

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.

|  |  |
| --- | --- |
| Catch | Palabra clave en Java que indica que el bloque siguiente de código maneja una excepción especificada. |
| unchecked exception | Una excepción que se maneja de manera opcional |
| Excepción comprobadas | Una excepción que se DEBE manejar. |
| throw | Indica que hay un problema con la interpretación de su programa. |
| Manejo de Errores | Evita que el intérprete ejecute el resto del código hasta que éste encuentre una palabra catch. |
| Charset | Error que indica un problema con el formato de codificación. |
| Excepción en tiempo de ejecución | Error que se produce mientras se ejecuta el programa, también denominado excepción. |
| Error logico | Error que se produce como resultado de una lógica de programación incorrecta. |
| Manejo de excepciones | Bloque de código que maneja excepciones al tratar casos donde se arroja una excepción. |
| Excepcion | Errores que se producen durante el tiempo de ejecución y que pueden ser corregidos o manejados por el código. |

Copyright © 2020 Oracle y/o sus filiales. Todos los derechos reservados. Oracle y Java son marcas comerciales registradas de Oracle y sus filiales. Todos los demás nombres pueden ser marcas comerciales de sus respectivos propietarios.

**Inténtelo/resuélvalo:**

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

* R// Los errores de sintaxis, errores lógicos y excepciones son tres tipos de problemas que pueden ocurrir en un programa de computadora.
* Error de sintaxis: Este tipo de error ocurre cuando se viola la estructura o la gramática del lenguaje de programación.  
  En otras palabras, el programa contiene una construcción que el compilador no puede interpretar.  
  Los errores de sintaxis se detectan durante la fase de compilación del programa.
* Error lógico: Este tipo de error ocurre cuando la lógica del programa es incorrecta, es decir, cuando el programa no produce los resultados esperados.
* Excepción: Este tipo de problema ocurre cuando ocurre un evento inesperado durante la ejecución del programa, como la división por cero, la lectura de un archivo inexistente o el intento de acceder a un índice de matriz que está fuera de los límites.

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// El problema es un punto y coma después del for, lo que hace que el cuerpo no se ejecute en el ciclo. Eliminar el punto y coma arregla el código para imprimir “Hello World" cuatro veces.

Solución

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// Una excepción común que he experimentado en mis programas es la excepción de división por cero. Esta excepción ocurre cuando intentamos dividir un número por cero, lo cual es una operación matemáticamente incorrecta.

Para manejar esta excepción, podemos utilizar un bloque de código try/catch. En el bloque try, colocamos el código que puede arrojar una excepción de división por cero. Luego, en el bloque catch, colocamos el código que maneja la excepción.

Por ejemplo, supongamos que tenemos el siguiente código en Java

**public** **class** Logica {

**public** **static** **void** main(String[] args) {

**int** a = 5;

**int** b = 0;

**int** c = a / b;

System.***out***.println("El resultado es: " + c);

}

}

utilizando un bloque try-catch

public class Logica {

public static void main(String[] args) {

int a = 5;

int b = 0;

try {

int c = a / b;

System.out.println("El resultado es: " + c);

} catch (ArithmeticException e) {

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

}

}

}

El código usado en esta pregunta se encuentra junto a este archivo en la clase lógica

1. Escriba un segmento de código que tenga:
   1. Un error de sintaxis.
   2. Un error lógico.
   3. Una excepción.

R// En la clase Sintaxis, lo que pasa es que se ha omitido un punto y coma en la segunda línea, ósea lo que resulta en un error de sintaxis. En la clase Lógica, se intenta dividir un número por el cero, lo que obviamente resulta en un error lógico. En la clase Excepción, se intenta acceder a un índice que está fuera de los límites de un arreglo, lo que resulta en una excepción de ArrayIndexOutOfBoundsException. En la clase PrincipalClass, pues he dejado creado instancias de cada clase y se ha llamado al método main() de cada una de ellas.

1. ¿Cuál es la diferencia entre una excepción marcada y una excepción sin marcar?

En Java, una excepción marcada checked exception y una excepción sin marcar unchecked exception son dos tipos diferentes de excepciones.

-Una excepción marcada es una excepción que debe ser manejada explícitamente en el código.

- excepción sin marcar es una excepción que no necesita ser manejada explícitamente en el código.

la principal diferencia entre una excepción marcada y una excepción sin marcar es que las excepciones marcadas deben ser manejadas explícitamente, mientras que las excepciones sin marcar no necesitan ser manejadas de forma explícita. Es importante destacar que se recomienda usar excepciones marcadas solo cuando se espera que el llamador pueda hacer algo para manejar la excepción, mientras que las excepciones sin marcar son adecuadas para situaciones en las que es poco probable que el llamador pueda recuperarse de la excepción.

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados. Oracle y Java son marcas comerciales registradas de Oracle y/o sus filiales. Todos los demás nombres pueden ser marcas comerciales de sus respectivos propietarios

2