



TECNOLÓGICO NACIONAL DE MÉXICO CAMPUS MORELIA

DIVISION DE ESTUDIOS PROFESIONALES

INGENIERÍA EN SISTEMAS COMPUTACIONALES

INVESTIGACION TIPOS DE EXCEPCIONES

PRESENTA

Nicolas Guzman Domínguez

PROFESOR:

Sebastian German Gonzalez Contreras

MORELIA, MICHOACÁN

(06/11/2024)

TIPOS DE EXCEPCIONES

Primero entendamos que las excepciones son eventos anómalos que pueden ocurrir durante la ejecución de un programa y que alteran el flujo normal de ejecución. Estos eventos representan situaciones inesperadas o errores que deben ser manejados de manera adecuada para garantizar que el programa continúe ejecutándose de y evitar así interrupciones.

Las excepciones se dividen en tres categorías principales, cada una heredada de la clase base Throwable. Estas categorías son:

- **Checked exceptions:** Son excepciones que se deben declarar en la firma del método o capturar explícitamente en un bloque try-catch. Si una excepción comprobada no se maneja correctamente, el código no se compilará. Estas excepciones se heredan de la clase Exception.
- **Unchecked exceptions:** Son excepciones que no están obligadas a ser manejadas explícitamente. Estas excepciones ocurren durante la ejecución del programa y no se requiere que sean declaradas en la firma del método o capturadas mediante un bloque try-catch. Se heredan de la clase RuntimeException.
- **Errors:** Son problemas graves que generalmente están fuera del control del programador y no deben manejarse explícitamente. Estos errores indican problemas serios que deberían detener la ejecución del programa. Se heredan de la clase Error.
-

El manejo de las excepciones en java se realiza con un TRY -CATCH

Las excepciones más comunes según las categorías antes presentadas son:

- **Checked Exceptions:**
 - ✓ IOException: Indica un error en una operación de entrada o salida (E/S).
 - ✓ FileNotFoundException: Se lanza cuando se intenta acceder a un archivo que no se encuentra en el sistema.
 - ✓ ParseException: Ocurre al intentar analizar una cadena para convertirla en un formato específico.
 - ✓ SQLException: Representa una excepción específica de la base de datos que ocurre durante la interacción con una base de datos.
- **Unchecked Exceptions:**
 - ✓ RuntimeException: Clase base para excepciones no comprobadas.
 - ✓ NullPointerException: Ocurre cuando intentas acceder a un objeto que es null.
 - ✓ ArrayIndexOutOfBoundsException: Se lanza cuando intentas acceder a un índice fuera del rango válido de un arreglo.
 - ✓ ArithmeticException: Indica un error aritmético, como la división por cero

Igualmente existe mas que esas por ejemplo:

- **FileNotFoundException:** Se lanza cuando se intenta acceder a un archivo que no existe en el sistema de archivos.
- **IOException:** Se lanza cuando ocurre un error durante una operación de entrada o salida (E/S), como la lectura o escritura de archivos.
- **ParseException:** Se lanza cuando se intenta analizar (parsear) una cadena y no tiene el formato esperado, como al convertir fechas o números.
- **SQLException:** Se lanza cuando ocurre un error durante la interacción con una base de datos, como una consulta SQL mal formulada.
- **ClassNotFoundException:** Se lanza cuando Java no puede encontrar la clase especificada durante la ejecución del programa.
- **NoSuchMethodException:** Se lanza cuando se intenta acceder a un método que no existe en una clase o interfaz.
- **InterruptedException:** Se lanza cuando un hilo es interrumpido mientras está ejecutando una tarea, como esperar o dormir.
- **EOFException:** Se lanza cuando se intenta leer más allá del final de un archivo.
- **MalformedURLException:** Se lanza cuando una URL tiene un formato inválido.
- **CloneNotSupportedException:** Se lanza cuando se intenta clonar un objeto que no implementa la interfaz Cloneable.
- **SocketException:** Se lanza cuando ocurre un error relacionado con las operaciones de red utilizando sockets.
- **TimeoutException:** Se lanza cuando un proceso no puede completarse dentro del tiempo límite especificado.
- **FileAlreadyExistsException:** Se lanza cuando se intenta crear un archivo en una ubicación donde ya existe un archivo con el mismo nombre.
- **NegativeArraySizeException:** Se lanza cuando se intenta crear un arreglo con un tamaño negativo.
- **IllegalArgumentException:** Se lanza cuando un método recibe un argumento inapropiado o fuera de los valores válidos.
- **IllegalStateException:** Se lanza cuando un método es invocado en un momento inapropiado, dado el estado del objeto.
- **ClassCastException:** Se lanza cuando se intenta hacer un cast de un objeto a un tipo incompatible con su tipo real.
- **UnsupportedOperationException:** Se lanza cuando se intenta ejecutar una operación no soportada por un objeto o clase.
- **ConcurrentModificationException:** Se lanza cuando una colección es modificada mientras se está iterando sobre ella.
- **NoSuchElementException:** Se lanza cuando se intenta acceder a un elemento que no existe en una colección o estructura de datos.