



APRENDERAPROGRAMAR.COM

CONCEPTO DE GESTIÓN DE
EXCEPCIONES EN JAVA.
TRATAMIENTO DE
ERRORES EN
PROGRAMACIÓN
(CU00926C)

Sección: Cursos

Categoría: Lenguaje de programación Java nivel avanzado I

Fecha revisión: 2029

Resumen: Entrega nº26 curso "Lenguaje de programación Java Nivel Avanzado I".

Autor: Manuel Sierra

INTRODUCCIÓN

Hasta ahora hemos omitido una de las características que han contribuido a que el lenguaje Java sea tan importante en el mundo de la programación informática. Se trata de su sistema de gestión de excepciones o comúnmente también llamado sistema de tratamiento de errores.



SISTEMA GESTIÓN DE EXCEPCIONES

Profundizaremos en la gestión de excepciones en próximos apartados del curso. Vamos a dar ahora unas orientaciones generales acerca de la evolución de la gestión de excepciones o errores.

Inicialmente en los lenguajes más antiguos de programación o incluso en el lenguaje ensamblador, no había sistemas de ningún tipo que controlaran los posibles problemas en tiempo de ejecución que pudiera haber en el código excepto aquellos errores más básicos de tipo sintáctico. ¿Qué tipo de problemas ocasionan errores en tiempo de ejecución? Cualquier circunstancia que impidiera que un programa continuara. Un ejemplo típico es una división por cero. Supongamos que tenemos dos variables A y B cuyos valores son $A = 5$ y $B = 0$. Si en una línea de código se incluye una operación B / A el resultado es cero. En cambio, si se incluye una operación A / B se trata de una indeterminación matemática y el ordenador no puede obtener un resultado para esta operación. Ante una situación de este tipo, con los lenguajes antiguos se producía un error y el programa dejaba de funcionar.

Hemos puesto un ejemplo pero la realidad es que existen cientos o miles de casos conocidos que pueden dar lugar a errores. Por ejemplo intentar acceder a ficheros que no existen, entradas de usuario cuyos tipos no coinciden con el esperado, etc.

Por eso en los lenguajes de programación más antiguos había que realizar una muy buena codificación de la funcionalidad deseada para después compilarla en la máquina y esperar que fallara lo menos posible.

Era muy importante la inicialización de las variables, así como el asegurarse de que se podía realizar previamente una tarea antes de realizarla. En este caso por ejemplo era muy usual en lenguajes como C el ver si podíamos leer o escribir un fichero después de haberlo creado.

Con estos primeros lenguajes se empezaron a ver las necesidades de un sistema de tratamiento de errores, y fue cuando muchos de los llamados procedimientos que no devolvían ningún tipo de resultado empezaron a transformarse en funciones que devolvían un entero. La funcionalidad era la misma, pero al devolver un valor (normalmente 0 ó 1) se permitía al programa principal que invocaba el procedimiento conocer el resultado de la operación (correcto o incorrecto). Así habitualmente si devolvía 0 significaba que el proceso había finalizado correctamente, y si devolvía cualquier otro tipo de valor entero entonces dependiendo de su valor era que se había producido un error.

Por ejemplo si un procedimiento transformado en función devolvía un 1 podía informar de que un parámetro necesario estaba null o vacío, si devolvía 2 podía significar que un recurso estaba ocupado o no se tenían permisos, etc.

Esto hacía que el tratamiento de errores fuera muy engorroso, y que además se mezclara código digamos funcional, con código de tratamiento de errores.

Este sistema de devolución de información sigue siendo utilizado en muchos lenguajes, incluido Java. Por ejemplo muchos métodos del api de Java devuelven valores booleanos true ó false para informar del resultado de la operación.

Finalmente los lenguajes fueron evolucionando hacia la incorporación de verdaderos sistemas gestores de excepciones con amplias posibilidades, como es el caso del lenguaje Java que estamos estudiando.

La incorporación de sistemas de tratamiento de errores permitió dar un gran paso en la mejora de la respuesta de los programas ante excepciones que se había comprobado que eran inevitables. Muchas veces se le presta poca atención o se olvida un poco a la hora de programar la gestión de errores, pero es una herramienta que resulta de gran ayuda para mejorar la calidad y rendimiento de los programas. Si un determinado lenguaje incorpora un sistema para el tratamiento de errores merece la pena estudiarlo y utilizarlo. Estos sistemas nos permitirán evitar las interrupciones indeseadas y bloqueos de los programas, facilitando el localizar más rápidamente la fuente de un determinado error, así como el darle un tratamiento adecuado manejándolo, acotándolo, haciendo un seguimiento o traza, o hasta incluso ignorándolo.

Hemos hecho una introducción para destacar la importancia del tratamiento de errores. Continuaremos el curso viendo su aplicación práctica, lo que nos ayudará a entender lo anteriormente expuesto.

Próxima entrega: CU00927C

Acceso al curso completo en aprenderaprogramar.com -- > Cursos, o en la dirección siguiente:
http://aprenderaprogramar.com/index.php?option=com_content&view=category&id=58&Itemid=180