

03

Qual catch?

Já vimos nesse curso duas formas de capturar várias exceções através do bloco `catch`. (1) A forma tradicional, que funciona desde inicio do Java, simplesmente repete o bloco `catch` para cada exceção:

```
try {
    metodoPerigosoQuePodeLancarVariasExcecoes();
} catch(MinhaExcecao ex) {
    ex.printStackTrace();
} catch(OutraExcecao ex) {
    ex.printStackTrace();
}
```

E (2) a forma mais atual, que foi introduzido no Java 7, permite definir as várias exceções no mesmo bloco (*multi-catch*):

```
try {
    metodoPerigosoQuePodeLancarVariasExcecoes();
} catch(MinhaExcecao | OutraExcecao ex) { //multi-catch
    ex.printStackTrace();
}
```

Você vai encontrar ambas as formas no seu dia-a-dia de desenvolvedor Java. Agora, veja **assinatura** do "método perigoso" em questão:

```
//funciona, podemos colocar duas exceções na assinatura
public void metodoPerigosoQuePodeLancarVariasExcecoes() throws MinhaExcecao, OutraExcecao-
    //código omitido
}
```

Vimos mais uma variação do *catch*, não sintática, mas conceitual. Qual afirmação abaixo pode ser usado para capturar todas exceções desse "método perigoso"?

Importante: Assume que ambas as exceções são *checked!*

Selezione uma alternativa

A

```
try {
    metodoPerigosoQuePodeLancarVariasExcecoes();
}
```

B

```
try {
    metodoPerigosoQuePodeLancarVariasExcecoes();
} catch(RuntimeException ex) {
```

```
    ex.printStackTrace();  
}
```

C

```
try {  
    metodoPerigosoQuePodeLancarVariasExcecoes();  
} catch(Exception ex) {  
    ex.printStackTrace();  
}
```