

Ayudantía 7

Alexander Inostroza – alexander.inostroza@usm.cl

Errores de Ejecución

Los errores que ocurren en tiempo de ejecución o runtime son problemas que, al momento de ejecutar las clases ya compiladas, suelen ocurrir por ingreso de datos, manipulación de dispositivos de entrada/salida, condiciones de borde, conversión de tipos de datos, etc.

En Java podemos manejar estos errores o **Exceptions**, a la vez que utilizarlos para escribir un mejor código.

Por ejemplo, para el código

```
public class Main{  
    public static void main(String[] args){  
        int[] arr = new int[10];  
        arr[11] = 1;  
    }  
}
```

El Java runtime nos entrega el siguiente error:

```
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 11  
out of bounds for length 10  
    at Main.main(Main.java:4)
```

Que quiere decir que se produjo una Exception en el programa “main” del tipo `ArrayIndexOutOfBoundsException` (índice del arreglo fuera de sus límites), e incluso muestra una descripción detallada del error. Finaliza mostrando en qué línea de nuestro código se produjo el error.

Podemos detectar estos errores de runtime en nuestro código y, en caso de que ocurra uno, ejecutar código opcional.

```
public class Main{
    public static void main(String[] args){
        try{
            /*
             * código que puede (o no) producir un error
             */
        }
        catch(Exception e){
            System.out.println("Ocurrió un error de ejecución.");
        }
    }
}
```

Podemos, además, crear funciones (métodos) que lancen errores:

```
import java.util.Scanner;

public class Main{
    public static void main(String[] args) {
        try{
            int edad = solicitarEdad();
            edadPositiva(edad);
            System.out.println("Valor ingresado es válido.")
        }
        catch(Exception e){
            System.out.println("Ocurrió un error de ejecución: " + e.getMessage());
        }
    }

    // Solicita edad al usuario, se queja si no recibe un número
    public static int solicitarEdad() throws Exception {
        int edad;
        Scanner scan = new Scanner(System.in);
        System.out.print("Ingrese su edad: ");
        edad = scan.nextInt();
        scan.close();
        return edad;
    }

    public static void edadPositiva(int edad) throws Exception {
        if(edad <= 0) throw new Exception("Edad nula o negativa");
    }
}
```

1. Correcto

Analice el siguiente código y describa su comportamiento.

```
import java.util.Scanner;

public class Main{
    public static void main(String[] args) throws Exception{
        Scanner scan = new Scanner(System.in);
        boolean flag = false;
        String input;
        while(true){
            System.out.print("input: ");
            try{
                input = scan.nextLine();
                if (input.equals("correcto")){
                    flag = true;
                    break;
                }
            }
            catch(Exception e){
                continue;
            }
        }
        if (flag) throw new Exception("Exception");
    }
}
```

2. Triángulos (otra vez)

- Escriba la clase **Triangulo**, la cual nos permita calcular área y perímetro de un triángulo. Puede utilizar la fórmula de Herón para calcular el área.
- Escriba un programa main que le solicite los lados de un triángulo al usuario y le muestre por pantalla el área y el perímetro del triángulo creado con los datos ingresados.

3. Calculadora de divisiones

Escriba el código para una calculadora de divisiones que le pida al usuario dos inputs numéricos, **a** y **b**, y realice la división **a/b**. Debe permitir calcular múltiples divisiones, hasta que el usuario escriba **FIN**.

4. Propuesto: balance de paréntesis

Para crear programas que interpreten un input del usuario, es una tarea común el balance de paréntesis, es decir, revisar que los paréntesis estén bien puestos. Esto se aprecia en programas que reciben un input numérico (como [WolframAlpha](#)) o también en lenguajes de programación con el uso de llaves, como pueden ser C o Java.

Escriba un programa que lea inputs del usuario compuestos de sólo paréntesis y, en caso de encontrar una expresión no balanceada, lance una **Exception** e indique en que símbolo (con un índice) se produce el error.

((()))	balanceado
((()))()	balanceado
)()((no balanceado
()()((()	no balanceado