UD7 Excepciones – Ejercicios A

En los ejercicios en los que haya que programar es fundamental hacer varias **pruebas de ejecución**, probando a que se produzcan excepciones para **comprobar** y **comprender** qué sucede en cada caso (según el tipo de excepción, cuándo no hay excepciones, etc.).

Ejercicio 1

¿Cuál es el nombre en Java de la clase que define las excepciones, y de la que debe heredar cualquier clase que queramos usar para representar una excepción?

Ejercicio 2

¿Cuál es el nombre en Java de la clase que representa las excepciones que se producen al invocar un método de un objeto cuyo valor es “null”?

Ejercicio 3

¿Cuál es el nombre en Java de la clase que representa las excepciones que se producen al obtener un comportamiento anómalo en la entrada y/o salida de información?

Ejercicio 4

Observa el siguiente fragmento de código:

```
String [] array_string = new String [25];
System.out.println (array_string [3].length());
```

¿Qué excepción se produciría en el mismo?

Ejercicio 5

Observa el siguiente fragmento de código:

```
String aux = “hola”;
int aux2 = Integer.parseInt (aux);
```

¿Qué sucedería al ejecutar el mismo?

Ejercicio 6

¿Cuál es la peculiaridad de las excepciones del tipo “RuntimeException” (o de las subclases de la misma)?

Ejercicio 7

¿Qué información de utilidad aporta el método “printStackTrace()” de una excepción?

Ejercicio 8

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 9

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 **ArithmeticException**, mostrando en cada caso un mensaje de error diferente en cada caso.

Ejercicio 10

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 rellenar el vector**.

Ejercicio 11

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.