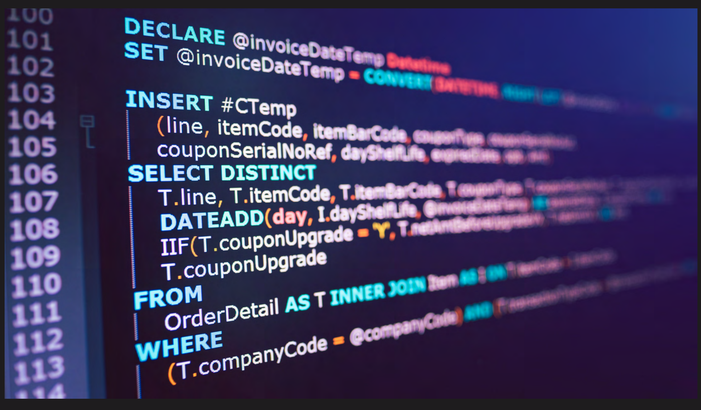
Instrucciones para manejar una excepción

Descripción del manejo de excepciones

Para manejar excepciones, primero debes identificar la declaración en la que puede ocurrir la excepción. Entonces, se vuelve necesario definir las instrucciones que deben ejecutarse para evitar errores. Consúltalas a continuación:

- **Try.** La declaración try se usa para contener la o las instrucciones que pueden generar una excepción.
- **Catch.** La declaración Catch agrupa instrucciones que se ejecutarán en caso de excepción. Puede existir más de un Catch porque cada captura apunta al mismo tipo de excepción. El programa encuentra automáticamente el Catch apropiado (bloque de código) de acuerdo con la excepción.
- **Throws.** Si ocurre una excepción en un método que no lo atrapa (Catch), debes especificar que se replegará, es decir, debe volver al método que lo llamó y buscar un controlador allí. La declaración Throws, al lado del encabezado de la función, se usa para indicar este hecho.
- **Throw.** La instrucción throw se utiliza para lanzar una excepción cuando el programador lo decide.
- **Finally.** Esta declaración resulta útil cuando no necesitamos asegurarnos de que se libere algún recurso o, simplemente, cuando las instrucciones deben ejecutarse independientemente de un error.



Ejemplo:

En el siguiente programa, se ha realizado, de forma intencional, una división entre cero. Observa que la división se lleva a cabo en el método división. Dado que este método no contiene manejadores de excepciones, se le ha agregado la cláusula “throws ArithmeticException” para indicar que se debe regresar al método que lo llamó para buscar un manejador. Como el método main no contiene un manejador, el programa termina con el error:

```
Exception in thread "majajajain" java.lang.ArithmeticException: / by zero
        at Ejemplo.division(Ejemplo.java:9)
        at Ejemplo.main(Ejemplo.java:5)
```

Ejemplo:

```
public class Divide {
    public static void main(String[] args) {
        int a = 10, b = 0;
        System.out.println("resultado = " + division(a, b));
    }
    public static int division (int x, int y) throws ArithmeticException
    {
        return x / y;
    }
}
```