



Exceções

Você já viu esse tema na Programação Orientada a Objetos, mas vamos fazer uma revisão e adicionar alguns novos conceitos que serão úteis para esse assunto.

O que são as exceções?

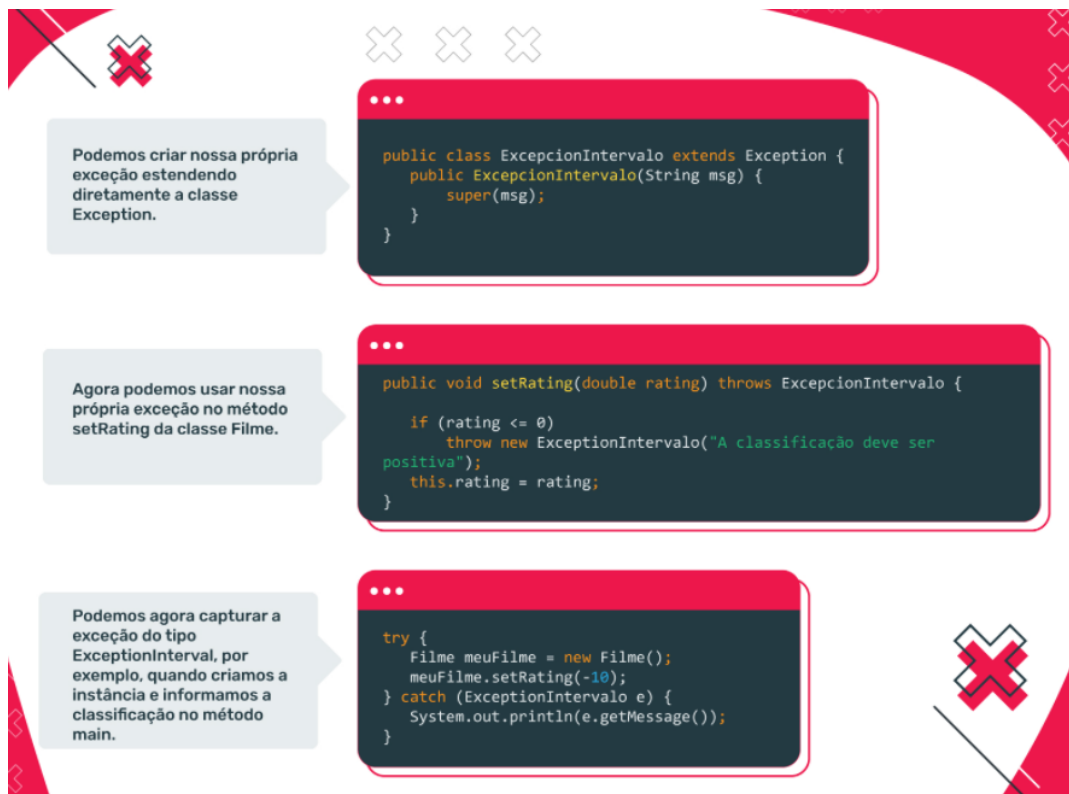
São os erros que podem ocorrer em nossas aplicações Java em tempo de execução, ou seja, aqueles que são gerados durante a execução da aplicação e levam ao seu encerramento inesperado.

O importante é saber que, como programadores, devemos nos antecipar a esses erros que podem fazer nossa aplicação parar de funcionar.

Existem muitos tipos de exceção e, quando ocorrem, é criado um objeto de uma determinada classe dependendo do tipo de erro ocorrido. Esse objeto conterá as informações sobre o erro e nos fornecerá métodos para obter essas informações.

Exception

No Java, podemos capturar as exceções e, assim, evitar que nosso programa pare quando elas ocorrem. Para isso, precisamos colocar em código um bloco Try catch.



Podemos criar nossa própria exceção estendendo diretamente a classe `Exception`.

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

Agora podemos usar nossa própria exceção no método `setRating` da classe `Filme`.

```
public void setRating(double rating) throws ExceptionIntervalo {
    if (rating <= 0)
        throw new ExceptionIntervalo("A classificação deve ser positiva");
    this.rating = rating;
}
```

Podemos agora capturar a exceção do tipo `ExceptionIntervalo`, por exemplo, quando criamos a instância e informamos a classificação no método `main`.

```
try {
    Filme meuFilme = new Filme();
    meuFilme.setRating(-10);
} catch (ExceptionIntervalo e) {
    System.out.println(e.getMessage());
}
```

Exceções com o Spring Boot

O tratamento de exceções e erros nas APIs, e o envio da resposta apropriada ao cliente são necessários para aplicações empresariais. É por isso que devemos aprender a lidar com exceções com o Spring Boot que, como já vimos, torna mais fácil para nós desenvolver aplicações.

O Spring Boot nos oferece várias alternativas quando se trata de lidar com diferentes erros, veremos em detalhes quais são.

Códigos de status

Abaixo, disponibilizamos todos os códigos de status possíveis que podem ser retornados no HTTP, para que você possa consultá-los quando precisar.

Conclusão

Por meio de um try catch podemos utilizar o `ResponseStatusException` para processar os erros desse método em específico.

Desta forma, vimos como podemos processar os diferentes erros em relação a aplicação, a classe e ao método.

Agora chegou o momento de praticar!

Exercício

Projeto Transportadora - Parte 1

Continuamos trabalhando no projeto que iniciamos na aula 39. Nesta etapa você tratará os possíveis erros em sua aplicação.

O líder técnico nos pede para:

- Criar uma Exception chamada **BadRequestException**;
- Lançar essa exceção a partir do serviço de pacote, caso seja realizada uma tentativa de registrar um pacote com um **código** que pertença a outro pacote;
- Usando `@ExceptionHandler`, crie um método no controller do pacote para processar essa exceção lançada no serviço do pacote;
- Retorne um `ResponseEntity` com código 400 (BAD REQUEST).

Por último, você deve testar se está funcionando corretamente utilizando o Postman.