Tarea 4: Excepciones

Ordiales Caballero, Iñaky

En el desarrollo de todo software uno de los aspectos más importantes a considerar es el manejo de las condiciones de error. Estas condiciones se diferencian entre las de error con las que no se puede hacer nada y finalizará abruptamente el programa, y las excepciones que pueden ser tratadas o manejadas con el objetivo de evitar una finalización abrupta.

En el lenguaje orientado a objetos Java, todas las excepciones son clases con métodos propios pero sin atributos. Las instancias de estas clases son lanzadas por la máquina virtual cuando encuentra un problema. Las excepciones heredan de la clase Exception que hereda de la clase Throwable que hereda de la clase Object. En esta tarea se revisarán algunas de las excepciones más comunes e importantes de Java.

No Verificadas: → RuntimeException → Exception

ArithmeticException

Se lanza cuando ocurre una condición matemática excepcional (normalmente indeterminación). La más común razón es la división de algún número entre cero.

```
>> javac PruebaArithmetic.java

>> java PruebaArithmetic

 Valores:
 a = 0
 b = 2
 c = -1
 Se calculara c = b/a...

 Se produjo una excepcion.
 No se pudo modificar c.
 Impresion de 2 entre 0 es = -1
```

NullpointerException

Se lanza cuando se intenta usar null en algún lugar donde se requiere un objeto. Por ejemplo accediendo atributos, obteniendo el largo o tamaño, usando métodos. Basicamente cuando se intenta usar un objeto null de referencia.

```
>> javac PruebaNullPointer.java

>> java PruebaNullPointer

Figura cuadrado = null
Se intenta el comando cuadrado.lados = 4
Se atrapo la excepcion NullPointerException
El objeto tiene como referencia null...


>>
```

Class Cast Exception

Se lanza en tiempo de ejecución cuando el programa intenta realizar un cast de objetos entre dos clases que no heredan una de otra. Un ejemplo sería intentar un casteo entre Integer y String, por eso hay métodos que lo hacen.

```
>> javac PruebaClassCast.java

>> java PruebaClassCast

Object x = new Integer(4);
System.out.println((String)x);

Exception in thread "main" java.lang.ClassCastException: java.lang.Integer cannot be cast to java.lang.String
        at PruebaClassCast.main(PruebaClassCast.java:6)
```

ArrayIndexOutOfBoundsException → IndexOutOfBoundsException
Se lanza para indicar que se ha intentado acceder a un arreglo con un índice negativo, igual o mayor al tamaño del arreglo.

```
>> javac PruebaArrayIndexOutOfBounds.java

>> java PruebaArrayIndexOutOfBounds


 Se produjo una excepcion.

 Se intento acceder al indice 6
 De un arreglo de tamanio: 6
```

StringIndexOutOfBoundsException → IndexOutOfBoundsException
Parecida a la anterior ya que una String es una cadena de chars, se lanza cuando se usa un índice negativo o mayor al largo de la string. Algunos métodos también marcan excepción con el índice igual al tamaño porque aunque existe, no es char.

```
>> javac PruebaStringIndexOutOfBounds.java

>> java PruebaStringIndexOutOfBounds

 Se produjo una excepcion.

 Se intento obtener caracter 11
 De un String de tamanio: 11

>>
```

NumberFormatException → IllegalArgumentException

La excepción indica que el programa ha intentado convertir una string en un tipo de dato numérico, pero la String no cumple con el formato de el tipo de dato especifico.

```
>> javac PruebaNumberFormat.java

>> java PruebaNumberFormat

Se intenta poner string: 4.44 a entero.

Exception in thread "main" java.lang.NumberFormatException: For input string: "4.44"
        at java.lang.NumberFormatException.forInputString(Unknown Source)
        at java.lang.Integer.parseInt(Unknown Source)
        at java.lang.Integer.parseInt(Unknown Source)
        at PruebaNumberFormat.main(PruebaNumberFormat.java:5)

>>
```

InputMismatchException → NoSuchElementException

Se lanza cuando una instancia de la clase Scanner recibe un valor o dato que no es acorde al tipo de dato que se le especificó recibiría, o que no entra en su rango.

```
>> javac PruebaInputMismatch.java

>> java PruebaInputMismatch
Ingrese un entero: h
Exception in thread "main" java.util.InputMismatchException
        at java.util.Scanner.throwFor(Unknown Source)
        at java.util.Scanner.next(Unknown Source)
        at java.util.Scanner.nextInt(Unknown Source)
        at java.util.Scanner.nextInt(Unknown Source)
        at PruebaInputMismatch.main(PruebaInputMismatch.java:8)

>> java PruebaInputMismatch
Ingrese un entero: 4.4
Exception in thread "main" java.util.InputMismatchException
        at java.util.Scanner.throwFor(Unknown Source)
        at java.util.Scanner.next(Unknown Source)
        at java.util.Scanner.nextInt(Unknown Source)
        at java.util.Scanner.nextInt(Unknown Source)
        at PruebaInputMismatch.main(PruebaInputMismatch.java:8)

>>
```

Verificadas: → Exception

ParseException

Es lanzada cuando se trata de utilizar los métodos parse() con una String que no tiene el formato adecuado. Algún error salió del parse debido al formato, pasa mucho con fechas.

```
>> javac PruebaParse.java

>> java PruebaParse

String pasada: 2011 11 19
Formato requerido: yyyy-MM-dd
java.text.ParseException: Unparseable date: "2011 11 19"
        at java.text.DateFormat.parse(Unknown Source)
        at PruebaParse.main(PruebaParse.java:15)

>> javac PruebaParse.java

>> java PruebaParse

String pasada: 2011-11-19
Formato requerido: yyyy-MM-dd
Sat Nov 19 00:00:00 CST 2011

>>
```

InterruptedException

Es muy específica ya que se lanza de un método de la clase Thread (hilo) e indica que un hilo esperando o dormido fue interrumpido, se usa para checar si un hilo sigue abierto.

FileNotFoundException → IOException

Excepción de los input/output streams (flujos) indica que se intentó abrir un archivo y este no existe para el pathname dado.

ClassNotFoundException → ReflectiveOperationException
Se lanza cuando se intenta cargar una clase (import) y ésta no se encuentra según el nombre dado o cuando directamente se intenta instanciarla y tampoco la encuentra según el nombre.

NoSuchFieldException → ReflectiveOperationException.
Se lanza al intentar acceder a un atributo de un objeto, pero el nombre del atributo no coincide con ningún atributo de la clase.

NoSuchMethodException
Se lanza cuando se intenta utilizar un método a través de algún objeto o clase, pero el nombre del método no coincide con ninguno de la clase o sus ancestros.

* Extra: no verificadas clases generales

IllegalArgumentException
Sus clases hijo lanzan excepciones para indicar que algún método recibió un parámetro ilegal, inapropiado o no esperado.

IllegalStateException
Sus clases hijo lanzan excepciones indicando que algún método fue llamado en un momento inapropiado o no permitido, es decir cuando la JVM no está en un estado para llevar a cabo las operaciones.

Bibliografía:

- Java™ Platform, Standard Edition 15 API Specification, (2020).
https://docs.oracle.com/en/java/javase/15/docs/api/index.html