



**Certified Tech
Developer**

The Ultimate Degree



Override

Redefinição de métodos, como evitar erros?

Objetivo

Vimos que para redefinir um método de uma superclasse, devemos sobrescrevê-lo, ou seja, devemos escrever o mesmo método com a mesma assinatura da subclasse

No entanto, o que acontecerá se, ao tentarmos sobrescrever esse método na subclasse, digitarmos incorretamente o nome ou esquecermos alguns de seus parâmetros? O que acontecerá é que a assinatura deste método seria diferente da classe pai e se estivéssemos tentando sobrescrever, esse erro seria descoberto durante a compilação, exceto como um método abstrato.

Mas, não se preocupe...

Em Java, há uma anotação chamada `@override` que podemos escrever para o método que estamos substituindo.

Com esta anotação, o compilador irá validar que o método que estamos sobrescrevendo possui uma assinatura:

```
public class Dog{  
    public String bark(){  
        return "auau";  
    }  
}
```

```
public class Doberman extends Dog{  
    @Override  
    public String bark(){  
        return "WOW WOW !!!";  
    }  
}
```

A anotação "@Override" pede ao compilador para verificar se estamos realmente redefinindo um método da superclasse. A seguir, mostramos um exemplo onde o compilador emitirá um erro, não encontrando na Superclasse Dog, um método com a mesma assinatura onde colocamos a anotação @Override.

Até a próxima!