



## Cuestionario Teórico - Clase 15

\*Obligatorio

### Cuestionario Teórico - Clase 15

#### Excepciones

¿Qué es una excepción? (Desde lo conceptual / teórico) \*

una excepción son situaciones inesperadas que no son parte del flujo normal del programa que ocurren durante el tiempo de ejecución y que detienen el flujo del mismo

¿Qué es el manejo de excepciones (exception handling)? (Desde lo conceptual / teórico) \*

el manejo de excepciones es la forma que tiene el programador para controlar los errores que pueden llegar a ocurrir durante la ejecución para así derivar el flujo del programa y evitar su cierre

¿Qué son las excepciones en tiempo de ejecución y qué tienen en común? (técnicamente) \*

las excepciones son objetos, tienen en común que heredan de exception

¿Qué es el Stack Trace o pila de llamadas? ¿En qué orden se lee? \*

es un registro de las llamadas a métodos que se fueron haciendo antes de que se produzca una excepción, se lee en orden LIFO

¿Cuál es la función de cada bloque dentro de una estructura try-catch? \*

try: contiene el código que queremos controlar y puede lanzar un excepción  
catch: es lo que se ejecuta cuando esa excepción ocurre, atrapa la excepción, pueden haber varios bloques catch

En el caso de una misma estructura try-catch que tiene más de un tipo de bloque catch, ¿Se podría ejecutar más de un bloque catch? ¿Por qué? \*

no, cada catch controla un tipo de excepción específica y una vez controlada no tiene necesidad de ser controlada nuevamente

¿En qué parte del código continúa la ejecución del programa una vez manejada una excepción? \*

directamente luego del catch que atrapo la excepción, puede ser código normal o en un bloque finally si lo implementamos

¿Existe una forma de capturar cualquier excepción sin importar su tipo? ¿Qué habría que considerar en ese caso si se tiene más de un tipo de bloque catch en la misma estructura? \*

si, creando un catch genérico que permita capturar cualquier tipo de excepción pero solo puede existir un solo catch genérico en un bloque try y este debe ir al final de todos los demás catch

¿Qué sucede cuando se lanza una excepción? ¿Qué sucede si no la manejo / controlo? \*

se interrumpe el programa y se busca el primer bloque catch que pueda controlar la excepción, si no está controlada se cierra el programa

¿Cómo lanzo una excepción? \*

con la palabra reservada throw

Dentro de un bloque catch, ¿cuál es la diferencia entre "throw;" y "throw ex;"? (ex es un identificador para la instancia de una excepción capturada) \*

throw ex lanza la misma excepcion que le llevo y reinicia el strack trace, throw lanza la misma excepcion sin reiniciar el stack trace

¿Se lanzan en tiempo de compilación o de ejecución? ¿Por qué? \*

en tiempo de ejecucion ya que son errores que se generan en el runtime cuando el programa esta corriendo, ademas son objetos y estos siempre se crean en tiempo de ejecucion

¿Cómo creo una excepción propia? \*

definiendo una nueva clase que herede de exception, puedo usar el constructor por defecto de la clase base o crear el propio

¿Qué es la propiedad InnerException? Describa a qué clase pertenece, su contenido y qué tipo de dato almacena. \*

es una propiedad que almacena una excepción anterior, pertenece a la clase exception, posee toda la información donde se lanzó una excepción, almacena tipo de dato exception

¿Cómo se carga un valor en la propiedad InnerException?

utilizo el constructor de la clase base y le paso el exception

¿Qué sucede si no se le proporciona un valor a la propiedad InnerException? ¿Por qué? \*

queda en null porque es un tipo por referencia, y su valor por defecto es null

De existir un bloque finally, ¿Bajo qué condiciones se ejecutará el código que contiene? \*

siempre se ejecuta el bloque finally, se produzca o no una excepcion

¿Para qué es útil? \*

para evitar la repetición de instrucciones y para liberar recursos tras el lanzamiento de una excepción

¿Dónde se ubica el bloque finally dentro de una estructura de manejo de excepciones? \*

despues del ultimo bloque catch

Se enviará una copia de tus respuestas por correo electrónico a la dirección que has proporcionado.

[Atrás](#)

[Enviar](#)

Página 2 de 2

Nunca envíes contraseñas a través de Formularios de Google.



Este contenido no ha sido creado ni aprobado por Google. [Notificar uso inadecuado](#) - [Términos del Servicio](#) - [Política de Privacidad](#)

Estás editando tu respuesta. Si compartes esta URL, otros usuarios también podrán editar la respuesta.

RELLENAR UNA  
RESPUESTA NUEVA

Google Formularios