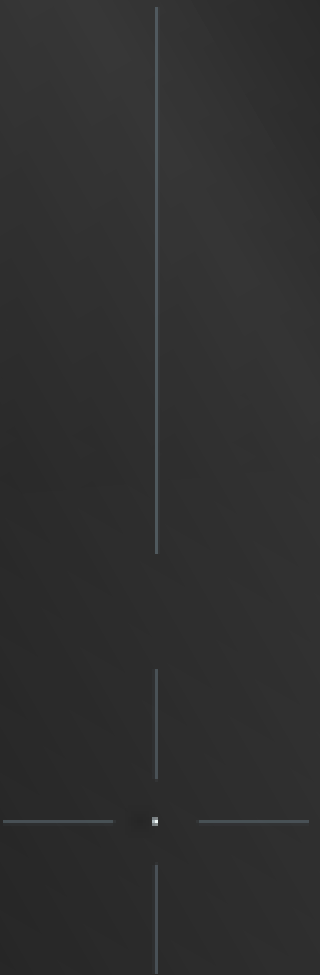


# JAVADOC

*Implementando documentação ao seu código fonte*

Prof. Benefrancis do Nascimento



# TÓPICOS

■ Definição e Forma de Funcionamento

---

■ Criando JAVADOC no IDE Eclipse

---

■ Criando JAVADOC no prompt

---

■ Atividades Complementares

---

■ Conclusão

---



# Definição

JavaDoc é um utilitário fornecido pela *Sun Microsystems* junto ao JDK para gerar documentação de códigos Java em formato HTML a partir de comentários no código fonte *Java*.

O *JavaDoc* é suportado pela maioria dos IDEs de mercado.



# Propósito desta aula

## ■ DEFINIR O QUE É JAVADOC

Entender a importância de se documentar o código fonte.

## ■ GERAR O JAVADOC NO ECLIPSE IDE

Esta é a maneira mais fácil de se executar o Javadoc. A maioria dos programadores Java sabem fazer assim.

## ■ TÉCNICAS PARA CRIAÇÃO DE DOCUMENTAÇÃO

Saber que é possível utilizar tags HTML e tags reservadas pelo próprio Javadoc

## ■ GERAR JAVADOC PELO PROMPT

É importante saber o que está acontecendo por baixo dos panos. Todo programador "raiz" tem esta preocupação. Pouquíssimos programadores Java sabem como fazer.



# Forma de funcionamento

Seu funcionamento baseia-se na inserção de textos explicativos em forma de um comentário especial, que antecede um escopo de uma classe ou método, tendo assim, a responsabilidade de apresentar o mesmo.

```
/**  TEXTO EXPLICATIVO  */
```

Esses comentários poderão conter tags especiais que especificam quais informações serão inseridas, com objetivo de manter uma massa de conhecimento reutilizável em qualquer projeto que faça uso da classe em questão. Dentro do comentário JavaDoc é possível usar tags HTML e tags reservadas do JavaDoc precedidas pelo caracter “@”, veremos a relação das tags reservadas mais utilizadas na tabela do próximo slide.

```
/**  
<h1>Inserindo a explicação</h1>  
@author Benefrancis do Nascimento  
@version 1.0  
*/
```



# Forma de funcionamento

TAG	SIGNIFICADO
@author	Identifica o autor da classe ou do método.
@deprecated	Identifica classes ou métodos obsoletos, e que não se recomenda seu uso porque possivelmente desaparecerá em versões posteriores. É interessante informar nessa tag, quais métodos ou classes podem ser usadas como alternativa ao método obsoleto.
@link	Possibilita a definição de um link para outro documento local ou remoto através de um URL.
@param	Mostra um parâmetro que será passado a um método.
@return	Mostra qual o tipo de retorno de um método. Não se deve pode usar em construtores ou métodos "void"
@see	Possibilita a definição referências de classes ou métodos, que podem ser consultadas para melhor compreender idéia daquilo que está sendo comentada.
@throws	Indica os tipos de exceções que podem ser lançadas pelo método.

# Vamos codificar!

Vamos exercitar a criação de classes com comentários JAVADOC e em seguida criaremos os arquivos de documentação

# Atividades Complementares

Leiam o artigo

"How to Write Doc Comments for the Javadoc Tool"

<https://www.oracle.com/technical-resources/articles/java/javadoc-tool.html>

Explique para mim, na próxima aula, o que acontece quando

incluímos no comentário tipo javaDOC o texto abaixo:

**@version %I%, %G%**



# Atividades Complementares

1º) Pesquisem na internet script para validação de CNPJ, CPF e PIS / PASEP

2º ) implementem os métodos :

`public static boolean isValid(String numero)` das classes CPF, CNPJ ;

3º) Criem a classe PISPASEP implementando a interface

`DocumentoPessoaFisicaFederal` e também implementem o método

`public static boolean isValid(String numero);` e

4º) Criem os comentários JAVADOC para os métodos e para a Classe PISPASEP.  
testaremos o seu código nas próximas aulas.



# Conclusão

Nesta aula nós conhecemos o JAVADOC, criamos algumas classes e inserimos tags HTML e tags reservadas pelo javaDoc.

Em seguida, geramos documentação por meio da Eclipse IDE. Por fim, vimos como gerar o JavaDoc por meio do prompt de comando.



# Agradeco!

profBenefrancis.Nascimento@fiap.com.br

Baixe os códigos desta aula em:

<https://github.com/Benefrancis/fiap>

