

Síntesis conceptual

Grado: Administración de sistemas informáticos en red
Asignatura: Programación básica
Unidad: 10. Control y manejo de excepciones

Resumen

Un programa tiene que contemplar cualquier situación de error que pueda alterar su estabilidad. Existen las excepciones que son eventos producidos por un error alterando el flujo normal de ejecución. Con las excepciones podemos analizar que parte del código puede producir un error y dar un tratamiento para restablecer la funcionalidad normal o terminar la ejecución del programa.

Existen dos tipos de excepciones según su gravedad **checked** y **unchecked**. Checked son excepciones de carácter leve que requiere hacer un tratamiento obligatorio. En cambio, las excepciones de tipo unchecked, no es recomendable hacer tratamiento debido a que se consideran graves, por ejemplo, que se quede sin memoria, y puede alterar el programa haciéndolo inestable y no pueda hacer su funcionalidad correctamente.

Java ya dispone de unas excepciones predefinidas con una jerarquía definiendo la raíz como *Throwable* y sus posteriores hijos en *Exception* o *Error*, aun así, es posible definir nuestras propias excepciones. Cada excepción se representa con una clase heredera de *Throwable* o excepciones inferiores, puede tener atributos y métodos propios o hacer uso de los métodos heredados, como por ejemplo, `printStackTrace()` que muestra por pantalla el flujo de llamadas en código hasta donde se ha producido el error.

Para capturar y dar tratamiento a una excepción se usará el bloque `try/catch/finally`. Dentro de `try` se pondrá todo el código que puede producir el error, dentro del `catch` el tratamiento que se hará en caso de que se produzca una excepción y `finally` (no es obligatorio), se añadirá código que se ejecutará siempre independientemente se produzca una excepción o no.

Por otro lado, es posible propagar una excepción para que se haga el tratamiento en el método que hizo la llamada inicial. Utilizaremos la palabra reservada ***throws*** en la parte final de la cabecera del método junto con la excepción. Si de lo contrario no se quiere propagar y se quiere lanzar una excepción, añadiremos la línea de código con la palabra reservada ***throw*** junto con una instancia de la excepción.

Conceptos fundamentales

- **Throwable:** Raíz de la jerarquía de excepciones, define el comportamiento de cualquier excepción
- **Checked:** Tipo de excepción que requiere hacer un tratamiento.

- **Unchecked:** Tipo de excepción que no requiere hacer un tratamiento.
- **Throws:** Palabra reservada para propagar en un método una excepción
- **Throw:** Palabra reservada para lanzar una excepción.
- **Pila de llamadas:** Estructura Lifo que guarda todo el flujo de llamadas a métodos del código en el programa. El último elemento siempre será el método *main* del programa, ya que es el primero en llamarse al ejecutarse.
- **Multicatch:** Dependiendo de la versión de java, es posible añadir en el bloque del catch la captura de más de una excepción para dar un mismo tratamiento.