.util.Scanner;

public class EjemploMultipleCatch {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        try {
            System.out.print("Introduce un número entero: ");
            int numero = scanner.nextInt();

            System.out.print("Introduce un divisor: ");
            int divisor = scanner.nextInt();

            int resultado = numero / divisor;
            System.out.println("El resultado de la división es: " + resultado);
        } catch (ArithmaticException e) {
            System.out.println("Error: No se puede dividir por cero.");
        } catch (InputMismatchException e) {
            System.out.println("Error: Entrada no válida. Se esperaba un número entero.");
        } finally {
            scanner.close();
        }
    }
}
```

Objetivo: Escribir un programa en Java que demuestre el uso de múltiples bloques **catch** para manejar diferentes tipos de excepciones.

Descripción del Programa:

1. **Crear un Array de Enteros:** Define un array de cinco elementos enteros, con los valores del 1 al 5.
2. **Solicitar Datos al Usuario:** El programa debe pedir al usuario que introduzca dos números enteros:
 - El primero representa un divisor.

- El segundo indica una posición en el array (entre 0 y 4).
3. **Realizar una División:** El programa intentará dividir el número en la posición indicada del array por el divisor proporcionado por el usuario.
4. **(9 puntos) Manejo de Excepciones:** Implementar el manejo de excepciones para:
- **(2 puntos) ArithmeticException:** En caso de que se intente una división por cero.
 - **(5 puntos) ArrayIndexOutOfBoundsException:** Si se accede a una posición del array que no existe.
 - **(2 puntos) InputMismatchException:** Si el usuario no introduce un número entero.

Instrucciones:

- Usa un bloque **try** para englobar el código que puede lanzar una excepción.
- Implementa tres bloques **catch** diferentes, cada uno para manejar un tipo específico de excepción mencionado.
- **(1 punto)** Asegúrate de cerrar cualquier recurso abierto usando un bloque **finally** (por ejemplo, cerrar el objeto **Scanner**).
- Muestra mensajes claros y específicos para cada tipo de error, para que el usuario entienda lo que salió mal.

Ejemplo de Salida:

- Si se introduce un divisor válido y una posición válida, el programa mostrará el resultado de la división.
- Si se introduce un 0 como divisor, el programa mostrará un mensaje indicando que no se puede dividir por cero.
- Si se introduce una posición fuera del rango del array (por ejemplo, 5 o -1), el programa indicará que la posición está fuera de límites.
- Si se introduce un valor no numérico para el divisor o la posición, el programa informará que se requiere un número entero.

Una vez terminado el programa, deberás subir su fichero java correspondiente a Aula Virtual.

IMPORTANTE: Recuerda añadir el comentario a nivel de clase donde indiques tu nombre y la fecha en que creaste el archivo.