Práctica de Excepciones

Luis Ángel Serrano Catalá | Prof. Ricardo Vegas Morales

Programación Orientada a Objetos, Unidad 5.

LIDTS 2°P, Universidad Autónoma de Chiapas.

¿Para qué sirven las excepciones?

Las excepciones son sirven para marcar que algo no sucedió como se esperaba. Para decir cuando un error a ocurrido.

En java hay diferentes tipos de excepciones ya construidas para marcarnos errores específicos pero nosotros podemos extenderlas y crear nuestras propias excepciones para manejar errores personalizados para nuestra librería o programa.

Ejemplos

Digamos queremos hacer una llamada HTTP a una API.

```
var client = HttpClient.newHttpClient();

var request = HttpRequest.newBuilder(URI.create("https://api.examplesite.com/get?
key=123456"))
   .header("accept", "application/json")
   .build();

var response = client.send(request, new JsonBodyHandler<>(APOD.class));
var data = response.body().get();
```

Esta API en particular es una que nosotros no creamos y no tenemos control sobre que datos hay y cuales no.

Entonces deberíamos validar los datos. Para esto podríamos usar un IF para checar si hay o no los elementos que queremos y hacer una cosa u otra pero puede suceder que los datos que queremos usar son esenciales para que nuestro programa funcione, así que si no los tenemos deberíamos mandar un error.

Para esto nos sirven las excepciones.

```
// ...

if(data.title !== null){
    System.out.println(data.title);
} else {
    throw new Exception("No se encontró título");
```

```
}
```

Como puedes ver, este error será más descriptivo a un mensaje de NullException genérico y podríamos crear nuestra propia clase personalizada de excepciones.

Y si esto es parte de una función podríamos envolverla dentro de un bloque Try Catch. Que nos servirá para "atrapar" errores del tipo que especifiquemos, dándonos más control sobre nuestro flujo de código.

```
try {
    LlamadaALaApi();
} catch(Exception e) { // La clase de excepción a atrapar
    System.out.prinln(e.getMessage());
}
```