

## Lección: Excepciones.

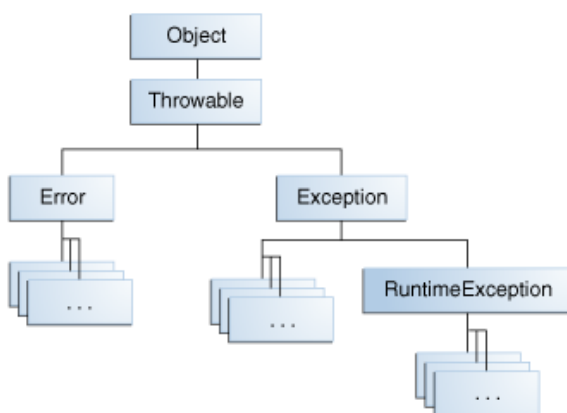
Una excepción es un evento el cual ocurre durante la ejecución de un programa, la cual interrumpe el flujo normal de las instrucciones del programa, cuando esto ocurre el método crea un objeto llamado Objeto de Excepción. El cual es llevado al sistema de ejecución que intenta encontrar algo para manejarlo. Se dice que todo código Java debe estar sujeto a *"The Catch or Specify Requirement"*, esto significa que el código que podría generar una excepción debe ir encerrado por un *try* o en un método que especifique como solucionar la excepción.

Aunque no todas las excepciones están sujetas a *"The Catch or Specify Requirements"*, solo las excepciones comprobadas pertenecen; en cambio las excepciones que tienen que ver con el mal funcionamiento del sistema o del hardware y con errores de programación, son excepciones que no se pueden anticipar o recuperarse.

Hay que tener claro que antes de capturar una excepción, el código debe generarla, y cualquier código puede hacerlo, desde un paquete escrito por otra persona hasta el entorno de ejecución de Java.

Existen numerosas clases de excepciones, las cuales son descendientes de la llamada *"Throwable class"* y todos los programas pueden diferenciar entre los distintos tipos de excepciones, además de que pueden crear sus propias clases de excepciones para dar a conocer un problema que puede ocurrir.

La *"Throwable class"* puede tener objetos que heredan directa o indirectamente de ella. La siguiente figura ilustra la *"Throwable class"* y sus subclases más significativas, además de sus dos descendientes directos.



La clase Error se produce cuando hay un fallo en un enlace dinámico u otro fallo en la JVM.

En la clase Excepción se crean objetos, los cuales la mayoría de programas pueden lanzar y atrapar. Una excepción indica que hubo un problema, pero no es un problema grave en el sistema. La mayoría de programas que escribimos lanzan y capturan Excepciones en lugar de Errores.

El estatuto *try* con recursos declara uno o más recursos. Un recurso es un objeto que debe ser cerrado después de que finaliza el programa con éste. La declaración del *try* asegura que cada recurso se cierre al final de la declaración del mismo. Las excepciones generadas desde la declaración del *try* con recursos son suprimidas, aunque estas se pueden recuperar llamando al método *Throwable.getSuppressed*.

Debido a que Java no requiere métodos para coger o especificar excepciones no comprobadas, los programadores pueden ser tentados a escribir código que lanza excepciones solo y sin restricción o hacer que todas sus subclases de excepción hereden de *RuntimeException*. Ambos permiten escribir código sin preocuparse por errores de compilación pero puede tener inconveniente para los programadores que si utilizan “The Catch or Specify Requirements”.

Las excepciones en tiempo de ejecución representan un problema para estos programadores, ya que son el resultado de un problema de programación, y no sería razonable esperar que el código de API pueda manejar y recuperarse de estas excepciones. Dentro de estas se incluyen problemas de aritmética como división entre cero; problemas de puntero, como tratar de acceder a un objeto a través de una referencia nula; etc. Estas excepciones pueden ocurrir en cualquier parte de un programa. Agregar excepciones de tiempo de ejecución en cada método reduciría la claridad del mismo.

Estas son algunas de las ventajas que presentan las excepciones:

Las excepciones proporcionan medios para separar los detalles de lo que se debe hacer cuando algo fuera de lo normal desde la lógica principal de un programa.

Tienen la capacidad de propagar la presentación de informes de error reportando la pila de llamadas de los métodos.

Todas las excepciones lanzadas dentro de un programa son objetos, por lo cual la agrupación o categorización de las excepciones son un resultado natural de la jerarquía de clases, esta agrupación ayuda a darle un mejor manejo a las excepciones mediante controladores; además de ayudar a la captura de una excepción en función de su grupo o tipo específico.