

# Programação Orientada a Objetos e a Linguagem Java

Rodrigo da Cruz Fujioka  
[rodrigofujioka@gmail.com](mailto:rodrigofujioka@gmail.com)



# Objetivos

[www.rodrigofujioka.com](http://www.rodrigofujioka.com)

- Apresentar a forma de criar documentação de código

*Rodrigo Fujioka - Ling de Prog 2*



# Documentação

[www.rodrigofujioka.com](http://www.rodrigofujioka.com)

- Quem escreveu não deve ser o único a poder ler o código
- Após desenvolvido um sistema precisa ser mantido (modificações, ajustes, novos requisitos, ..)
- Muitas vezes o software não é mantido pela mesma equipe que desenvolveu

*Rodrigo Fujioka - Ling de Prog 2*



# Documentação

[www.rodrihofujioka.com](http://www.rodrihofujioka.com)



*Rodrigo Fujioka - Ling de Prog 2*



# Documentação

[www.rodigofujioka.com](http://www.rodigofujioka.com)



Rodrigo Fujioka - Ling de Prog 2 2





# Documentação

[www.rodrihofujioka.com](http://www.rodrihofujioka.com)



Rodrigo Fujioka - Ling de Prog 2 2



# Documentação

[www.rodrigofujioka.com](http://www.rodrigofujioka.com)

- Assim, o software deve conter documentação
- Quando a documentação está separada do código, torna-se difícil mantê-la.

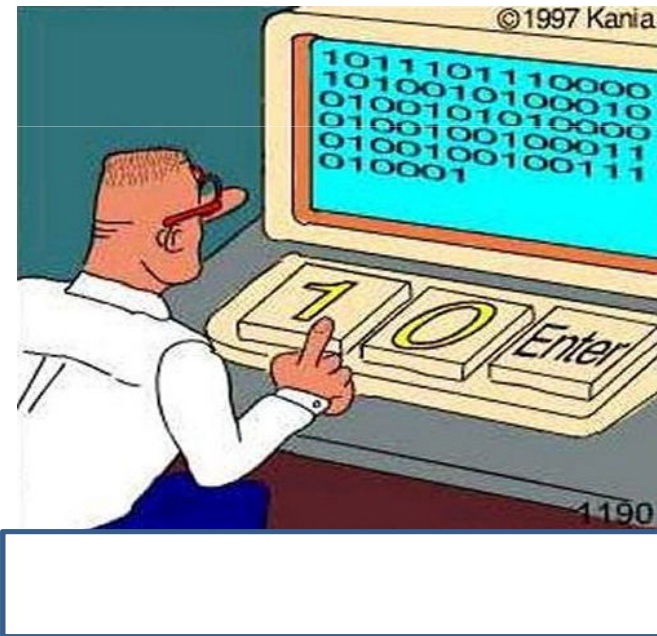
*Rodrigo Fujioka - Ling de Prog 2*



# Documentação

www.rodriogofujioka.com

- Assim, o software deve conter documentação
- Quando a documentação está separada do código, torna-se difícil mantê-la.



Rodrigo Fujioka - Ling de Prog 2 2



# Javadoc

[www.rodrihofujioka.com](http://www.rodrihofujioka.com)

- O J2SE possui uma ferramenta denominada javadoc que gera documentação html a partir do código fonte e dos comentários contidos nele.
- A documentação gerada possui um formato padrão
- A API Java disponibilizada pela Sun é toda documentada utilizando o javadoc.
- Existem comentários de classe, de método, de variável

*Rodrigo Fujioka - Ling de Prog 2*



# Javadoc

[www.rodrihofujioka.com](http://www.rodrihofujioka.com)

↙  
/\*\* Comentário de classe \*/

public class TesteDoc {

↙  
/\*\* Comentário de variável \*/

public int i;

↙  
/\*\* Comentário de método \*/

public void m();

}

Rodrigo Fujioka - Ling de Prog 2



# Javadoc

www.rodrigofujioka.com

- Comentários **javadoc** com HTML embutido:

```
/**  
 * Você pode <b>até</b> inserir uma lista em  
 * HTML  
 * <ol>  
 * <li> Item 1  
 * <li> Item 2  
 * <li> Item 3  
 * </ol>  
 */
```

- Existem tags específicas que podem ser colocadas dentro dos comentários.

Rodrigo Fujioka - Ling de Prog 2



# Javadoc

www.rodrigofujioka.com

- Comentários **javadoc** com HTML embutido:

```
/**  
 * Você pode <b>até</b> inserir uma lista em  
 * HTML  
 * <ol>  
 * <li> Item 1  
 * <li> Item 2  
 * <li> Item 3  
 * </ol>  
 */
```

- Existem tags específicas que podem ser colocadas dentro dos comentários.

Rodrigo Fujioka - Ling de Prog 2



# Javadoc

www.rodrigofujioka.com

## Tags gerais:

- **@deprecated** – *adiciona um comentário de que a classe, método ou variável não deve ser usada.*
  - *@deprecated <texto explicativo e possivelmente indicando o elemento (classe, método ...) que substitui o atual>*
- **@since** – *indica a versão do JDK quando o elemento em questão foi adicionado à API Java*
  - *@since <versão (Ex: 1.4) >*

Rodrigo Fujioka - Ling de Prog 2





# Javadoc

www.rodriofujioka.com

## Tags gerais:

- **@deprecated** – *adiciona um comentário de que a classe, método ou variável não deve ser usada.*
  - *@deprecated <texto explicativo e possivelmente indicando o elemento (classe, método ...) que substitui o atual>*
- **@since** – *indica a versão do JDK quando o elemento em questão foi adicionado à API Java*
  - *@since <versão (Ex: 1.4) >*

Rodrigo Fujioka - Ling de Prog 2



# Javadoc

www.rodrigofujioka.com

- Tags gerais:
  - *@see* – adiciona um link à seção "Veja também" da documentação
    - *@see <nome da classe>*
    - *@see <nome da classe#nome método>*

Rodrigo Fujioka - Ling de Prog 2



# Javadoc

www.rodrigofujioka.com

Tags para comentários de classes e interfaces:

- **@author** – *autor do elemento*
  - *@author <informação sobre o autor (nome, e-mail, site ...)>*
- **@version** – *número da versão atual*
  - *@version <número da versão>*

Rodrigo Fujioka - Ling de Prog 2



# Javadoc

www.rodrigofujioka.com

Tags para comentários de métodos:

- *@param* – *descreve os parâmetros de um método acompanhado por uma descrição*
  - *@param <nome do parâmetro> <descrição>*
- *@return* – *descreve o valor retornado*
  - *@return <descrição>*
- *@throws* – *indica as exceções que um dado método lança*
  - *@throws <nome da classe> <descrição>*

Rodrigo Fujioka - Ling de Prog 2



# Javadoc

www.rodrigofujioka.com

Tags para comentários de métodos:

- *@param* – *descreve os parâmetros de um método acompanhado por uma descrição*
  - *@param <nome do parâmetro> <descrição>*
- *@return* – *descreve o valor retornado*
  - *@return <descrição>*
- *@throws* – *indica as exceções que um dado método lança*
  - *@throws <nome da classe> <descrição>*

Rodrigo Fujioka - Ling de Prog 2





# Javadoc

[www.rodriogofujioka.com](http://www.rodriogofujioka.com)

```
/**
 * Esta classe representa uma calculadora
 * @author Rodrigo Fujioka
 * @author rodrigofujioka@gmail.com
 * @version 1.0
 * @since 1.5
 */
public class Calculadora {

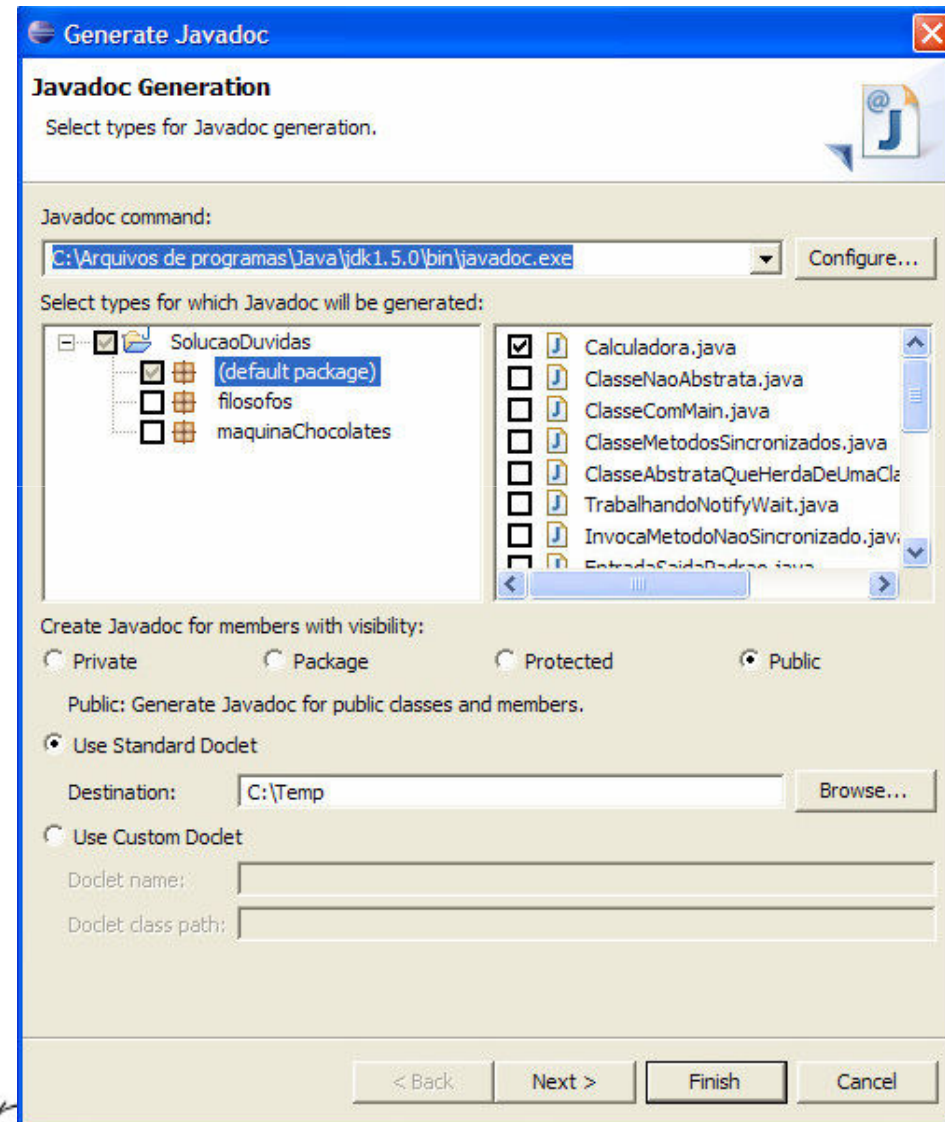
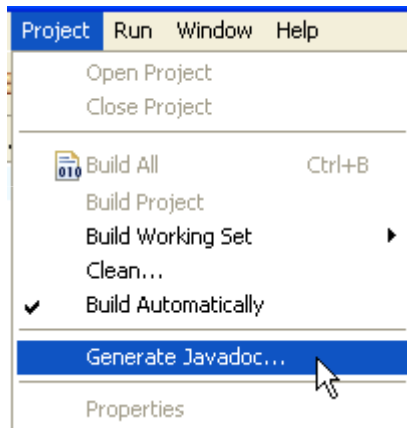
    /**
     * Método responsável por somar 2 números
     * @param a primeiro número a ser somado
     * @param b segundo número a ser somado
     * @return soma dos números passados como parâmetros
     */
    public double somar(double a, double b) {
        return a + b;
    }
}
```

*Rodrigo Fujioka - Ling de Prog 2*



# Gerando Documentação - Eclipse

www.rodrigofujioka.com



Rodrigo Fujioka - Lingua de 1 rogo 2 2

# Como Usar a Documentação

www.rodrigofujioka.com

## Descrição da classe escolhida

- hierarquia
- documentação detalhada, propriedades, métodos, etc.

Lista de Pacotes

Lista de classes e interfaces do pacote escolhido

Overview Package Class Use Tree Deprecated Index Help

PREV CLASS NEXT CLASS

SUMMARY: NESTED | FIELD | CONSTR | METHOD

DETAIL: FIELD | CONSTR | METHOD

Frames NO FRAMES

java.swing

Class JOptionPane

java.lang.Object

↳ java.awt.Component

↳ java.awt.Container

↳ javax.swing.JComponent

↳ javax.swing.JOptionPane

All Implemented Interfaces:

ImageObserver, MenuContainer, Serializable, Accessible

public class **JOptionPane**  
extends [JComponent](#)  
implements [Accessible](#)

JOptionPane makes it easy to pop up a standard dialog box that prompts users for a value or informs them of something. For information about using JOptionPane, see [How to Make Dialogs](#), a section in *The Java Tutorial*.

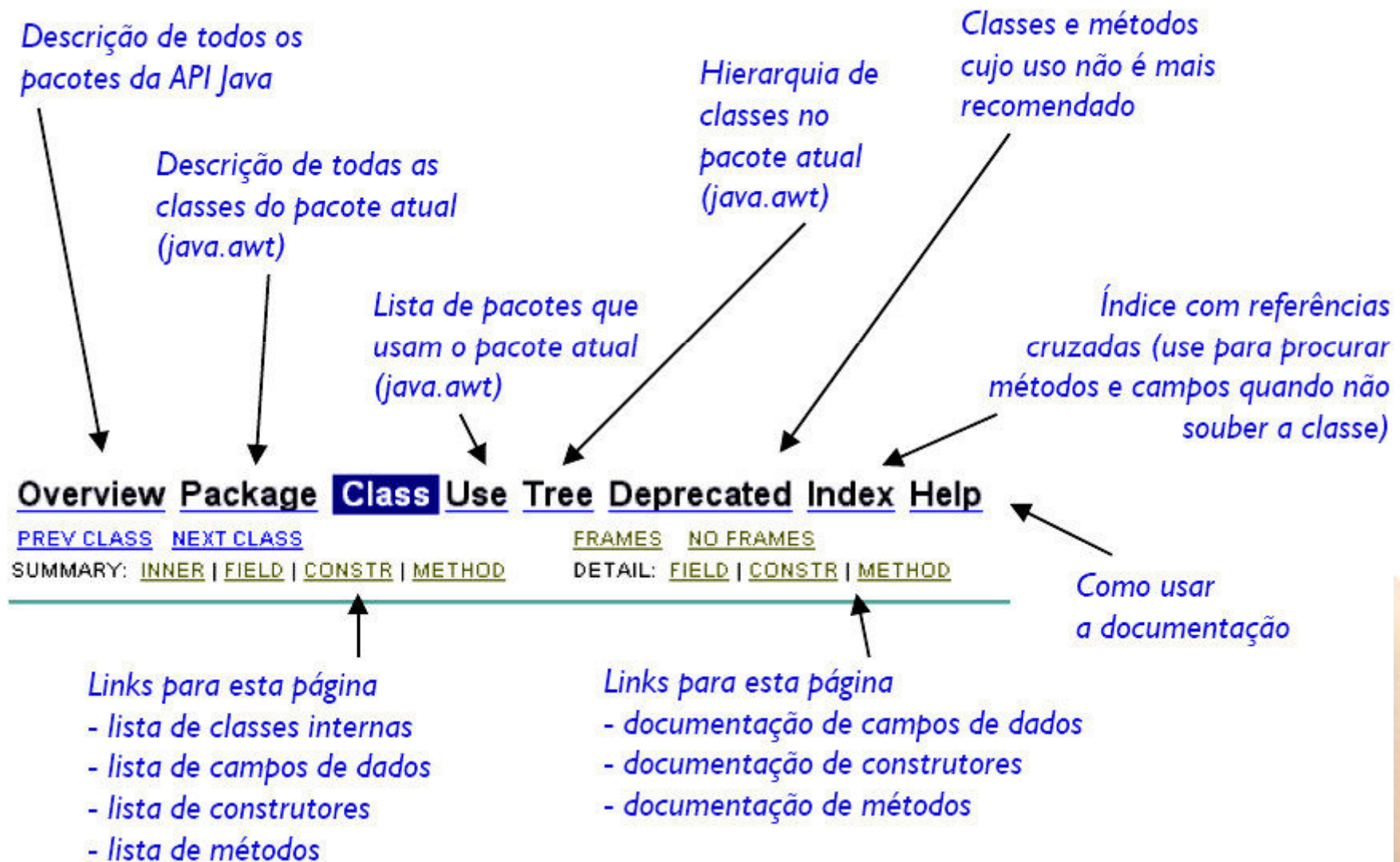
While the JOptionPane class may appear complex because of the large number of methods, almost all uses of this class are one-line calls to one of the static `showXXXDialog` methods shown below:

Method Name	Description
<code>showConfirmDialog</code>	Asks a confirming question, like <code>yes/no/cancel</code> .

Rodrigo Fujioka - Ling de Prog 2

# Como Usar a Documentação

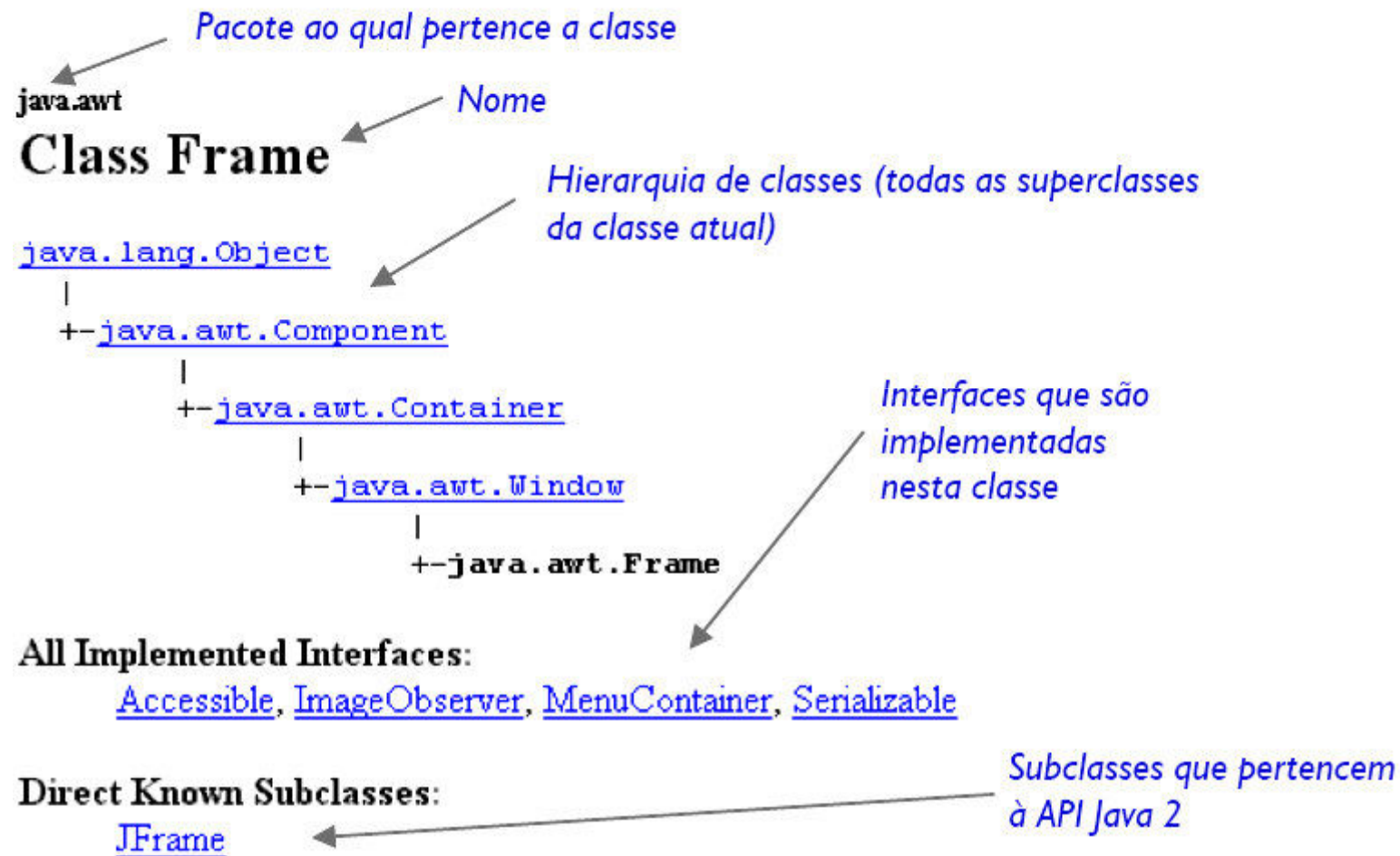
www.rodriofujioka.com





# Como Usar a Documentação

www.rodrihofujioka.com

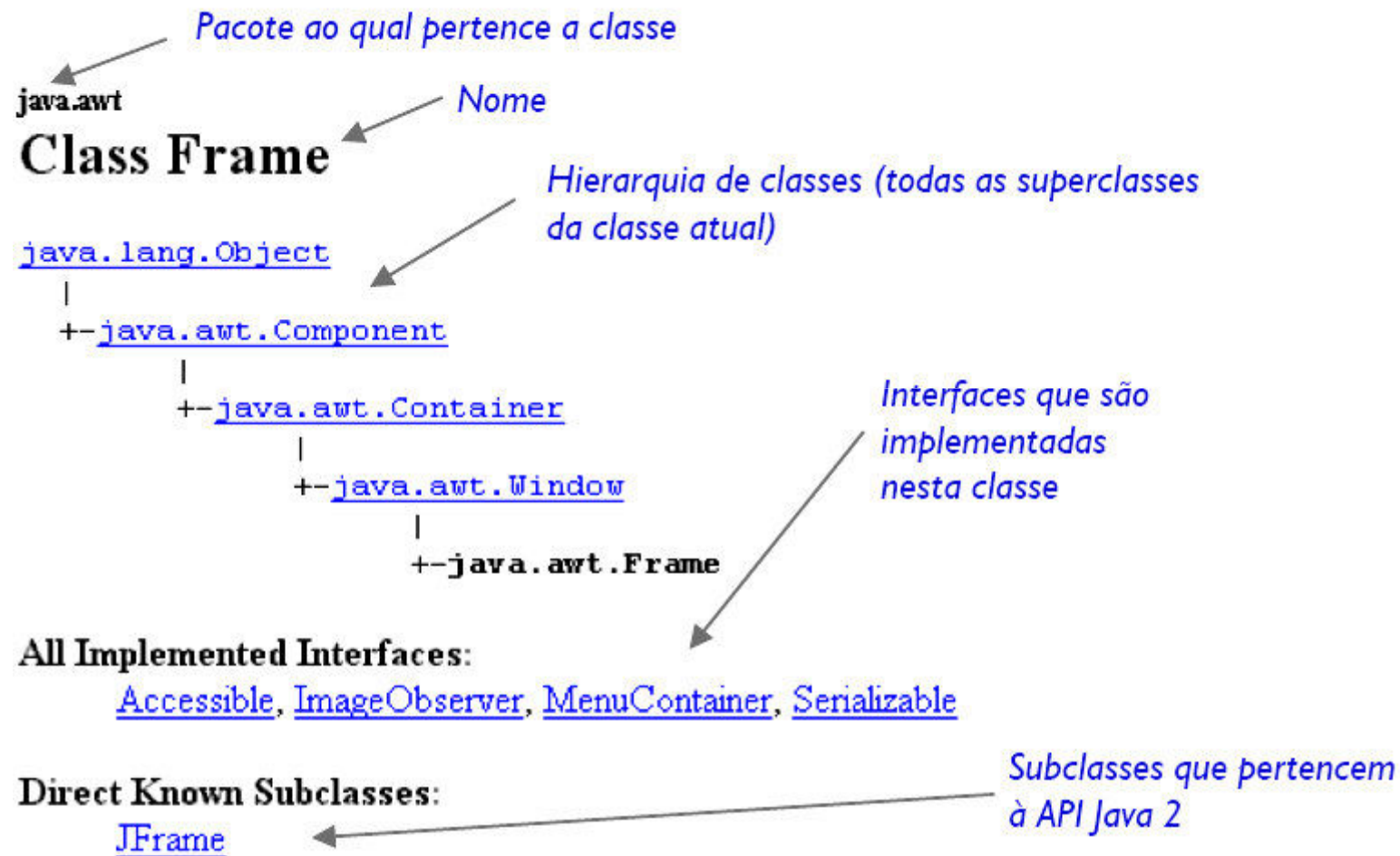


Rodrigo Fujioka - Ling de Prog 2



# Como Usar a Documentação

www.rodrigofujioka.com



Rodrigo Fujioka - Ling de Prog 2

# Como Usar a Documentação

www.rodrigofujioka.com

- Listas de atributos, métodos e construtores

## Field Summary

static long	<a href="#">MAX_VALUE</a> A constant holding the maximum value a long can have, $2^{63}-1$ .
static long	<a href="#">MIN_VALUE</a> A constant holding the minimum value a long can have, $-2^{63}$ .
static int	<a href="#">SIZE</a> The number of bits used to represent a long value in two's complement binary form.
static <a href="#">Class</a> <Long>	<a href="#">TYPE</a> The <code>Class</code> instance representing the primitive type long.

## Constructor Summary

<a href="#">Long</a> (long value)	Constructs a newly allocated Long object that represents the specified long argument.
<a href="#">Long</a> (String s)	Constructs a newly allocated Long object that represents the long value indicated by the String parameter.

## Method Summary

static int	<a href="#">bitCount</a> (long l) Returns the number of one-bits in the two's complement binary representation of the specified long value.
byte	<a href="#">byteValue</a> () Returns the value of this Long as a byte.
int	<a href="#">compareTo</a> ( <a href="#">Long</a> anotherLong)

Rodrigo Fujioka - Ling de Prog 2

# Como Usar a Documentação

www.rodrigofujioka.com

- Listas de atributos, métodos e construtores

## Field Summary

static long	<a href="#">MAX_VALUE</a>	A constant holding the maximum value a long can have, $2^{63}-1$ .
static long	<a href="#">MIN_VALUE</a>	A constant holding the minimum value a long can have, $-2^{63}$ .
static int	<a href="#">SIZE</a>	The number of bits used to represent a long value in two's complement binary form.
static <a href="#">Class</a> <Long>	<a href="#">TYPE</a>	The <a href="#">Class</a> instance representing the primitive type long.

## Constructor Summary

<a href="#">Long</a> (long value)	Constructs a newly allocated Long object that represents the specified long argument.
<a href="#">Long</a> (String s)	Constructs a newly allocated Long object that represents the long value indicated by the String parameter.

## Method Summary

static int	<a href="#">bitCount</a> (long l)	Returns the number of one-bits in the two's complement binary representation of the specified long value.
byte	<a href="#">byteValue</a> ()	Returns the value of this Long as a byte.
int	<a href="#">compareTo</a> ( <a href="#">Long</a> anotherLong)	

Rodrigo Fujioka - Ling de Prog 2

# Como usar a Documentação - Construtor

www.rodrihofujioka.com

## Frame

```
public Frame(String title,  
            GraphicsConfiguration gc)
```

Assinatura do construtor  
(mostra tipos dos parâmetros)

Constructs a new, initially invisible Frame object with the specified title and a GraphicsConfiguration.

O que faz o construtor

### Parameters:

title - the title to be displayed in the frame's border. A null value is treated as an empty string, "".

gc - the GraphicsConfiguration of the target screen device. If gc is null, the system default GraphicsConfiguration is assumed.

Descrição de cada parâmetro

### Throws:

IllegalArgumentException - if gc is not from a screen device.

Exceções que o construtor pode provocar

### See Also:

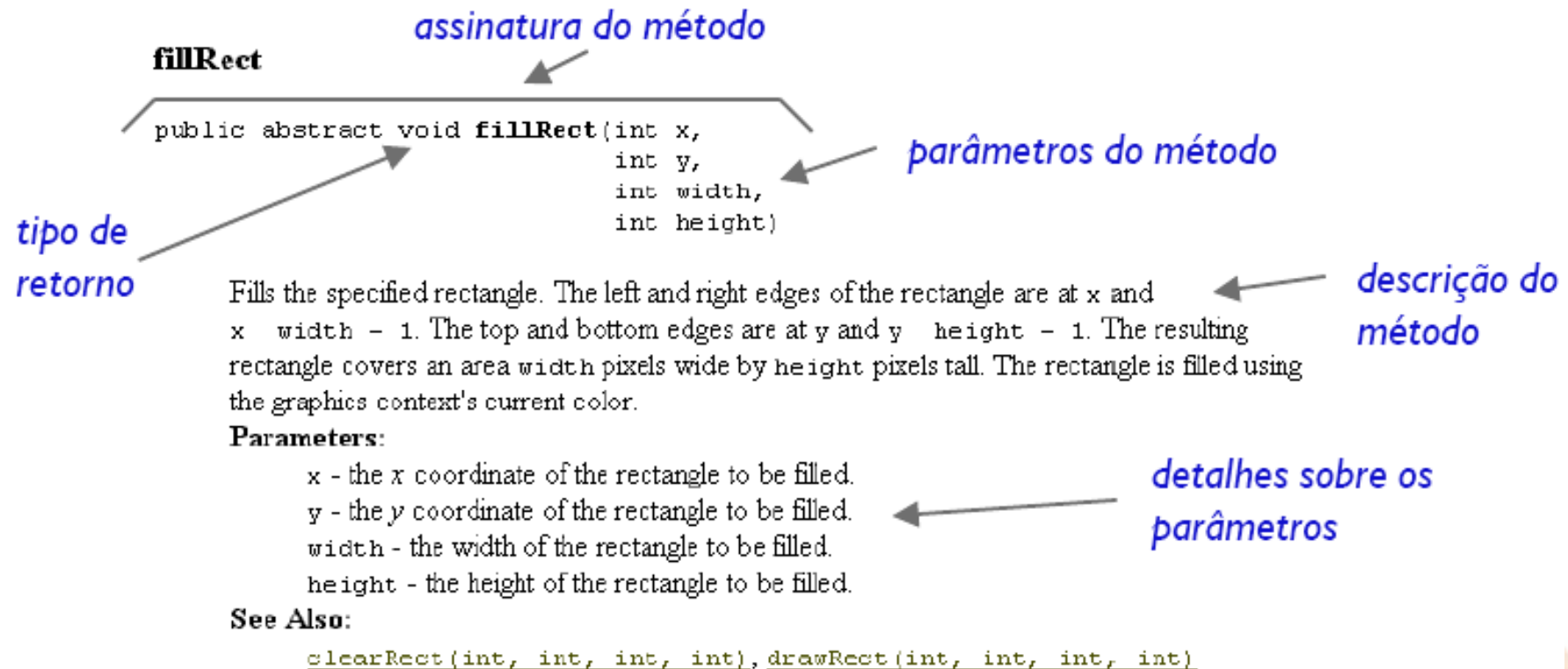
Component.setSize(int, int), Component.setVisible(boolean),  
GraphicsConfiguration.getBounds()

Métodos relacionados

Rodrigo Fujioka - Ling de Prog 2

# Como usar a Documentação - Método

www.rodrigofujioka.com



Rodrigo Fujioka - Ling de Prog 2



# Javadoc

www.rodrigofujioka.com

- On-line:
  - <http://java.sun.com/javase/6/docs/api/>
- Download:
  - <http://java.sun.com/javase/downloads/>

Rodrigo Fujioka - Ling de Prog 2



# Dúvidas?

[www.rodrihofujioka.com](http://www.rodrihofujioka.com)

? ? ? ? ? ? ? ?  
? ? ? ? ? ? ? ?  
? ? ? ? ? ? ? ?  
? ? ? ? ? ? ? ?  
? ? ? ? ? ? ? ?

*Rodrigo Fujioka - Ling de Prog 2*

