Una excepción es un evento que puede ocurrir durante la ejecución de un programa y que interrumpe la ejecución normal del mismo. Cuando se genera una excepción dentro de un método, el método crea un objeto y lo lanza. Dicho objeto se denomina exception object y contiene información sobre el error, incluyendo el tipo de error y el estado del programa cuando ocurrió el error. Las excepciones se pueden capturar para tratarlas de forma adecuada y dejar a la aplicación en un estado estable.

Existen tres tipos fundamentales de errores:

* Errores: Indican un error irrecuperable. Lo normal es que el programa se interrumpa.
* Excepciones no comprobables: Se generan en tiempo de ejecución. Por ejemplo, invocar a un método usando una variable a null.
* Excepciones comprobables: Constituyen fallos menores, como leer de un fichero que no existe.

Existen tres tipos fundamentales de soluciones:

* Separar el manejo de errores del código normal: el código susceptible de generar excepciones se separa del resto de código. Fácilmente localizable.
* Propagar los errores sobre la pila de llamadas: Si un método lanza una excepción, o bien la captura o bien la relanza sobre la pila de llamadas.
* Agrupar los errores y diferenciarlos: Permite tratar de forma similar errores similares.

Una excepción se puede producir por varios motivos:

* Porque un método reciba datos de entrada que no sabe procesar.
* Porque se produzca una situación excepcional, como por ejemplo invocar a un método con un objeto a null: NullPointerException.
* Porque se produzca un error de entrada/salida. Por ejemplo intentar leer un fichero que no existe: IOException.

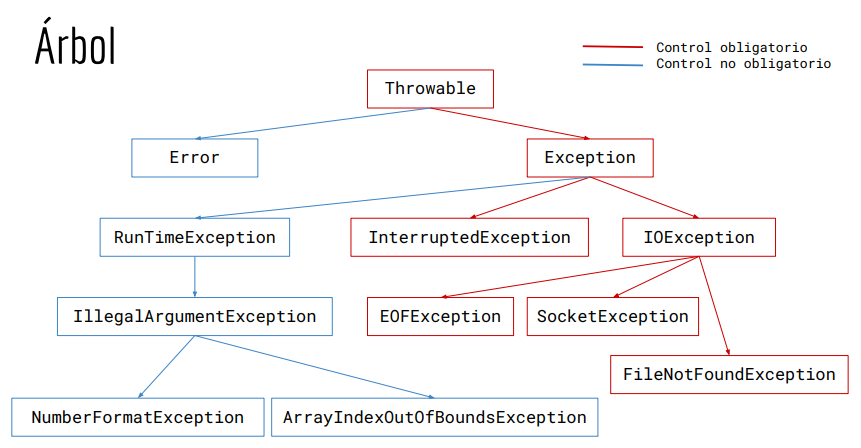
Cuando un método lanza una excepción, puede:

* Tratarla de forma que la aplicación siga estable.
* Lanzarla para que otro método (siguiendo la pila de llamadas) la trate.

Para lanzar una excepción se utiliza: throw new NombreException(...)

Si un método lanza una excepción, su cabecera debe indicarlo. throw new NombreException(...) visibilidad tipo nombreMetodo throws NombreException {...}

Cuando se lanza una excepción con throw, lo que se lanza realmente es un objeto de una clase que representa una excepción. La clase correspondiente a la excepción debe de existir y puede contener distintas constructoras y/o métodos. Un método puede lanzar distintos tipos de excepciones. Cuando la excepción se lanza, esta sube por la pila de llamadas hasta que es capturada o bien interrumpe abruptamente la aplicación.



Cuando un método es susceptible de lanzar excepción, siempre que sea posible conviene darle un tratamiento adecuado. Para ello se usa try / catch. Si un método tiene un bloque try / catch capaz de capturar todas las posibles excepciones que genere su ejecución, entonces el método ya no lanza excepción.

Consideramos la clase Arithmetic. Además, asumimos que las clases ExceptionNegativeNumber y ExceptionDivisionByZero extienden de la clase Error. Debemos considerar todas las excepciones que un código puede lanzar.

Un bloque try / catch puede contener tantos catch como se desee. Los catch, en caso de que exista una jerarquía de excepciones, deben ir ordenados de forma correcta, esto es, desde debajo de la jerarquía hacia arriba. Si no están relacionadas, entonces el orden es irrelevante. Como norma siempre indicaremos en la cabecera de un método (cuando sea posible) el tipo de la excepción concreta que lanza. Intentaremos evitar poner super clases. Da igual que la excepción se capture o no se capture; si se ejecuta un bloque try, entonces su finally correspondiente siempre se ejecuta.