 



Conceptos fundamentales de Java 6-2: Manejo de errores Actividades prácticas

# Objetivos de la lección:

* Descripción de distintos tipos de errores que pueden ocurrir y cómo se manejan en Java
* Descripción de excepciones utilizadas en Java
* Determinación de las excepciones que se arrojan para una clase base
* Escribir un código para manejar una excepción arrojada mediante el método de una clase base

# Vocabulario:

Identifique el término correspondiente a cada una de las definiciones a continuación.

|  |  |
| --- | --- |
| Throw | Palabra clave en Java que indica que el bloque siguiente de código maneja una excepción especificada. |
| Unchecked exceptions | Una excepción que se maneja de manera opcional |
| Excepciones controladas | Una excepción que se DEBE manejar. |
| Error | Indica que hay un problema con la interpretación de su programa. |
| Excepción | Evita que el intérprete ejecute el resto del código hasta que éste encuentre una palabra catch. |
| Syntax Error | Error que indica un problema con el formato de codificación. |
| Capturar | Error que se produce mientras se ejecuta el programa, también denominado excepción. |
| Error de lógica | Error que se produce como resultado de una lógica de programación incorrecta. |
| Try/catch block | Bloque de código que maneja excepciones al tratar casos donde se arroja una excepción. |
| Runtime error | Errores que se producen durante el tiempo de ejecución y que pueden ser corregidos o manejados por el código. |

# Inténtelo/resuélvalo:

1. Describa la diferencia entre error de sintaxis, error lógico y excepción.

**R/. Error de sintaxis: este tipo de error ocurre cuando el código no sigue la sintaxis correcta de Java. Por ejemplo, si olvida poner un punto y coma al final de una línea de código.**

**Errores de lógica: este tipo de error se produce cuando el código no funciona como se esperaba. Por ejemplo, si escribe la condición incorrecta en la declaración "if", o si usa la variable incorrecta en la expresión matemática.**

**Excepción: este tipo de error ocurre cuando ocurre una situación inesperada durante la ejecución del programa. Por ejemplo, si intenta dividir un número por cero, se lanzará una excepción**

1. ¿Cuál es el problema del siguiente código? Debe imprimir " Hello World! " cuatro veces a la pantalla.

String str = “Hello World”;

for(int i = 0; i < 4; i++);

{

System.out.println(str); str+= ”!”;

}

**R/. Tiene un Error de lógico** La línea del bucle for tiene un punto y coma al final.

**Código correcto:**

String str = "Hello World";

for(int i = 0; i < 4; i++) {

System.out.println(str);

str += "!";

}

1. Describa una excepción que haya experimentado antes en su programa. Explique de qué manera se podría manejar con un bloque de código try/catch.

**R/.** En un programa una excepción que me ha ocurrido es dividir por cero o que la variable no corresponde con la condición. La manera de manejar este error en el bloque de código try/catch es las dos excepciones son:

Para la división por cero seria:

try {

int resultado = 10 / 0;

} catch (ArithmeticException e) {

System.out.println("No se puede dividir por cero.");

}

Para la variable que no corresponde con la condición seria:

try {

int edad = Integer.parseInt("20años");

} catch (NumberFormatException e) {

System.out.println("La cadena no es un número válido.");

}

1. Escriba un segmento de código que tenga:
   1. Un error de sintaxis.
   2. Un error lógico.
   3. Una excepción.
2. ¿Cuál es la diferencia entre una excepción marcada y una excepción sin marcar?

**R/.** Las excepciones marcadas son excepciones que el compilador conoce y obliga al programador a manejar. Estas excepciones deben declararse en el mismo código. Las excepciones no verificadas son excepciones que el compilador no conoce y que el programador no obliga a manejar. Estas excepciones son errores de tiempo de ejecución.