

EducaCiência FastCode

Fala Galera,

o Artigo: 22/2020 Data: Maio/2020

o Público Alvo: Desenvolvedores – Iniciantes

o Tecnologia: Java

o Tema: Assinatura @Override

Link: https://github.com/perucello/DevFP

Neste artigo, vamos tentar sanar algumas dúvidas sobre a Assinatura @Override. Antes de mais nada, precisamos entender o porquê desta assinatura, sendo assim, vamos tentar dedicar um tempo a este assunto.







Esta assinatura @Override, serve realmente para informar-lhe que o método anotado (@Override) está sobrescrevendo um método da Superclasse.

Partindo para o princípio do nosso artigo CRUD-Inserir "Artigo 20 - CRUD método Inserir", no nosso pacote Model, temos nossa classe responsável pelos métodos Getters/Setters e exatamente após esta implementação fizemos a anotação @Override informando que o retorno "nome" seria um ToString anotado em @Override.

Isso significa que no código , o @Override anota que os métodos neste caso, ToString() sobrescreve os métodos da nossa Superclasse – em alguns casos a própria IDE sugere esta anotação ao desenvolvedor.

Sendo assim, isso só fará sendo ao programador quando realmente recebe um retorno ou faz algo de errado.

Em outras palavras, esta anotação, informa que está sobrescrevendo um método que está sendo na realidade herdado, e não fazendo uma sobrecarga no método.

Vale lembrar, que sem a anotação @Override, o código irá compilar normalmente, porém, neste momento sem esta anotação, você sobrecarregará o método e não sobrescrevendo. Esta anotação, foi implementada após o Java, justamente para ser usada em métodos que sobrescrevem métodos herdados de suas superclasses.

"The **@Override** annotation is one of a default Java annotation and it can be introduced in **Java 1.5** Version.

The **@Override** annotation indicates that the **child class method is over-writing its base class method**.

The **@Override** annotation can be useful for two reasons

It extracts a warning from the compiler if the annotated method doesn't actually override anything.

It can improve the readability of the source code."

```
class BaseClass {
   public void display() {
        System.out.println("In the base class,test() method");
    }
} class ChildClass extends BaseClass {
    @Override
    public void display() {
        System.out.println("In the child class, test() method");
    }
} // main class
public class OverrideAnnotationTest {
    public static void main(string args[]) {
        System.out.println("Goverride Example");
        BaseClass test = new ChildClass();
        test.display();
    }
}

Resultado

@Override Example
In the child class, test() method
```

Fonte: https://www.tutorialspoint.com/importance-of-override-annotation-in-java

Até mais! Espero ter ajudado!

