

ACADEMIA JAVA

P1- IMPLEMENTACION TRY CATCH - XIDERAL: SEMANA 2

NOMBRE: JOSE MANUEL JIMENEZ HDZ

IMPLEMENTACION MULTI-CATCH Y TRY WITH RESOURCES.

MULTICATCH O CATCH DE IMPLEMENTACION MULTIPLE:

Antes de Java 7, teníamos que capturar solo un tipo de excepción en cada bloque catch. Entonces, cada vez que necesitábamos manejar más de una excepción específica, pero tomar alguna acción para todas las excepciones, teníamos que tener más de un bloque catch que contuviera el mismo código.

En el siguiente código, tenemos que manejar dos excepciones diferentes, pero realizar la misma acción para ambas. Así que necesitábamos tener dos bloques catch diferentes a partir de Java 6.0.

Ejemplo:

```
try
{
    int n = Integer.parseInt(scn.nextLine());

    if (99%n == 0)
        System.out.println(n + " is a factor of
99");
}
//a partir de esta línea de aplica el multicatch
catch (ArithmeticException ex)
{
    System.out.println("Arithmetic " + ex);
}
catch (NumberFormatException ex)
{
    System.out.println("Number Format Exception " +
ex);
}
```

TRY WITH RESOURCES O DECLARACION DE PRUEBA CON RECURSOS:

En Java, la declaración de prueba con recursos o try with resources, es una declaración de prueba que declara uno o más recursos. El recurso es como un objeto que debe cerrarse después de finalizar el programa. La instrucción try-with-resources garantiza que cada recurso se cierre al final de la ejecución de la instrucción.

Puede pasar cualquier objeto que implemente java.lang.AutoCloseable, que incluye todos los objetos que implementen java.io.Closeable.

El siguiente ejemplo escribe una cadena en un archivo. Utiliza una instancia de FileOutputStream para escribir datos en el archivo. FileOutputStream es un recurso que debe cerrarse después de que el programa termine con él. Entonces, en este ejemplo, el cierre del recurso se realiza por sí mismo.

Ejemplo:

```
// A partir de esta línea se usa el try with resource
try (FileOutputStream fileOutputStream = new FileOutputStream("/
java7-new-features/src/abc.txt")) {
    String msg = "Welcome to javaTpoint!";
    byte byteArray[] = msg.getBytes(); //converting string into b
yte array
    fileOutputStream.write(byteArray);
    System.out.println("Message written to file successfully!");
} catch (Exception exception) {
    System.out.println(exception);
}
```