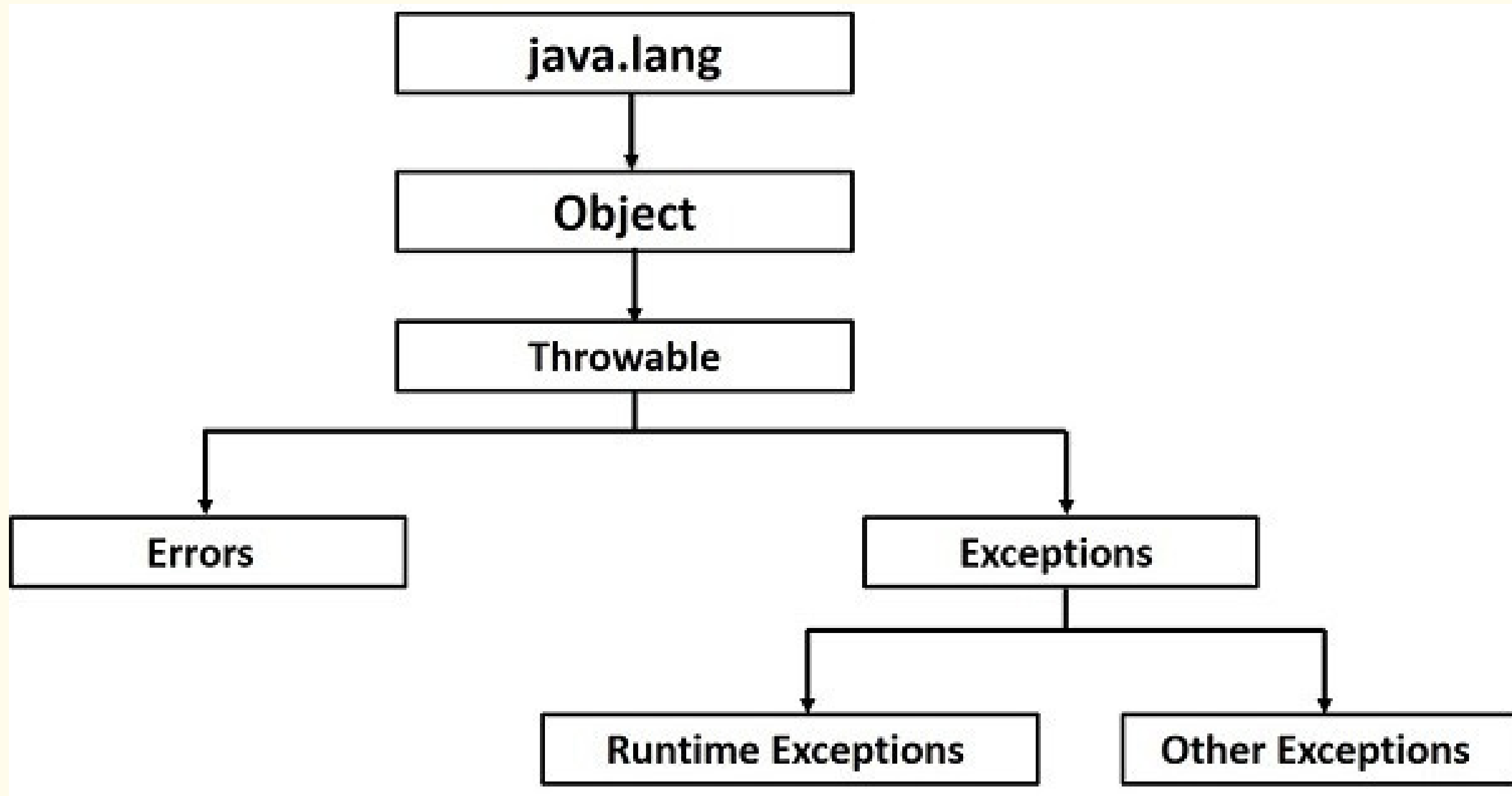




Multicatch & try-with-resources

# TRATAMIENTO DE ERRORES



## Errores (Errors)

Los errores son eventos excepcionales que generalmente están fuera del control del programador y no deberían ser capturados ni manejados por el programa.

Los errores suelen estar relacionados con problemas en el entorno de tiempo de ejecución o problemas en la máquina virtual de Java (JVM)



## Excepciones (Exceptions):

Las excepciones son eventos no deseados o inesperados que ocurren durante la ejecución de un programa, pero que aún pueden ser manejados y recuperados.

### Checked Exceptions


Son aquellas excepciones que el compilador requiere que sean gestionadas o declaradas en el método que las lanza.

### Unchecked Exceptions

Son aquellas excepciones que el compilador no requiere que sean manejadas o declaradas

# Ejemplo Checked Exceptions

ArithmeticException, es una "Checked Exception".



```
1  public class Main {  
2      public static void main(String[] args) {  
3  
4          try {  
5              dividir(10, 0);  
6          } catch (ArithmeticException e) {  
7              System.out.println("Error: No se puede dividir entre cero.");  
8          }  
9      }  
}
```



```
1  public static void dividir(int numerador, int denominador) throws ArithmeticException {  
2      if (denominador == 0) {  
3          throw new ArithmeticException("División por cero.");  
4      }  
5  
6      int resultado = numerador / denominador;  
7      System.out.println("Resultado de la división: " + resultado);  
8  }  
9  }
```

ArrayIndexOutOfBoundsException, es "Unchecked Exception".



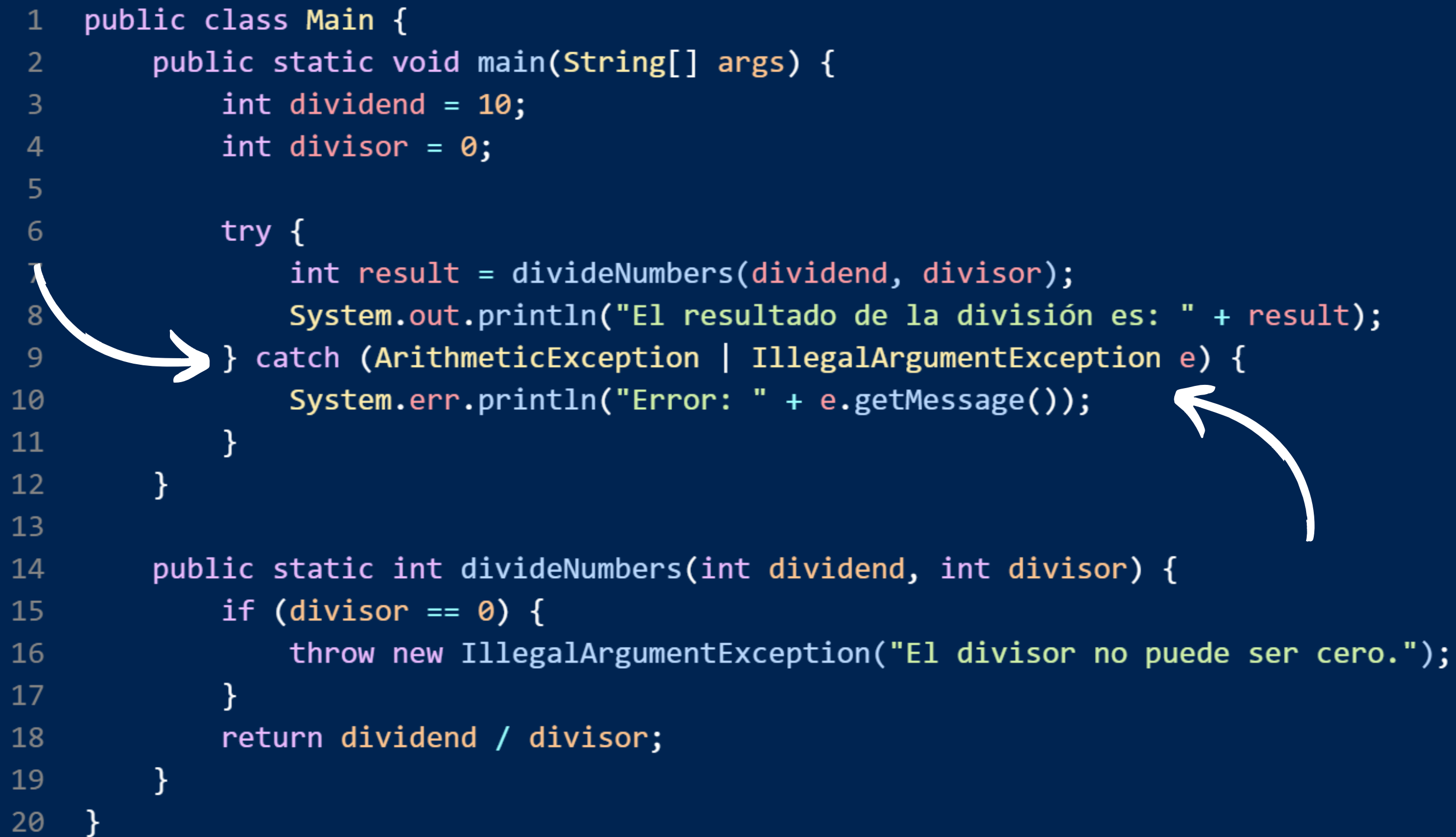
```
1  public class Main {
2      public static void main(String[] args) {
3          int[] numeros = { 1, 2, 3, 4, 5 };
4          int indice = 10; // Índice fuera del rango del arreglo
5
6          try {
7              int numero = obtenerNumero(numeros, indice);
8              System.out.println("Número obtenido: " + numero);
9          } catch (ArrayIndexOutOfBoundsException e) {
10              System.out.println("Error: Índice fuera de rango.");
11          }
12      }
13
14      public static int obtenerNumero(int[] arreglo, int indice) {
15          return arreglo[indice]; // Posible ArrayIndexOutOfBoundsException
16      }
17  }
18
```

## **MULTICATCH**

Es una característica de Java introducida en Java 7 que permite capturar múltiples tipos de excepciones en un solo bloque catch

Lo puedes utilizar manejar diferentes tipos de excepciones que puedan ocurrir durante la operación:





```
1  public class Main {
2      public static void main(String[] args) {
3          int dividend = 10;
4          int divisor = 0;
5
6          try {
7              int result = divideNumbers(dividend, divisor);
8              System.out.println("El resultado de la división es: " + result);
9          } catch (ArithmeticException | IllegalArgumentException e) {
10             System.err.println("Error: " + e.getMessage());
11         }
12     }
13
14     public static int divideNumbers(int dividend, int divisor) {
15         if (divisor == 0) {
16             throw new IllegalArgumentException("El divisor no puede ser cero.");
17         }
18         return dividend / divisor;
19     }
20 }
```