



Barranqui



Java
user group



Te invita a su próxima Meetup

SÁBADO

19 NOV, 2022

Modo Virtual

Hora: 02:00 a 03:00 PM

GMT-5

Marco Chulz

<https://lbry.tv/@elmar.dott:8>
@ElmarDott

www.JUGBAQ.org



Charla: Swallowed Exceptions

Consultant



Elmar Dott

(Marco Schulz) estudió ciencias de la computación en HS Merseburg, Alemania y tiene un título de ingeniero en ingeniería de software. Tuitea como @ElmarDott regularmente sobre varios temas técnicos. Los principales temas de su campo de trabajo son la gestión de compilaciones y configuraciones, la arquitectura de software y la gestión de versiones.

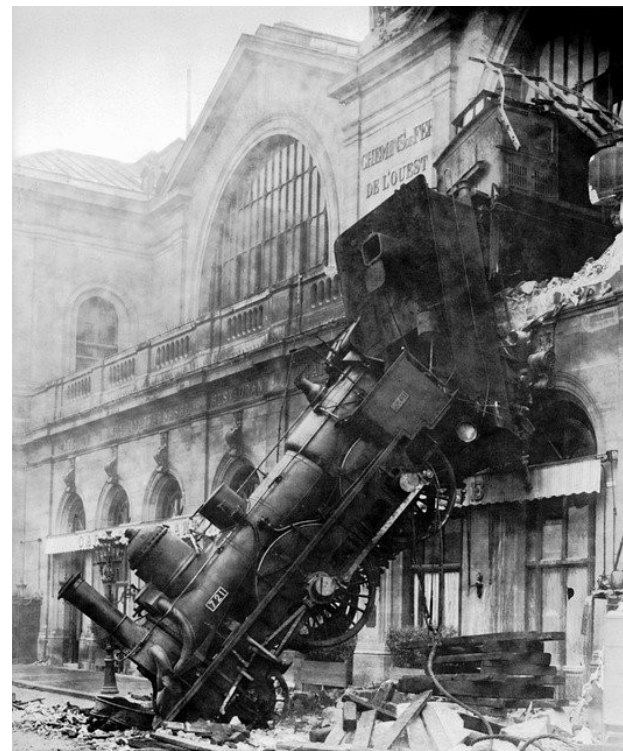
Aproximadamente más de 20 años está trabajando en diferentes grandes proyectos de aplicaciones web en todo el mundo. Es consultor / formador independiente. Para compartir su conocimiento da charlas en conferencias, si no está escribiendo un nuevo artículo sobre ingeniería de software.

<https://elmar-dott.com>

+ Consultor + Escritor + Orador + Entrenador +

Agenda

- 👤 Un Déjà vu
- 👤 La estructura de las excepciones
- 👤 Logging & Testing
- 👤 Mantenibilidad
- 👤 Flujo de Información
- 👤 Microservices



Definicion

Oracle: Una excepción es un evento, que ocurre durante la ejecución de un programa, que interrumpe el flujo normal de las instrucciones del programa.

Las excepciones son puntos de interrupción predeterminados en el flujo del programa que ayudan a finalizar de manera segura una rutina en un escenario de error previsto.

<https://docs.oracle.com/javase/tutorial/essential/exceptions/definition.html>

Falta cercano

```
public void foo() {  
    try {  
        String content = Files.readString(Path.of("file.txt"));  
  
        try {  
            int size = Cipher.getMaxAllowedKeyLength("AES");  
        } catch (NoSuchAlgorithmException ex) {  
            // do something  
        }  
  
    } catch (IOException ex) {  
        // do something  
    }  
}
```


Editor hints

```
92 public boolean foo() {  
93     boolean success = false;  
94  
95     try {  
96         String content = Files.readString(Path.of("file.txt"));  
97  
98         if (content != null && content.isEmpty()) {  
99             int size = Cipher.getMaxAllowedKeyLength("AES");  
100  
101  
102  
103  
104         } catch (IOException ex) {  
105             // do something  
106         }  
107         return success;  
108     }  
109 }
```

- 💡 Add throws clause for java.security.NoSuchAlgorithmException
- 💡 Add catch Clause
- 💡 Surround Statement with try-catch
- 💡 Remove unused "size" >

Conocimiento del Libro

```
public void foo() throw UnexpectedException {  
    try {  
        if(fail) {  
            throw new MyOwnException('message');  
        }  
    } catch (Exception ex) {  
        System.err.println(ex.getMessage());  
    } finally {  
        // clean up  
    }  
}
```

Testing I

```
@Test
void check() throws MyException {
    assertThrows(MyException.class, () -> {
        MyObject.foo();
    });
}
```


Logging & Test Driven Development

Maven / Junit 5 / Logback

```
INFO 2022-09-20 16:12:03 org.europa.together.service.MailClientService | instance class
ERROR 2022-09-20 16:12:03 org.europa.together.application.JdbcActions | SQLException: ERROR: duplicate key value violates unique constraint
  Detail: Key (idx)=(88888888-4444-4444-4444-ccccccccc) already exists.
DEBUG 2022-09-20 16:12:03 org.europa.together.application.JdbcActions | File (org/europa/together/sql/configuration-test.sql): INSERT INTO AF
DEBUG 2022-09-20 16:12:03 o.e.together.application.ConfigurationHbmDAOTest | TEST CASE: updateNotExist
TRACE 2022-09-20 16:12:03 o.e.together.application.ConfigurationHbmDAOTest | TEST CASE TERMINATED.
DEBUG 2022-09-20 16:12:03 org.europa.together.utils.StringUtils | generateUUID() da790e2b-5618-470e-bab7-f854ed22da7c
DEBUG 2022-09-20 16:12:03 org.europa.together.application.JdbcActions | File (org/europa/together/sql/configuration-test.sql): INSERT INTO AF
```

Mantenibilidad - API Design



- Desvincular Dependencias
- Pattern: Proxy, Wrapper & Facade
- Las Excepciones de las Bibliotecas también deben encapsularse

Jackson JSON

```
public String serializeAsJsonObject(final T object)
    throws MyJsonProcessingException {
    try {
        ObjectMapper mapper = new ObjectMapper();
        return mapper.writeValueAsString(object);
    } catch (JsonProcessingException ex) {
        throw new
            MyJsonProcessingException(ex.getOriginalMessage());
    }
}
```

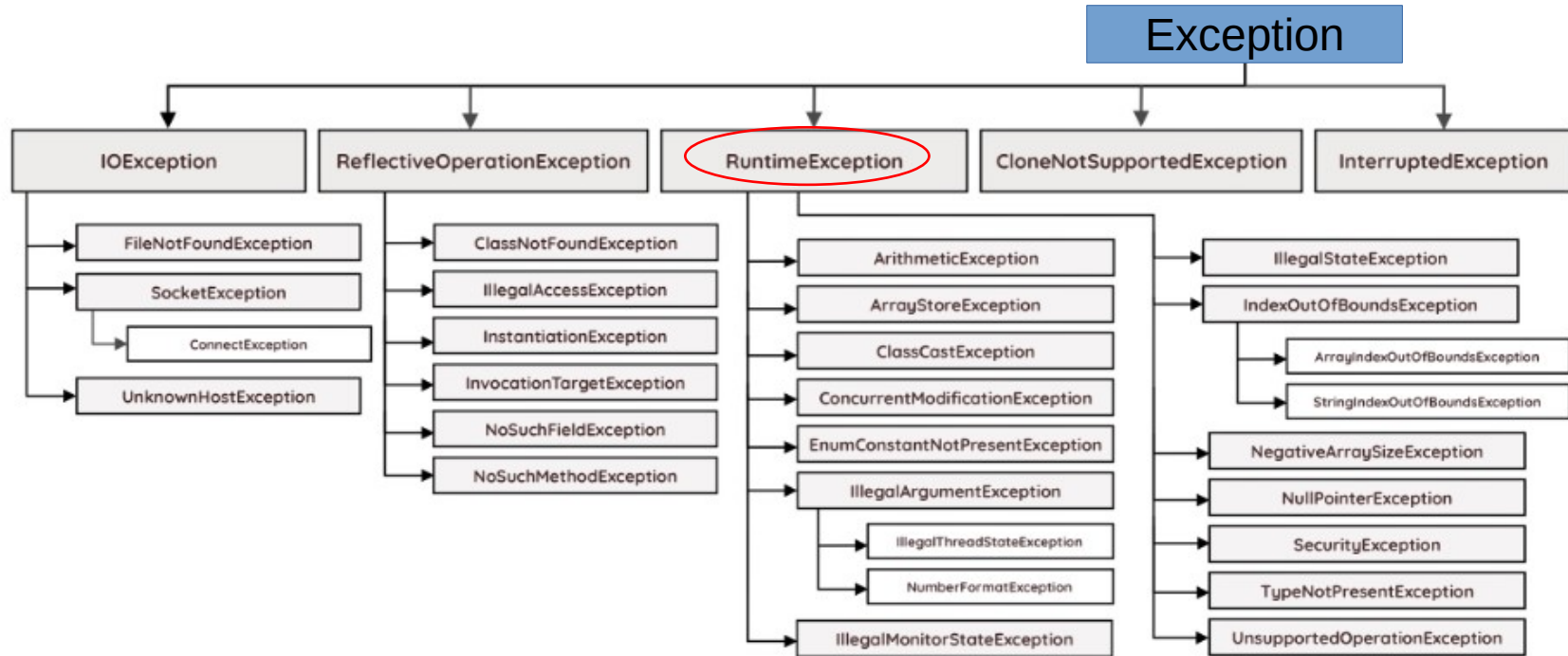
Mi propio privado - Excepciones

```
public class MyException extends Exception {  
    public MyException(final String msg) {  
        super(msg);  
    }  
}
```

Testing II

```
@Test()  
void exceptionObject() {  
    MyException ex =  
        new MyException("MyException");  
    assertEquals("MyException", ex.getMessage());  
}
```

Jerarquía



Error contra Excepcion

- las llamadas excepciones no comprobadas
- representan Errores graves
OutOfMemoryError
- termina el Programa
- también puede ser 'atrapado'



No todos los métodos son adecuados para capturar o lanzar excepciones.

- Regla 1: Low Level Métodos lanzan excepciones
- Regla 2: High Level Métodos capturan excepciones
- Regla 3: todo lo que hay entre 'solo' pasa

Ejemplo

- Hibernate ORM – EntityManager // tirar
- Las clases DAO o construcciones similares delegan lanzando la excepción a 'n' arriba
- ninguna información es "tragada" como resultado
- Clases de servicio / IU, etc. luego inicie el manejo de errores

Aplicacion

```
public void foo() throw UnexpectedException {  
    try {  
        if(fail) {  
            throw new MyOwnException('message');  
        }  
    } catch (Exception ex) {  
        System.err.println(ex.getMessage());  
    } finally {  
        // clean up  
    }  
}
```

NullPointerException

```
public List<Integer> foo() {  
    List<Integer> result = new ArrayList<>();  
  
    // do something  
  
    return result;  
}
```

Microservices

```
public Response fetchRes(@PathParam("res") String resName) {
    Response response = null;
    try {
        ResourcesDO resource = resourceDAO.find(resName);
        response = Response.status(Response.Status.OK)
            .type(MediaType.APPLICATION_JSON)
            .entity(resourceDAO.asJson(resource))
            .encoding("UTF-8")
            .build();
    } catch (EmptyResultDataAccessException | NoResultException ex) {
        response = Response.status(Response.Status.NOT_FOUND).build();
    } catch (Exception ex) {
        response = Response.status(Response.Status.INTERNAL_SERVER_ERROR).build();
    }
    return response;
}
```


Referenzen

[1] <https://www.yegor256.com/2022/08/30/dont-group-exception-catchers.html>

[2] Clean Code, 2009, Robert C. Martin, ISBN 0-13-235088-2

[3] Effective Java 3rd Edition, 2018, Joshua Bloch, ISBN-10: 0-13-468599-7

Marco Schulz, 2021, Continuous Integration mit Jenkins, Rheinwerk, ISBN: ISBN 978-3-8362-7834-8
<https://www.rheinwerk-verlag.de/continuous-integration-mit-jenkins/>



Credentials



Homepage : <https://elmar-dott.com>
GitHub : <https://github.com/ElmarDott>
AnchorFM : <https://anchor.fm/elmar-dott>
Twitter : <https://twitter.com/ElmarDott>
Speaker Deck : <https://speakerdeck.com/elmardott>

Lbry : <https://lbry.tv/@elmar.dott:8>
BitChute : <https://www.bitchute.com/channel/3IyCzKdX8Ip0/>



Danke / thank you / Gracias



BITCHUTE

