

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. |
| Excepciones no comprobadas | Una excepción que se maneja de manera opcional |
| Excepciones comprobadas | Una excepción que se DEBE manejar. |
| RuntimeException | Indica que hay un problema con la interpretación de su programa. |
| try | Evita que el intérprete ejecute el resto del código hasta que éste encuentre una palabra catch. |
| Error de sintaxis | Error que indica un problema con el formato de codificación. |
| excepción | Error que se produce mientras se ejecuta el programa, también denominado excepción. |
| Error lógico | Error que se produce como resultado de una lógica de programación incorrecta. |
| try/catch | Bloque de código que maneja excepciones al tratar casos donde se arroja una excepción. |
| Excepciones | 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// los erros de sintaxis ocurren antes de la ejecución del programa es decir en la compilación ya que estos errores de sintaxis son por ejemplo falta de puntos y coma, mala invocación de variables y métodos. Por otro lado, los errores lógicos ocurren durante la ejecución es decir que el programador utilizo mal la lógica y el resultado del programa no es el deseado. Y los errores de excepción se clasifican en dos excepciones no controladas y excepciones controladas es decir que las excepciones comprobadas no son culpa del programados de Java y las excepciones comprobadas son culpa del programador de Java.

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+= ”!”;

}

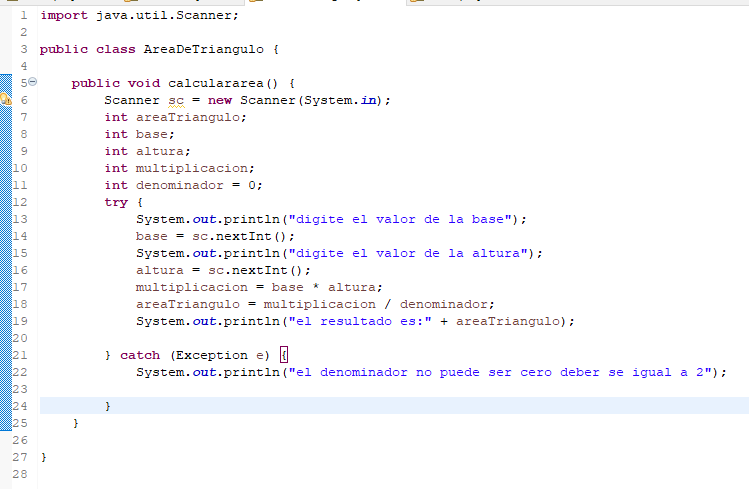
R//

el problema del siguiente código radica en el bucle ya que al final de la declaración del ciclo for se tiene un punto y como, y este punto y coma no permite que el bucle se ejecute de manera adecuada provocando que el “hellow Word!” se imprima una sola vez. Para dar solución al problema debemos quitar el punto y como que se encuentra al lado del for y también para que se imprima la misma palabra cuatro veces debemos quitar el “str=”!” para que este signo de exclamación no se aumente cada vez que imprimamos el “helow Word!”

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

Un programa que le pide la altura y la base de un triángulo para calcular el área de un triangulo con la formula (base\*altura) /2, hice un bloque try/catch para que el programa capture si el denominador es 0 e imprima “el denominador no puede ser cero deber ser igual a 2" y permita continuar con el resto del programa.

.



1. Escriba un segmento de código que tenga:
   1. Un error de sintaxis.

R// error de sintaxis en el for ya que le falta un punto y coma que separa la condición del contador (i++).

Int num=0;

For (int i=0; i<10 i++){

System.out.println(i);

num++;

}

* 1. Un error lógico.

Imprimir números del 1 al 10.

int num=0;

for (int i=0; i<= 10 ; i++) {

System.out.println(i);

num++;

}

* 1. Una excepción.

int num1 = 10;

int num2= 0;

try {

int result = num1 / num2;

System.out.println("Resultado: " + result);

} catch (ArithmeticException e) {

System.out.println("no es posible dividir entre cero");

}

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

La diferencia entre las excepciones marcadas y las excepciones sin marcar es que en las excepciones marcadas no son culpa del programador si no que suceden por factores externos y se deben de tartar con un try/catch para contrarrestar el error, a comparación de las excepciones marcadas son culpa del programador y en la mayoría de los casos pueden ser por errores lógicos.