

Centro Universitario de Ciencias Exactas e Ingenierías

CUCEI



Computación Tolerante a Fallas

2023B/D06

Profesor: López Franco Michel Emanuel

Otras Herramientas para manejar errores

Gutiérrez Galán Ruben Alejandro

Código: 214798315

Contenido

Introducción:	3
Gestión de errores:	3
Que es una excepción:	3
Métodos para el manejo de las excepciones o errores:	4
Conclusión:	5
Bibliografía:	5

Introducción:

El manejo de errores, también conocido como manejo de excepciones, es una práctica fundamental en la programación que implica prever y tratar situaciones excepcionales o inesperadas que puedan ocurrir durante la ejecución de un programa. Estas situaciones excepcionales, llamadas "excepciones" o "errores", pueden ser causadas por diversos factores, como entradas incorrectas, problemas de red, falta de recursos, problemas de lógica en el código, entre otros.

Gestión de errores:

Durante la ejecución de un programa se pueden producir errores con diversos niveles de severidad:

- Un índice fuera de rango.
- Una división por cero.
- Un fichero que no existe.
- Una conexión de red que falla. etc.

El código sería poco legible si continuamente se tuviese que comprobar las posibles condiciones de error sentencias tras sentencias.

El control de errores hace que el código sea difícil de leer y modificar. Se pierde el flujo lógico de ejecución.

Que es una excepción:

El mecanismo de las excepciones proporciona una forma clara de gestionar posibles errores sin oscurecer el código.

Las excepciones no nos liberan de hacer la detección, de informar y de manejar los errores, pero nos permiten escribir el flujo principal de nuestro código en un sitio y de tratar los casos excepcionales separadamente.

La excepción es un evento (una señal) que interrumpe el flujo normal de instrucciones durante la ejecución de un programa como consecuencia de un error (condición excepcional).

Cuando ocurre un error en un método, entonces se crea un objeto de excepción o lo lanza (throw) al sistema de ejecución.

Este objeto contiene información sobre el error, incluso el tipo y el estado del programa donde ocurrió.

El sistema de ejecución recorre la pila de llamadas buscando un método que contenga un bloque de código adecuado que maneje la excepción (manejador de excepción).

Si el sistema no encuentra un manejador adecuado, entonces el programa termina (mostrando información detallada sobre ella).

Hay tres puntos de vista para las excepciones:

- **El que lanza (eleva) la excepción:**
Durante la ejecución de un método, se encuentra una situación anómala que no sabe cómo resolver e impide que el método cumpla su tarea, por lo tanto, crea una excepción y la lanza,

para informar al nivel superior de la situación anómala y de que la tarea no pudo completarse.

- **El que propaga la excepción:**

Durante la ejecución de un método, recibe una excepción, no sabe cómo tratarla y recuperar el error, por lo que el método no puede completar su tarea, y por lo tanto, propaga la excepción, para informar al nivel superior de la situación anómala y de que la tarea no pudo completarse.

- **El que maneja (captura) la excepción:**

Durante la ejecución de un método, recibe una excepción, y sí sabe cómo resolver la situación que provoca un determinado tipo de excepción, dispone de un tratamiento para ella y recupera el error para poder completar su tarea.

Métodos para el manejo de las excepciones o errores:

- **Bloques try-catch:** Los bloques try y catch se utilizan para capturar excepciones específicas y manejarlas de manera adecuada. El código dentro del bloque try es monitoreado en busca de excepciones, y si se encuentra una excepción que coincide con el tipo especificado en el bloque catch, se ejecuta el código de manejo de errores.
- **Throw/Raise:** En algunos lenguajes, puedes "lanzar" una excepción explícitamente utilizando la palabra clave throw o raise. Esto te permite señalar que ha ocurrido un error en una parte específica del código y propagar ese error hacia arriba en la pila de llamadas hasta que se maneje adecuadamente.
- **Manejo de errores personalizados:** Puedes definir tus propias clases de excepciones personalizadas para representar situaciones específicas de error. Esto permite un manejo más granular y detallado de los errores en tu programa.
- **Validación de entradas:** Una forma de prevenir errores es validar y verificar las entradas de usuario antes de procesarlas. Esto ayuda a evitar que se produzcan errores debido a entradas incorrectas o maliciosas.
- **Logs y mensajes de error:** Incorpora sistemas de registro (logging) para registrar información sobre lo que sucede en tu programa. Esto puede ser útil para rastrear y solucionar errores después de que ocurren.
- **Pruebas unitarias y pruebas de integración:** Realiza pruebas exhaustivas en tu código para detectar y corregir errores antes de que lleguen a la producción. Las pruebas unitarias y las pruebas de integración son prácticas comunes para identificar problemas en el código.
- **Manejo de flujos alternativos:** Piensa en los diferentes escenarios que podrían llevar a errores y desarrolla planes para manejar esos casos. Esto puede incluir el uso de valores por defecto, mostrando mensajes de error amigables al usuario o tomando medidas para recuperarse de manera segura.
- **Documentación:** Proporciona documentación clara sobre cómo manejar ciertos errores y situaciones inesperadas. Esto es útil para otros desarrolladores que trabajen en el código y para ti mismo en el futuro.

Conclusión:

Los errores en la programación son algo que no siempre podremos evitar, aun cuando se crea una planificación previa y se tiene muy en claro lo que se busca desarrollar es imposible prevenir del todo que un fallo se presente durante el proceso de creación o inclusive durante la fase final del proyecto. Aun peor es cuando estos errores se presentan durante la entrega del producto finalizado o tiempo después de que ya se ha colocado en el mercado. Es por ello que como medidas preventivas a posibles fallos se han desarrollado las excepciones, estas cumplen la función de aislar de cierta manera el error permitiendo que el programa siga en funcionamiento, suele enviarse un mensaje al usuario mostrando que ha ocurrido un problema y se procede a iniciar con las acciones que permitirán corregir el error.

Bibliografía:

- Dpto. Lenguajes y Ciencias de la Computación. E.T.S.I. Informática. Universidad de Málaga. (n.d.). Tratamiento de excepciones. <http://www.lcc.uma.es/>. Retrieved August 26, 2023, from http://www.lcc.uma.es/~vicente/docencia/poo/teoria/poo_3_except.pdf