Ejercicios Tema 9. Gestión de excepciones

*Todos los ejercicios de este boletín deben estar dentro de un **package** llamado **tema09** (siguiendo las reglas que hemos visto en clase, ex: como.germangascon.tema09).*

1. Escribe un método que se encargue de solicitar números al usuario hasta que se indique un número negativo. Se tiene que validar que el usuario no inserte letras.

2. Escribe un método que solicite 10 números con decimales. El programa debe visualizar el número más alto de todos los insertados. Si se insertan letras, se pedirá de nuevo el número. Contabiliza cuántas veces se produce esta excepción y muéstralo al final del método.

3. Escribe un programa que lea por teclado una secuencia de números hasta que el usuario inserte letras. Al final del programa, debe visualizarse el número más alto de todos los que se hayan introducido. Puede ocurrir que un usuario no llegue a insertar ningún número. En este caso, se indicará con el mensaje correspondiente. Debe emplearse un método para solicitar los números.

4. Para comprobar el funcionamiento de algunas excepciones, haz lo siguiente:
   a) Crea un método que reciba un array de enteros. Este método irá leyendo números desde teclado y guardándolos en el array. ¿Qué situaciones se pueden producir y que hacer en ellas?
      • Que se inserte una letra: informar al usuario.
      • Que se sobrepase la capacidad de la array: informar al usuario, finalizar y mostrar el contenido del array.
      • Que el array no esté creado: informar al usuario y finalizar.
   b) Desde el main() crea un array de 5 números enteros y llama al método. Haz diferentes pruebas para forzar las dos primeras excepciones.
   c) Después declara (sin crearlo) un array en el main() y llama al método.

5. Haz un programa que cumpla los siguientes requisitos:
   a) Queremos crear 2 alumnos. De cada alumno guardaremos el nombre, la edad y la altura. Al solicitar los datos al usuario comprueba que al introducir la edad y la altura no se inserten letras. Si es así, repite el proceso hasta que se inserten números.
   b) Crea los 2 alumnos desde el main(). Visualiza la información de cada uno de ellos y después indica qué alumno es mayor.

6. Diseña un método llamado dividirEntreArray que reciba como parámetros un número entero y un array de enteros. El método mostrará por pantalla el resultado de la división entre el número recibido y cada uno de los elementos del array.
   A continuación, escribe un programa que invoque al método con el número 2 y un array con los elementos -2, -1, 0, 1 y 2.

a) Reescribe el método para capturar la excepción derivada del intento de división por 0, de forma que no se detenga la ejecución del programa y se continúe con la división del resto de elementos del array.
b) Reescribe el método anterior con las sentencias de código necesarias para que la excepción no se llegue a producir (comprobar que el divisor tiene que ser distinto de 0).

7. Escribe un método, de nombre mostrarCadenesArray, que reciba como parámetro un array de cadenas de caracteres. El método mostrará por pantalla el primer carácter de cada una de las cadenas contenidas en la array.
   a) Se debe evitar que se produzca una excepción *NullPointerException* si alguna de las posiciones del array contiene una referencia no inicializada (null).

8. Crea un programa que provoque las siguientes excepciones, las trate y muestre un mensaje indicando que la excepción ha sido tratada:
   a) ArithmeticException
   b) NullPointerException
   c) IndexOutOfBoundsException

9. Modifica el programa anterior y crea 3 clases de excepciones personalizadas, haciendo uso de la palabra reservada extends, para los 3 tipos de excepciones anteriores. Cuando se produzca una excepción de los 3 tipos anteriores deberemos lanzar nuestra propia excepción y mostrar un mensaje personalizado.

10. Modifica el código de la calculadora que implementamos en el tema 4 para que tenga un control de excepciones completo. Después simula entrada de datos aleatorios tanto de las opciones del menú como de los números con los cuales operar. Los datos aleatorios podrán ser números o letras. Para poder ver el funcionamiento de la simulación haz uso de Thread.sleep(número_de_milisegundos).

11. Modifica el código del programa para gestionar alumnos que hicimos en el tema anterior para que tenga un control de excepciones completo. Después simula entrada de datos aleatorios tanto de las opciones del menú como de los datos que se deben introducir. Los datos aleatorios podrán ser números o letras. Para poder ver el funcionamiento de la simulación haz uso de Thread.sleep(número_de_milisegons).