

1. Ejercicios

Nota

En estos ejercicios es fundamental hacer varias pruebas para comprobar y comprender qué sucede en cada caso (según el tipo de excepción, cuando no hay excepciones, etc.).

A no ser que se indique lo contrario, al lanzar una excepción deberás incluir un mensaje breve sobre el error (`new Exception("...")`), y cuando captures excepciones deberás mostrar la pila de llamadas (`printStackTrace()`).

Ejercicio 1. Implementa un programa que pida al usuario un valor entero utilizando un `nextInt()` (de *Scanner*) y luego muestre por pantalla el mensaje “Valor introducido: ...”. Se deberá tratar la excepción *InputMismatchException* que lanza `nextInt()` cuando no se introduce un entero válido. En tal caso se mostrará el mensaje “Valor introducido incorrecto”.

Ejercicio 2. Implementa un programa que pida dos valores *enteros* (a y b) utilizando un `nextInt()` (de *Scanner*), calcule su división (a/b) y muestre el resultado por pantalla. Se deberán tratar de forma independiente las dos posibles excepciones, *InputMismatchException* y *ArithmeticException*, mostrando en cada caso un mensaje de error diferente en cada caso.

Ejercicio 3. Implementa un programa que cree un array tipo *double* de tamaño 5 y luego, utilizando un bucle, pida cinco valores por teclado y los introduzca en el array. Tendrás que manejar la posible (o posibles) excepciones y seguir pidiendo valores hasta rellenar completamente el array.

Ejercicio 4. Implementa un programa que cree un vector de enteros de tamaño N (número aleatorio entre 1 y 100) con valores aleatorios entre 1 y 10. Luego se le preguntará al usuario qué posición del vector quiere mostrar por pantalla, repitiéndose una y otra vez hasta que se introduzca un valor negativo. Maneja todas las posibles excepciones.

Ejercicio 5. Implementa un programa con tres métodos:

- `void imprimePositivo(int positivo)`: Imprime el valor p. Lanza una *Exception* si $p < 0$
- `void imprimeNegativo(int negativo)`: Imprime el valor n. Lanza una *Exception* si $p \geq 0$
- La función *main* para realizar pruebas. Puedes llamar a ambos métodos varias veces con distintos valores, hacer un bucle para pedir valores por teclado y pasarlos a las funciones, etc. Maneja las posibles excepciones.

Ejercicio 6. Crea una excepción de usuario llamada *NegativoPositivoExcepcion* y modifica el código del ejercicio 5 para utilizar tu excepción personalizada en vez de la excepción genérica.

2. Bibliografía

Adaptación de los ejercicios prácticos del CEEDCV.