

# SOLUCIÓN EJERCICIOS DE EXCEPCIONES

## Problema 1

Haz un programa que lea un texto por teclado, y lo pase a número (usa para ello el método `Integer.parseInt()`). Prueba con los siguientes datos:

- 15
- Dieciséis

¿Qué sucede en cada caso, y por qué?

```
public static void main (String[] args) {  
    Scanner sc = new Scanner(System.in);  
    System.out.print("Introduce texto: ");  
    int numero = Integer.parseInt(sc.next());  
    System.out.println("El numero es " + numero);  
}
```

El “15” string lo pasa a 15 entero. “Dieciséis” daría error.

```
run:  
Introduce texto: 15  
El numero es 15  
BUILD SUCCESSFUL (total time: 3 seconds)
```

```
run:  
Introduce texto: Dieciséis  
Exception in thread "main" java.lang.NumberFormatException: For input string: "Dieciséis"  
    at java.lang.NumberFormatException.forInputString(NumberFormatException.java:65)  
    at java.lang.Integer.parseInt(Integer.java:492)  
    at java.lang.Integer.parseInt(Integer.java:527)  
    at javaapplication30.JavaApplication30.main(JavaApplication30.java:17)  
Java Result: 1  
BUILD SUCCESSFUL (total time: 6 seconds)
```

## Problema 2

Controla la excepción del ejercicio anterior, y da un aviso al usuario de “número incorrecto” si sucede un error.

```
public static void main (String[] args) {  
    Scanner sc = new Scanner(System.in);  
    System.out.print("Introduce texto: ");  
    int numero;  
    try {  
        numero = Integer.parseInt(sc.next());  
    }  
    catch (NumberFormatException nf) {  
        System.out.println("Numero incorrecto " + nf.getMessage());  
    }  
}
```

```
run:  
Introduce texto: dieciséis  
Numero incorrecto For input string: "dieciséis"  
BUILD SUCCESSFUL (total time: 7 seconds)
```

### Problema 3

Haz un programa que divida dos números, y controla que el divisor sea distinto de 0. Si no es así, lanza una `ArithmeticException` (es genérica de Java, ya existe)

```
public static void main (String[] args) {
    Scanner sc = new Scanner(System.in);
    int n1, n2;
    System.out.print("Introduce numerador: ");
    n1 = sc.nextInt();
    System.out.print("Introduce denominador: ");
    n2 = sc.nextInt();
    try{
        int division = n1 / n2;
        System.out.println("El resultado es: " + division);
    }catch(ArithmeticException e){
        System.out.println("Error. El divisor ha de ser distinto de 0");
    }
}
```

```
run:
Introduce numerador: 5
Introduce denominador: 0
Error. El divisor ha de ser distinto de 0
BUILD SUCCESSFUL (total time: 3 seconds)
```

### Problema 4

Haz una función que calcule el factorial de un número. Por ejemplo, el factorial de 5 es:  $5 \times 4 \times 3 \times 2 \times 1$ . El de 3 es  $3 \times 2 \times 1$ .

Si el usuario intenta un factorial de un número  $<$  que 0, o que  $>20$ , lanza una `LimiteException`. Nota: esta excepción no existe en Java, debes crearla heredando de la clase `Exception`.

```
public static void main (String[] args) {
    Scanner sc = new Scanner(System.in);
    int num;
    System.out.print("Introduce numero: ");
    num = sc.nextInt();
    try{
        int facto = factorial(num);
        System.out.println("Factorial de " + num + ": " + facto);
    }
    catch(LimiteException e){
        System.out.println("En numero entrado debe de estar entre 0 y 20");
    }
}
```

```
public static class LimiteException extends Exception {
}
```

```

public static int factorial(int num) throws LimiteException {
    int factorial = 1;
    if (num>20 || num<0) {
        throw new LimiteException();
    }
    for (int x = num; x > 0; x--) {
        factorial = factorial * x;
    }
    return factorial;
}

```

## Problema 5

Hay 2 maneras de tratar con las excepciones:

- Capturarlas con try-catch
- No capturarlas (declarando "throws xxException" en la función).

¿En qué casos nos interesará capturarlas, y en cuales no?

Con throw se requiere el código llamado para manejar la excepción. Con try-catch está muy explícito y lo maneja el solo y no requiere un código externo.

Cual interesa dependerá de si queremos usar nuestro propio código, nuestras propias excepciones o los ya definidos en java.

## Problema 6

Crea un paquete llamado matematicas, en el que escribirás una clase llamada Calculo con dos métodos. El primero de los métodos se llamará factorial y calculará el factorial de un número. El método factorial deberá lanzar una excepción, a la que llamarás LimiteException, en el caso de que el número a calcular sea menor que 0 ó mayor que 20. El segundo de los métodos devolverá true si el número es primo y false en caso contrario, llámalo isPrimo. Ambos métodos deben utilizar el mismo atributo long para realizar sus operaciones.

```

package matematicas;
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner reader = new Scanner(System.in);
        Calculo pepe = new Calculo();
        for(int i=1;i<=20;i++){
            if (pepe.isPrimo(i))
                System.out.println(i + " es primo");
            else
                System.out.println(i + " no es primo");
            try{
                System.out.println("Factorial de " + i + " es " + pepe.factorial(i));
            }catch(LimiteExcepcion e){
                System.out.println(e.getMessage());
            }
        }
    }
}

```

```

public static class Calculo {
    public long factorial (long num) throws LimiteExcepcion{
        long resultado = 1;
        if ((num < 0) || (num > 20)){
            throw new LimiteExcepcion();
        }
        for (int i=1;i<=num;i++){
            resultado = resultado*i;
        }
        return resultado;
    }
    public boolean isPrimo(long num){
        for (int i = 2; i<num; i++){
            if (num%i == 0) return false;
        }
        return true;
    }
}

public static class LimiteExcepcion extends Exception {
    // Creamos una Excepción nueva específica para la clase Calculo.
    public String getMessage() {
        return("Valor fuera de rango (0 - 20)");
    }
}
}

```

## Problema 7

Haz un programa que tras pedir un número entero, calcule su factorial. Utiliza el método factorial de la clase Calculo, hecho en el ejercicio anterior. Maneja la excepción LimiteExcepcion que lanza el método factorial, con la estructura try... catch...finally. Haz que se puedan ir introduciendo sucesivos cálculos hasta que se introduzca la cifra 0 (cero).

```

package matematicas;
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner reader = new Scanner(System.in);
        System.out.print("Introduce un numero (0 para acabar): ");
        int i=reader.nextInt();
        Calculo pepe = new Calculo();
        while(i!=0){
            try{
                System.out.println("Factorial de " + i + " es " + pepe.factorial(i));
            }catch(LimiteExcepcion e){
                System.out.println(e.getMessage());
            }
            System.out.print("Introduce otro numero (0 para acabar): ");
            i=reader.nextInt();
        }
    }
}

```

## Problema 8

Crea un generador de números primos. El programa empezará a generar números entre dos números que pedirá desde el teclado. Tanto el primero como el último número, deberán estar entre 1 y 100.000.000. Si el primer número es mayor que el último, el programa volverá a pedir los números al usuario. Utiliza el método isPrimo de la clase Calculo, hecho en el ejercicio 2.

```
package matematicas;
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner reader = new Scanner(System.in);
        long numero1, numero2;
        Calculo micalculo = new Calculo();
        do{
            System.out.print("Introduzca el primer número: ");
            numero1=reader.nextLong();
            System.out.print("Introduzca el segundo número: ");
            numero2=reader.nextLong();
        }while(numero1>numero2 || numero1<1 || numero2>100000000);
        for (long i=numero1; i<=numero2; i++){
            if (micalculo.isPrimo(i)){
                System.out.print(i + " ");
            }
        }
    }
    public static class Calculo {
        public long factorial (long num) throws LimiteExcepcion{
            long resultado = 1;
            if ((num < 0) || (num > 20)){
                throw new LimiteExcepcion();
            }
            for (int i=1;i<=num;i++){
                resultado = resultado*i;
            }
            return resultado;
        }
        public boolean isPrimo(long num){
            for (int i = 2; i<num; i++){
                if (num%i == 0) return false;
            }
            return true;
        }
    }
    public static class LimiteExcepcion extends Exception {
        // Creamos una Excepción nueva específica para la clase Calculo.
        public String getMessage() {
            return("Valor fuera de rango (0 - 20)");
        }
    }
}
```

## Problema 9

Haz un programa que tras pedir dos números, diga si el mayor es múltiplo del menor. Encapsula el proceso de averiguar si un número es múltiplo de otro, en un método que llamarás sonMultiplos, que estará en la clase Calculo del paquete matematicas. El programa tendrá que ir preguntando pares de números y respondiendo si son múltiplos o no hasta que se introduzca el número cero en uno de los números.

```
package matematicas;
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner reader = new Scanner(System.in);
        long numero1, numero2;
        Calculo micalculo = new Calculo();
        System.out.print("Introduzca el primer número: ");
        numero1 = reader.nextLong();
        System.out.print("Introduzca el segundo número: ");
        numero2 = reader.nextLong();
        while (numero1!=0 && numero2!=0) {
            if (numero1 > numero2){
                if (micalculo.sonMultiplos(numero1, numero2))
                    System.out.println(numero1 + " es multiplo de " + numero2);
                else System.out.println(numero1 + " no es multiplo de " + numero2);
            }else{
                if (micalculo.sonMultiplos(numero2, numero1))
                    System.out.println(numero2 + " es multiplo de " + numero1);
                else System.out.println(numero2 + " no es multiplo de " + numero1);
            }
            System.out.print("Introduzca el primer número: ");
            numero1 = reader.nextLong();
            System.out.print("Introduzca el segundo número: ");
            numero2 = reader.nextLong();
        }
    }
    public static class Calculo {
        public boolean sonMultiplos(long mayor, long menor){
            if (mayor%menor==0) return true;
            else return false;
        }
        public long factorial (long num) throws LimiteExcepcion{
            long resultado = 1;
            if ((num < 0) || (num > 20)){
                throw new LimiteExcepcion();
            }
            for (int i=1;i<=num;i++){
                resultado = resultado*i;
            }
            return resultado;
        }
    }
}
```