UTN F r a
Avellaneda

Programación II

Clase 09a

Programación Orientada a Objetos

Temario

Excepciones

- Definición
- Causas
- Usos
- Objeto

Exception Handling

- Problemática
- Solución
- Mecanismo

01.

Excepciones

Excepciones

Definición

Excepción en programación es un mecanismo que se utiliza para notificar que durante la ejecución de un programa se ha encontrado un evento o condición que hace que el comportamiento se desvíe del flujo normal o deseado.

Excepciones

Causas

- **Error en la lógica código**

  Dividir por cero, acceder una lista con índice inexistente.

- **Datos no válidos**

  Errores de *casting*, ingreso de datos de usuarios.

- **Falta de recursos**

  Memoria o espacio insuficiente, acceso a archivos que no existen.

Excepciones

Usos

- **Notificación**

Además de informar de la detección de un evento anormal, provee información adicional para su identificación.

- **Integridad**

Al no poder procesar el evento anómalo, provoca la interrupción de la ejecución del proceso.

Excepciones

Objeto Excepcion

- **Representación**

  Un evento es representado con un objeto en cuya estructura transporta información:

  Nombre que indica la naturaleza del evento.
  Método y **línea de código** dónde se detectó.
  Descripción con información adicional más legible.
  Inner Exceptions lista de excepciones acumuladas (stack).

- **En C#**

  Todos los objetos que representan excepciones derivan de la clase `System.Exception`.

Excepciones

Causas

- **Error en la lógica código**

  Dividir por cero, acceder una lista con índice inexistente.

- **Datos no válidos**

  Errores de *casting*, ingreso de datos de usuarios.

- **Falta de recursos**

  Memoria o espacio insuficiente, acceso a archivos que no existen.

- **Custom**

  Definidas por los desarrolladores.

02.
Exception Handling

Exception Handling

Problemática

- **Interrupción**

Una interrupción no “manejada” interrumpirá indefectiblemente la ejecución del programa.

- **Previsibilidad**

Algunas excepciones, si bien no son deseadas, su ocurrencia está prevista dentro de los cálculos en la etapa de diseño.

Exception Handling

Solución

- **Exception Handling**

Mecanismo para poder interceptar la presencia de una excepción y poder realizar acciones en consecuencia.

Lanzar otra excepción que complemente informativamente la original.
Corregir el flujo del programa ejecutando un bloque de código alternativo.
No hacer nada.

Exception Handling

Exception Handling en C#

TRY

En este espacio colocaremos el código del que sabemos que bajo determinadas circunstancias puede llegar a lanzar una excepción.

La excepción puede venir del código que estamos explícitamente invocando o provenir de un flujo más adelantado dentro de la jerarquía de tareas que ejecuta la invocación.

Es importante identificar las excepciones concretas que pueden llegar a tener lugar para su correcto manejo.

Exception Handling

Exception Handling en C#

CATCH

Cuando se detecta una excepción en el bloque **try** el programa buscará un bloque **catch** cuya declaración coincida con el tipo de excepción.

Una vez capturada la excepción puede ser asignada a una variable para su posterior inspección.

Exception Handling en C#

CATCH

Puede haber múltiples bloques **catch**.

Si no se especifica la excepción, por defecto captura `Exception` que es la más genérica de todas. Sólo puede haber un **catch** con este tipo de excepción.

Exception Handling

Exception Handling en C#

FINALLY

Este bloque se ejecuta luego del bloque **try** si es que no hubo excepciones o luego de **catch** si las hubo y estaban declaradas.

Este bloque es opcional y sirve como herramienta para un cierre más limpio del proceso de manejo de excepciones.