

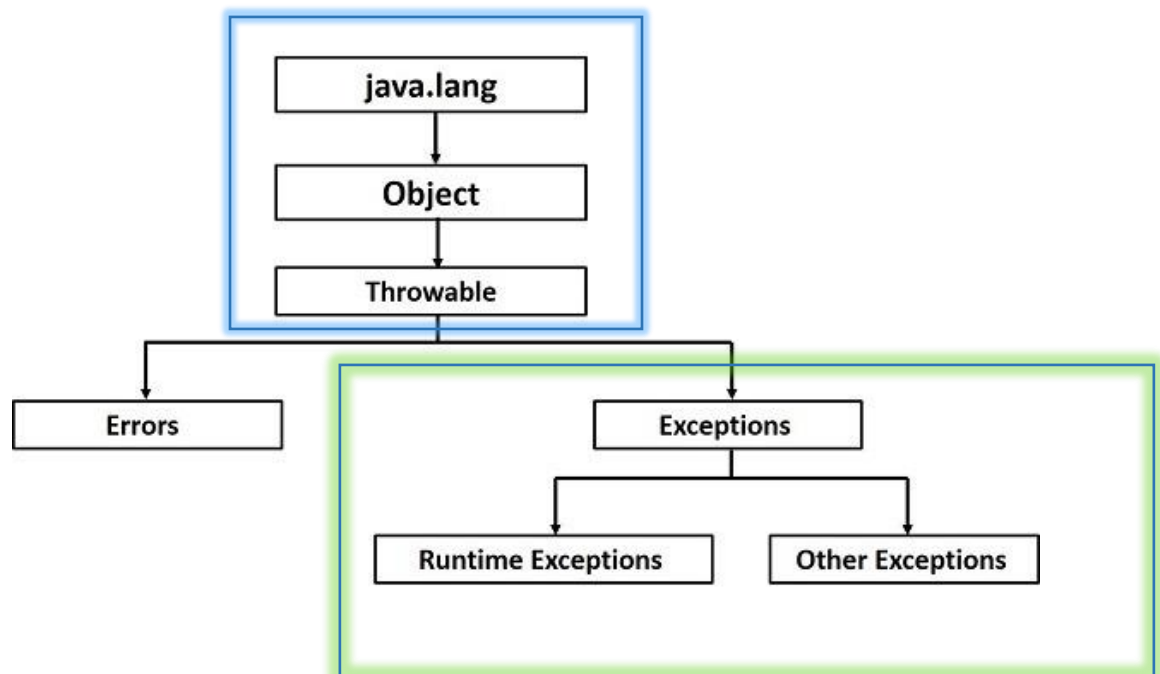
Exceptions:

Las excepciones podrían describirse como eventos o condición inusuales que interrumpen el flujo normal de ejecución de un programa. Representan situaciones excepcionales, como errores o problemas, que pueden ocurrir durante la ejecución de un programa, estas permiten manejar dichas situaciones de manera controlada y predecible, en lugar de permitir que el programa falle abruptamente. La diferencia con los errores es que estos son problemas no recuperables están fuera del control del programador

Tenemos que las excepciones heredan de la clase Throwable (como casi todo, Throwable hereda de la clase Object), y las clases de las excepciones heredan de Throwable.

Dentro de la clase de Excepciones existen dos tipos:

- Runtime exceptions o no verificadas: Son aquellas que el compilador Java no nos va obligar a darles un tratamiento para manejarlas.
- Other Exception o verificadas: Son todas aquellas que si necesitaremos proporcionar un tratamiento para manejarlas.

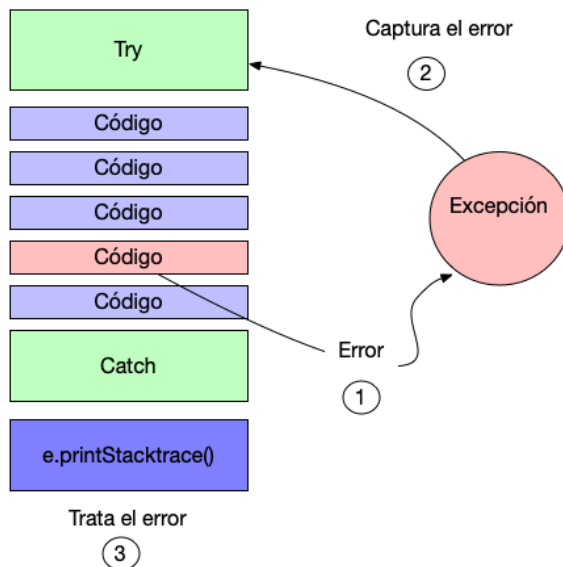


Cuando una excepción ocurre en un programa de Java, el flujo normal de ejecución se interrumpe y se busca una sección de código diseñada para manejar esa excepción. Paso a paso, esto es lo que en el deber ser, debería ocurrir cuando una excepción sucede:

1.-Lanzamiento de Excepción: Cuando ocurre una situación excepcional, como una división por cero o un acceso a una referencia nula, una excepción se lanza. Esto se hace automáticamente por el sistema o explícitamente en el código.

2.- Búsqueda de Manejadores: El sistema comienza a buscar un bloque try-catch adecuado para manejar la excepción. Comienza en el método actual y se propaga hacia arriba en la pila de llamadas (llamada "pila de llamadas" o "pila de ejecución") hasta que se encuentra un bloque catch que puede manejar la excepción específica lanzada.

3.-Ejecución del Bloque Catch: Una vez que se encuentra un bloque catch que coincide con el tipo de excepción lanzada, el código dentro del bloque catch se ejecuta. Este bloque contiene el código para manejar la excepción de manera apropiada, como imprimir un mensaje de error, tomar medidas correctivas, o simplemente registrar la excepción.



3.1.- Try : Dentro del bloque try, se coloca el código que podría generar una excepción. Este bloque es el que monitorea y busca excepciones.

3.2.- Catch: Si ocurre una excepción dentro del bloque try, el control se transfiere al bloque catch correspondiente. El bloque catch debe especificar el tipo de excepción que puede manejar. Si la excepción lanzada coincide con el tipo especificado, el código dentro del bloque catch se ejecutará.

3.3.-Manejo de la Excepción (catch Block): Dentro del bloque catch, se pueden realizar acciones específicas para manejar la excepción. Esto podría incluir imprimir un

mensaje de error, tomar medidas correctivas, registrar la excepción o notificar al usuario.

3.4.-Finally: Opcionalmente, se puede agregar un bloque finally después de un bloque catch. El código en el bloque finally siempre se ejecutará, ya sea que se haya lanzado una excepción o no. Se usa comúnmente para liberar recursos o realizar acciones finales.

4.-Continuación del Flujo: Después de que se maneje la excepción, la ejecución continúa después del bloque catch correspondiente, como si la excepción nunca hubiera ocurrido.

Si no se encuentra ningún bloque catch adecuado en la pila de llamadas, el programa terminará abruptamente y mostrará un mensaje de error en la consola, es ahí donde radica la importancia del manejo de excepciones.