

UD4 – Actividad 02 – Colecciones y Estructuras de Datos

IMPORTANTE

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 **A** 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 int **A** y **B** utilizando un **nextInt()** (de Scanner), calcule **A/B** y muestre el resultado por pantalla. Se deberán tratar de forma independiente las dos posibles excepciones, **InputMismatchException** y **ArithmaticException**, mostrando en cada caso un mensaje de error diferente en cada caso.

Ejercicio 3

Implementa un programa que cree un vector tipo *double* de tamaño 5 y luego, utilizando un bucle, pida cinco valores por teclado y los introduzca en el vector. Tendrás que manejar la/las posibles excepciones y **seguir pidiendo valores hasta llenar completamente el vector**.

Ejercicio 4

Implementa un programa con tres funciones:

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

Ejercicio 5

Implementa una **clase Gato** con los atributos nombre y edad, un constructor con parámetros, los getters y setters, además de un método imprimir() para mostrar los datos de un gato. El nombre de un gato debe tener al menos 3 caracteres y la edad no puede ser negativa. Por ello, tanto en el constructor como en los setters, deberás comprobar que los valores sean válidos y lanzar una ‘Exception’ si no lo son. Luego, haz una **clase principal con main** para hacer pruebas: instancia varios objetos Gato y utiliza sus setters, probando distintos valores (algunos válidos y otros incorrectos). Maneja las excepciones.

Ejercicio 6

Crea **una copia del programa anterior** y modifica el main para hacer lo siguiente:

- Crea un ArrayList<Gato>. Luego, utilizando un bucle, pide al usuario que introduzca los datos de 5 gatos: utiliza un Scanner para pedir los datos, instancia el objeto y guárdalo en el ArrayList. Por último, imprime la información de los gatos.
- Maneja las posibles excepciones de modo que en el ArrayList solo almacenemos objetos Gato válidos y el bucle se repita hasta crear y almacenar correctamente 5 gatos.

Licencia



[CC BY-NC-SA 3.0 ES Reconocimiento - No Comercial - Compartir Igual \(by-nc-sa\)](#)

No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

NOTA: Esta es una obra derivada de la original realizada por Carlos Cacho y Raquel Torres.