

Mãos na massa: Finally e try with resources

Vamos começar fazendo alguns testes agora.

1) Comece criando a classe `Conexao`:

```
public class Conexao{

    public Conexao() {
        System.out.println("Abrindo conexão");
    }

    public void leDados() {
        System.out.println("Recebendo dados");
        // throw new IllegalStateException();
    }

    public void fecha() {
        System.out.println("Fechando conexão");
    }
}
```

2) Agora para testarmos nossa conexão, precisaremos criar outra classe, a `TesteConexao`. Não se esqueça do método `main`:

```
public class TesteConexao {

    public static void main(String[] args) {

    }
}
```

3) Dentro do método `main`, de nossa classe recém-criada, colocaremos um bloco `try` e `catch` para fazermos uso da nossa conexão:

```
try{
    Conexao con = new Conexao();
    con.leDados();
    con.fecha();
} catch(IllegalStateException ex){
    System.out.println("Deu erro na conexão");
}
```

Não esqueça de remover o comentário da linha respectiva dentro da classe `Conexao`.

4) Queremos fechar nossa conexão sempre, mesmo em caso de erros, então precisamos garantir que o método `con.fecha()` sempre seja chamado. Para isso, temos o bloco `finally`. Faça o seguinte:

```
Conexao con = null;
try{
    con = new Conexao();
    con.leDados();
    con.fecha();
} catch(IllegalStateException ex){
    System.out.println("Deu erro na conexão");
} finally {
    con.fecha();
}
```

5) Ainda podemos melhorar o nosso código, para isso, faremos a declaração da conexão dentro dos parênteses do `try`. Para isso, faça o seguinte:

```
try(Conexao conexao = new Conexao() ){
}
```

6) Por conta da declaração acima, precisamos fazer com que nossa classe `Conexao` implemente a interface `AutoCloseable` e implemente o método `close()`. Deixaremos então do seguinte modo:

```
public class Conexao implements AutoCloseable{

    public Conexao() {
        System.out.println("Abrindo conexao");
    }

    public void leDados() {
        System.out.println("Recebendo dados");
        throw new IllegalStateException();
    }

    @Override
    public void close() {
        System.out.println("Fechando conexao");
    }
}
```

7) Agora faremos a chamada do método `leDados()` dentro do nosso novo `try`, comentando o código antigo.

```
try(Conexao conexao = new Conexao() ){
    conexao.leDados();
}
```

8) Ainda precisamos fazer o nosso novo `catch` também. Que ficará do seguinte modo:

```
try(Conexao conexao = new Conexao() ){
    conexao.leDados();
} catch(IllegalStateException ex){
```

```
System.out.println("Deu erro na conexão");  
}
```