3. Documentación.

3.1 Documentación Javadoc con NetBeans

O kit de desenvolvemento da plataforma Java SE, http://www.oracle.com/technetwork/java/javase/tech/index.html, está formada por:

JDK=javac+ferramentas e utilidades (entre as que está javadoc) +JRE(Runtime+Java SE API(librerías, paquetes)+JVM)

A ferramenta javadoc é un xerador de documentación que extrae información sobre as clases, clases internas, métodos, interfaces, e campos baseándose nos comentarios Javadoc e no código fonte, e que gardará nun grupo de arquivos HTML que poderá ser consultado facilmente.

Este proceso de documentación pode facerse directamente con javadoc na liña de comandos ou o que é máis cómodo, pode utilizarse un IDE como NetBeans que mediante un entorno gráfico integrado con outras tarefas de programación permite documentar baseándose na ferramenta estándar javadoc, é dicir, o que chamaremos crear documentación Javadoc.

Baseándose nun proxecto Java de NetBens farase a creación da documentación Javadoc en dúas etapas:

Inserir comentarios Javadoc no código fonte.

Xerar documentación Javadoc para un proxecto. Esta documentación é unha especificación da API (Application Programming Interface) do proxecto que contribuirá a unha mellor comprensión do proxecto por parte do programador. Un exemplo desta documentación é a *Java Platform, Standard Edition 7 API Specification*.

Despois de ter a documentación e de que NetBeans coñeza onde está gardada esa documentación, poderase consultar directamente dende o arquivo fonte. Un paso previo para que NetBeans coñeza onde está gardada a documentación de Java SE é agregala a NetBEans.

Se á documentación Javadoc xerada (ou especificación API) se lle engaden exemplos, definicións de termos comúns de programación, a descrición de erros e as súas solucións, teríase unha guía de programación do proxecto. Un exemplo desta guía é o paquete de documentación de JDK. Cabe destacar que na descrición de erros da guía de programación distínguese normalmente entre:

- Erros de especificación API presentes na declaración de métodos ou nos comentarios que afectan á sintaxe ou a semántica.
- Erros de código que se poden producir na implementación. Normalmente distribúense separados nun informe de erros. Sen embargo, pódense incluír en comentarios Javadoc utilizando a etiqueta *@bug*.

Agregar documentación Javadoc a NetBeans

Algunhas operacións relacionadas coa documentación Javadoc en NetBeans requiren que se agregue a documentación Javadoc a NetBeans. Isto faise en dous pasos:

- Descargar o arquivo de documentación Java SE.
- Agregar o arquivo descargado a NetBeans.

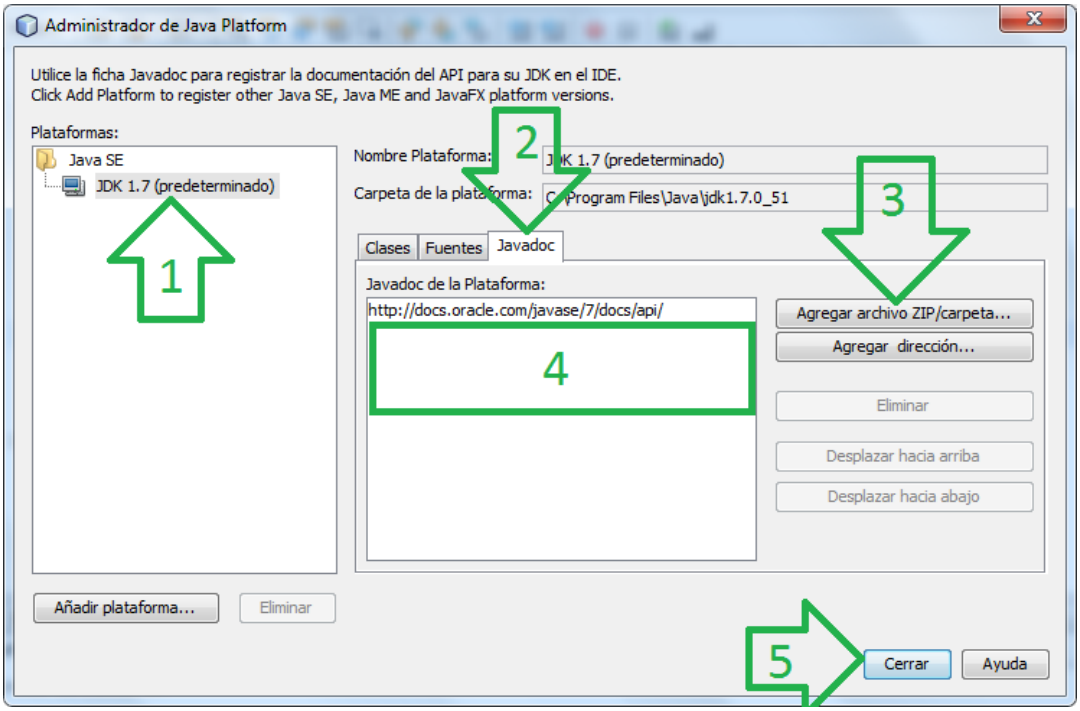
Descargar Java SE Documentation

Descárgase Java SE 7 Documentation (que non está incluida no JDK), da páxina web de Oracle http://www.oracle.com/technetwork/java/javase/downloads/index.html#docs. O arquivo *.zip* pódese gardar en calquera localización, por exemplo en *C:\Program Files\Java\jdk1.7.0_51\documentacion*.

Agregar Java SE Documentation a NetBeans

No menú principal, elíxese a opción Herramientas->Plataformas Java. Aparece a ventá Administrador de Java Platform na que se debe de:

- Paso 1. Seleccionar a plataforma a administrar.
- Paso 2. Elixir a pestana Javadoc,
- Paso 3. Premer no botón de Agregar arquivo zip.
- Paso 4. Aparecerá unha nova ventá na que poderá elixir o arquivo que se descargou no paso anterior e que aparecerá reflectido no apartado Javadoc de la plataforma:
- Paso 5. Premer no botón de Cerrar.



Comentarios Javadoc no código fonte

Os comentarios Javadoc serán aqueles comentarios Java (empezan por /* e finalizan por */) que serán utilizados por Javadoc para xerar a documentación dun proxecto.

Forma

Os comentarios Javadoc empezan por /** e finalizan con */ e poden incluír texto, código HTML e etiquetas Javadoc. A forma clásica de colocación destes comentarios e adaptada pola maioría dos programadores Java é a seguinte:

- Tódalas liñas de comentario teñen a mesma sangría que o código que documenta.
- A primeira liña do comentario só contén /**.
- As seguintes liñas empezan por espazo en branco, * e espazo en branco seguidos do texto do comentario.

- A última liña de comentario só contén espazo en branco e */.

Etiquetas

As etiquetas Javadoc (http://docs.oracle.com/javase/7/docs/technotes/tools/windows/javadoc.html#javadoctags) son expresións que empezan por @ e que se colocan dentro dos comentarios Javadoc. A primeira versión de Javadoc é Javadoc 1.0 e posteriormente foron xurdindo novas versión con novas etiquetas. As etiquetas máis usadas son as seguintes e colócanse normalmente seguindo a orde:

Etiqueta	Javadoc	Descrición
@author nome_autor	1.0	Só para clases e interfaces. Utilízase para indicar o nome ou nomes de programadores. Os nomes pódense separar por comas.
@version versión	1.0	Só para clases e interfaces. Utilízase para describir a versión. A descrición pode ser un texto explícito ou servirse de algún sistema de control de versión de código fonte (SCCS) como Subversion. Neste caso, pódense utilizar as cadeas "%I%" e/ou "%G%" para indicar o número de versión e/ou a data con formato mm/dd/yy.
@param nome_parametro descrición	1.0	Por convención deberíase de poñer sempre. Utilízase para describir os parámetros de cada método ou construtor.
@return descrición	1.0	Para describir o dato de retorno de cada método (non construtor). Non se usa con métodos void e por convención debería de poñerse sempre nos outros casos.
@throws nome_clase descrición	1.2 1.0	Utilízase para describir unha excepción lanzada por un método. Outra etiqueta sinónima é: @exception nome_clase_excepcion descrición
@see paquete.clase#membro texto	1.0	Utilízase para que no documento HTML xerado apareza un enlace na sección "See Also" (ou un texto con enlace) á documentación doutro paquete, clase, método ou campo. NetBeans guiará na descrición da referencia mediante axuda en liña. O membro pode ser un método ou un campo.
@since texto	1.1	Utilízase para indicar a versión do produto no que foi engadido o elemento.

Existen tamén as etiquetas en liña que se encerran entre { e } como por exemplo:

Etiqueta	Javadoc	Descrición
{@code texto}	1.5	Aparece na documentación HTML como <code>texto</code>
{@docRoot}	1.3	Representa a localización do directorio raíz do sitio HTML xerado. Esta etiqueta utilízase para incluír un arquivo, como unha páxina de copyright ou un logo.
{@inheritDoc}	1.4	Permite copiar documentación da clase máis próxima da que herda ou da interface implementada máis próxima ao sitio da etiqueta.
{@link paquete.clase#membro texto }	1.2	É igual que @see, pero xera o enlace en liña en lugar de colocalo na sección "See Also".
{@linkplain paquete.clase#membro texto}	1.4	Igual cá anterior pero xera o enlace en texto plano en lugar de dentro da etiqueta <code>. Neste caso, <b> non se interpretaría como etiqueta HTML.
{@literal texto}	1.5	Permite ver o texto de forma literal sen ser interpretado como texto HTML.
{@value [paquete.clase#constante]}	1.4	Mostra o valor dunha constante. Pódese utilizar sen nome da constante cando está dentro do comentario da constante.

Estilo

Normalmente, por convención, os comentarios seguen unhas regras de estilo:

- Usar a etiqueta HTML <code> para as palabras claves e os nome, é dicir, nomes de paquetes, clases, métodos, interfaces, campos, exemplos, palabras clave de Java.
- Restrinxir, dentro do posible, o uso de {@ link URL} xa que dificultan a lectura da documentación.
- Omitir parénteses cando se utiliza a forma xeral de métodos ou construtores e só utilizalos cando se quere facer referencia a unha específica forma.

- Utilizar breves descricións sobre todo no resumo inicial e nas descricións dos parámetros.
- Utilizar a terceira persoa dos tempos verbais. Por exemplo na descrición dun método: Obtén o valor da área do círculo en lugar de Obtense o valor da área do círculo.
- Empezar as descricións cun tempo verbal. Por exemplo na descrición dun método: Obtén o valor da área do círculo en lugar de Este método permite obter o valor da área do círculo.
- Nas descricións de clase, interface ou campo omitir o suxeito. Por exemplo: Etiqueta de botón en lugar de Este campo é unha etiqueta de botón.
- Usar este en lugar de o para referirse a un obxecto da presente clase. Por exemplo: Obtén o conxunto de ferramentas para este compoñente en lugar de Obtén o conxunto de ferramentas para o compoñente.
- Engadir na descrición algo máis co nome. Os mellores nomes son os que se documentan a si mesmos, pero na descrición debe haber máis información que a simple repetición da información que dá o nome. Por exemplo apra o método public void establecerX(int valorX) é mellor poñer:

```
/**
  * Establece el valor de la coordenada x.
  * Se admite cualquier valor entero
  * @param valorX valor de la coordenada x
  */
     x=valorX;
```

  que poñer establece X que só repetiría o nome do método.
- Utilizar a palabra campo para referirse a variables de clase e utilizar as palabras campo de fecha, campo numérico ou campo de texto para referirse aos obxectos correspondentes da clase TextField.
- Non utilizar abreviaturas non estándar na descrición. Por exemplo utilizar por exemplo en lugar de p.e.

Localización

Os comentarios Javadoc colócanse xusto antes da definición do elemento que comentan; campos, métodos, interfaces e clases.

Por exemplo, os comentarios de clase xusto antes da definición da clase.

```
1  package circulo;
2
3  /**
4   * Clase <b>Circulo</b> para pruebas en NetBeans
5   * @author profesor
6   * @version 1.0
7   */
8  public class Circulo {
```

Por exemplo, os comentarios de campos xusto antes da declaración de cada campo.

```
9       /**
10       * coordenada x
11       */
12      private int x;
13      /**
14       * coordenada y
15       */
16      private int y;
```

```
22      /**
23       * Constructor para la clase Circulo que asigna los valores de las
24       * coordenadas x, y y el valor del radio
25       * @param valorX valor de la coordenada x
26       * @param valorY valor de la coordenada y
27       * @param valorRadio valor del radio
28       */
29      public Circulo(int valorX, int valorY, double valorRadio) {
30          establecerX(valorX);
31          establecerY(valorY);
32          establecerRadio(valorRadio);
33      }
```

Por exemplo, os comentarios de método e interfaces xusto antes da declaración da firma.

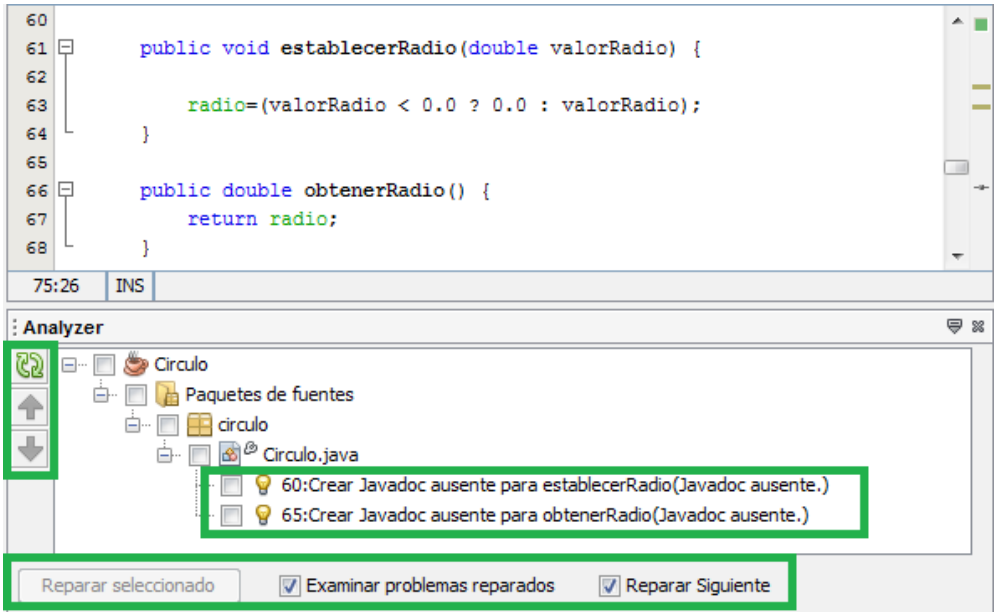
Analizar Javadoc estaticamente

NetBeans permite analizar estaticamente o código fonte e emitir informe sobre como mellorar a documentación Javadoc. A análise pódese facer de dúas maneiras:

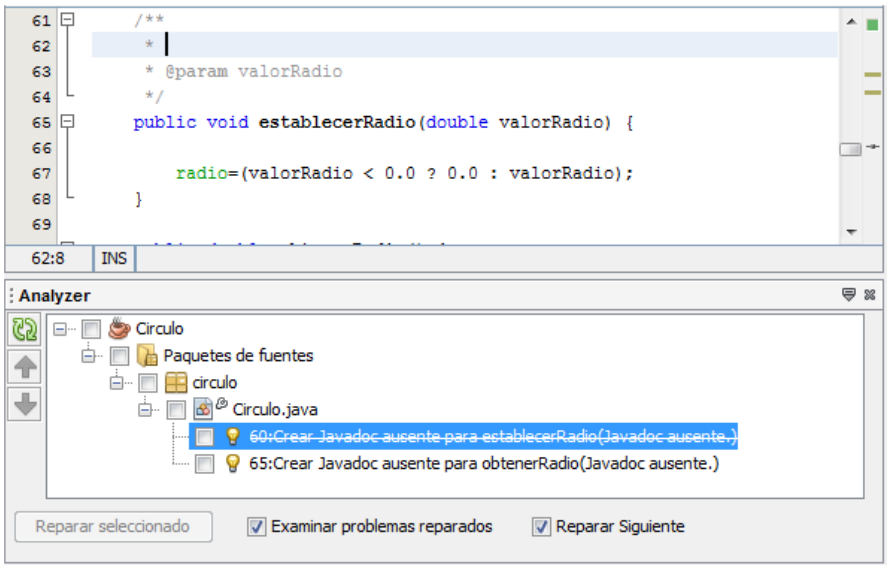
- Na ventá de proxecto, seleccionar o arquivo a analizar, facer clic dereito e elixir *Herramientas->Analizar Javadoc* ou
- No menú principal e co cursor na ventá de edición do código fonte a analizar, elixir *Herramientas-> Analizar Javadoc*.

En calquera dos dous casos anteriores, ábrese unha pestana Analyzer co informe dos problemas de documentación encontrados e na que se pode ver:

- Unha barra de ferramentas á esquerda que permiten actualizar o informe sobre a análise, ir ó anterior problema ou ir ó seguinte problema.
- A lista de problemas encontrados coa liña na que se encontrou o problema e unha breve descrición do mesmo. Os problemas poden ser seleccionados para ser reparados.
- Unha parte inferior na que se poden reparar os problemas marcados, que permita volver a examinar problemas reparados e na que se pode seleccionar a reparación do seguinte problema.

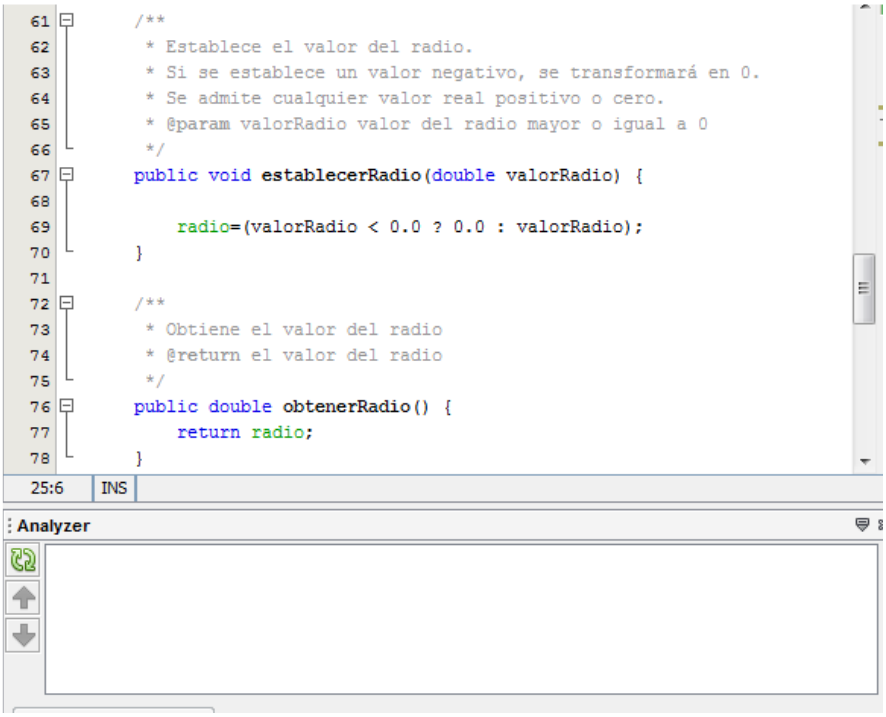


De marcar algún dos problemas e premer en Reparar seleccionado, aparece inserido no código a estrutura básica docomentario Javadoc, incluíndo etiquetas Javadoc cando sexa posible. Por exemplo pode aparecer @param se é un método con parámetros, ou @return cando é un método diferente de void ou non aparecer ningún código e só aparecer a estrutura de comentario cando é un método void que non ten parámetros. Despois de reparalo, verase na ventá Analyzer desmarcado e coa descrición tachada.



A operación de Reparar seleccionado realiza a primeira parte da reparación do problema pero é o programador quen deberá de revisar e completar eses comentarios e para iso conta coa axuda en liña de NetBeans que permite visualizar a lista de posibles códigos Javadoc axeitados cando se teclea @ dentro da estrutura de comentario Javadoc.

Despois de solucionar tódolos problemas de documentación Javadoc que detecta NetBeans, e despois de actualizar a ventá Analyzer, esta debe aparecer baleira.



Xerar documentación Javadoc

NetBeans permite crear automaticamente documentación Javadoc para un proxecto, é dicir, xera un conxunto de páxinas HTML que describan as clases, interfaces, construtores, métodos e campos dun proxecto, a partir do código fonte e dos comentarios Javadoc embebidos no código.

Os pasos a seguir son:

- Seleccionar o proxecto na ventá de proxectos.
- Xerar Javadoc. Pódese facer de calquera das dúas maneiras seguintes:
  - Ejecutar->Generate Javadoc(Circulo) do menú principal ou
  - Facer clic dereito e elixir Generar Javadoc.
- A documentación Javadoc xerada verase en primeira instancia no navegador externo pero tamén se poderá ver no IDE dende os arquivos fonte e utilizando a ventá Javadoc, ou no IDE utilizando o índice de busca de Javadoc.

Javadoc ante os seguintes casos particulares:

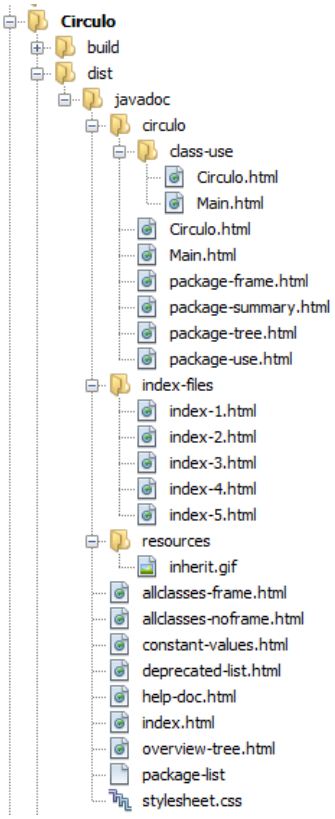
- un método nunha clase sobreescribe un método dunha superclase,
- un método nunha interface sobreescribe un método nunha superinterface ou
- un método nunha clase implementa un método dunha interface,

que resolve:

- por defecto xerando un Overrides na documentación para o método e cun enlace ao método que sobreescribe nos dous primeiros casos,
- xerando un Specified by na documentación cun enlace ao método que se implementa, no terceiro caso.
- en calquera dos tres casos, o método pode ter comentarios Javadoc escritos polo programador, que tamén aparecerán na documentación xerada.

Localización

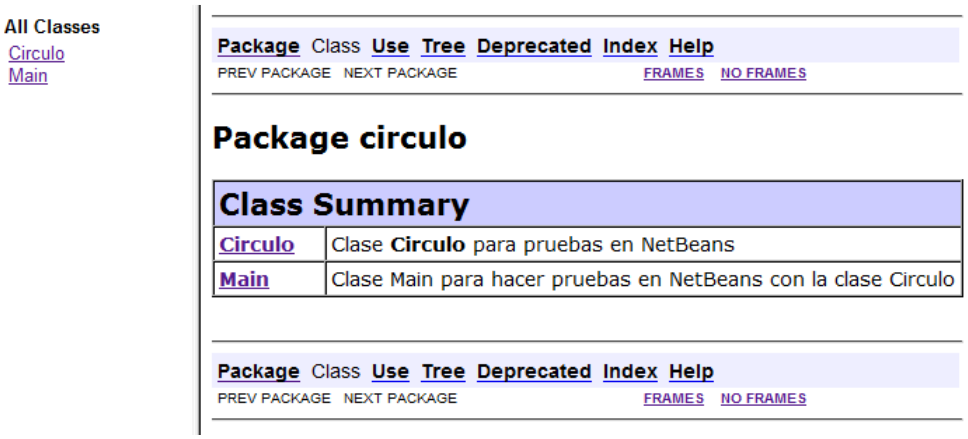
NetBeans xera as páxinas web necesarias, almacénaas dentro do proxecto na carpeta *dist/Javadoc* e lanza index.html no navegador designado.



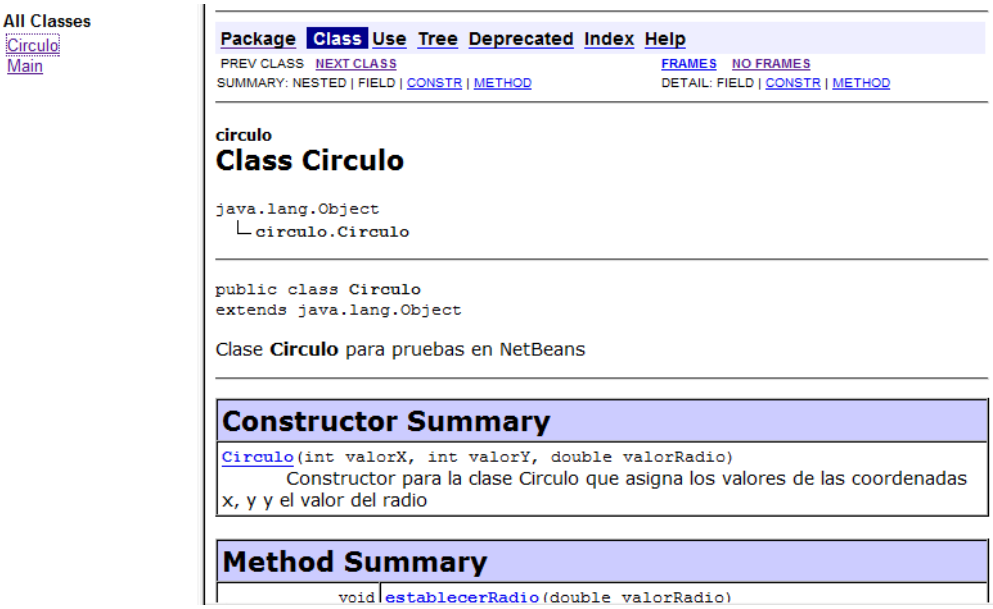
Forma

As páxinas web xeradas teñen a mesma forma cás da especificacións API de Java. Son páxinas HTML 4.01 Frameset que por defecto son vistas utilizando frames aínda que isto pode cambiarse. De utilizar frames, as páxinas estarán divididas en tres ou dous frames dependendo de se hai máis dun paquete. Os tres frames son: frame de paquetes, frame de clases (debaixo do de paquetes), frame de detalle (á dereita dos dous frames anteriores). Por defecto o frame de detalle aparece cunha barra de navegación na parte superior. O menú Help describe a forma das páxinas (en inglés).

No exemplo actual só hai un paquete, polo que as páxinas xeradas por Javadoc non terán o frame de paquetes. Na páxina principal descríbese o paquete único circulo. Os comentarios a carón de cada clase extráense dos comentarios Javadoc de cada clase. Observe que que a palabra Circulo aparece en bold ou grosa tal e como se colocou no comentario Javadoc correspondente e que non aparecen referencias ao autor nin a versión nin aparecen os campos de tipo private.



De seleccionar a clase Circulo no frame de clases, aparecería no frame de detalle información sobre esa clase en varios apartados: descrición xeral, resumo de métodos e campos e detalle de cada un deles.



All Classes
Circulo
Main

Package Class Use Tree Deprecated Index Help
PREV CLASS NEXT CLASS FRAMES NO FRAMES
SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

circulo
Class Circulo

java.lang.Object
  └ circulo.Circulo

public class Circulo
extends java.lang.Object

Clase **Circulo** para pruebas en NetBeans

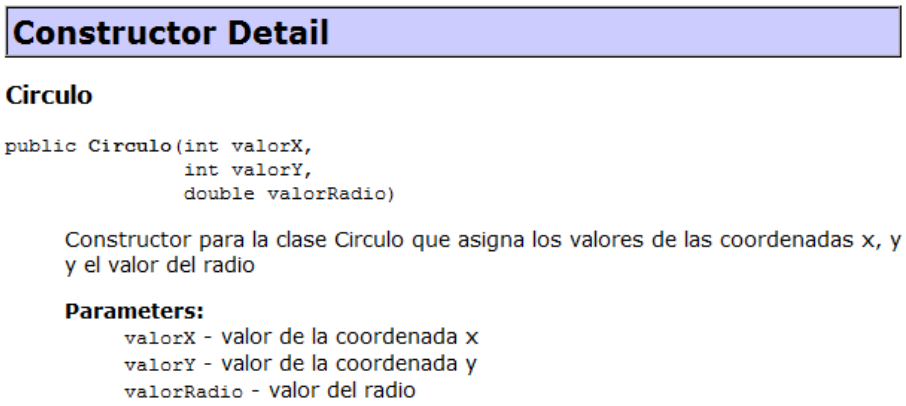
Constructor Summary

Circulo(int valorX, int valorY, double valorRadio)
Constructor para la clase Circulo que asigna los valores de las coordenadas x, y y el valor del radio

Method Summary

void establecerRadio(double valorRadio)

Nos detalles do método construtor aparece:



Constructor Detail

Circulo

```
public Circulo(int valorX,
               int valorY,
               double valorRadio)
```

Constructor para la clase Circulo que asigna los valores de las coordenadas x, y y el valor del radio

Parameters:

`valorX` - valor de la coordenada x
`valorY` - valor de la coordenada y
`valorRadio` - valor del radio

que se extrae do código:

```java
/**
 * Constructor para la clase Circulo que asigna los valores de las
 * coordenadas x, y y el valor del radio
 * @param valorX valor de la coordenada x
 * @param valorY valor de la coordenada y
 * @param valorRadio valor del radio
 */
public Circulo(int valorX, int valorY, double valorRadio) {
    establecerX(valorX);
    establecerY(valorY);
    establecerRadio(valorRadio);
}
```

Na clase Main pode aparecer un enlace á documentación da clase Circulo no apartado See Also

circulo

Class Main

```
java.lang.Object
  └ circulo.Main
```

```
public class Main
extends java.lang.Object
```

Clase Main para hacer pruebas en NetBeans con la clase Circulo

See Also:

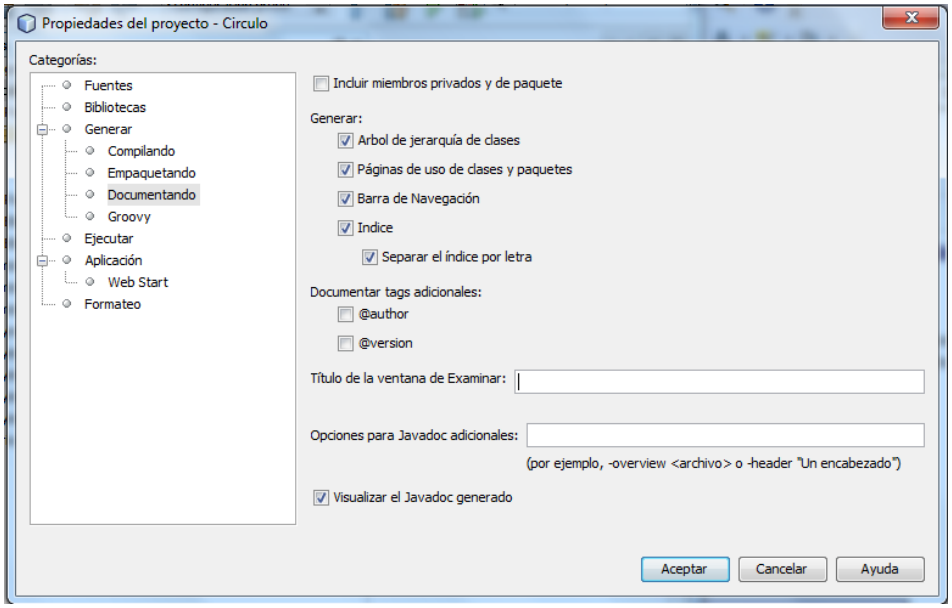
Circulo

que se extrae do código

```
1   package circulo;
2
3   import java.text.DecimalFormat;
4
5   /**
6    * Clase Main para hacer pruebas en NetBeans con la clase Circulo
7    * @author profesor
8    * @version 1.0
9    * @see circulo.Circulo
10   */
11  public class Main {
12      public static void main(String[] args) {
```

Configurar a xeración de documentación Javadoc para un proxecto

Para cambiar a documentación Javadoc xerada para un proxecto, débese seleccionar o proxecto na ventá de proxectos, facer clic dereito e elixir Propiedades. Aparece a caixa de diálogo das propiedades do proxecto na que se selecciona Generar->Documentando e pódense modificar os valores por defecto.



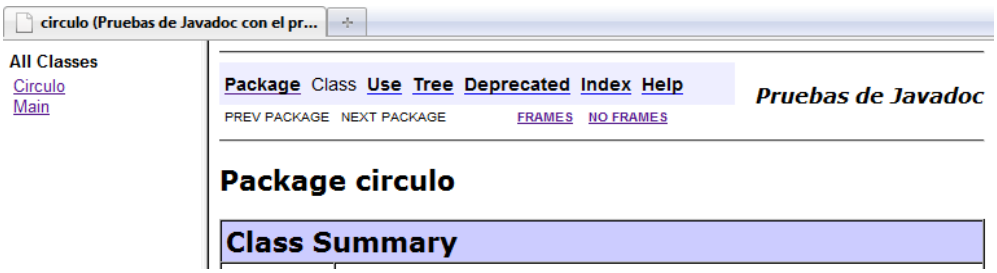
Pódese marcar Incluir miembros privados y de paquete para que nos arquivos xerados aparezan os campos que son privados e pódense marcar os tags @author e @version, para que nos arquivos xerados apareza a documentación relativa a esas etiquetas Javadoc. Neste caso aparecerá incluída a seguinte documentación:

Version:
1.0
Author:
profesor

Field Summary	
private double	radio radio
private int	x coordenada x
private int	y coordenada y

Pódese poñer un título para que apareza como contido da etiqueta <title> das páxinas HTML. En Opciones para Javadoc adicionales poderíanse poñer opcións de Javadoc (ver opcións completas en http://docs.oracle.com/javase/1.5.0/docs/tooldocs/windows/Javadoc.html). Por exemplo para que á dereita da barra de navegación apareza o texto Pruebas de Javadoc en letra grosa deberíase de escribir:

```
-header "<b>Pruebas de Javadoc</b>"
```



Ver documentación Javadoc dende o arquivo fonte

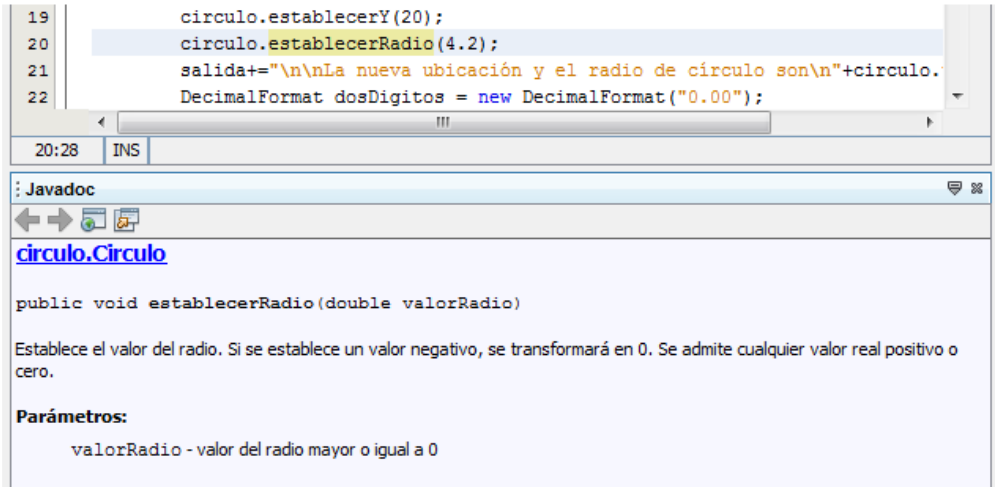
Para ver a documentación Javadoc dun elemento dende o código fonte, pódese facer de dúas maneiras.

- Vendo o resultado no navegador externo:

  Colocar o cursor sobre o elemento do código fonte do que se quere ver a documentación e premer Alt+F1 ou tamén facer clic dereito e elixir Mostra Javadoc. Se non funciona este forma é porque se necesita agregar Javadoc a NetBeans.

- Vendo o resultado na ventá de Javadoc (Ventana->Other->Javadoc):

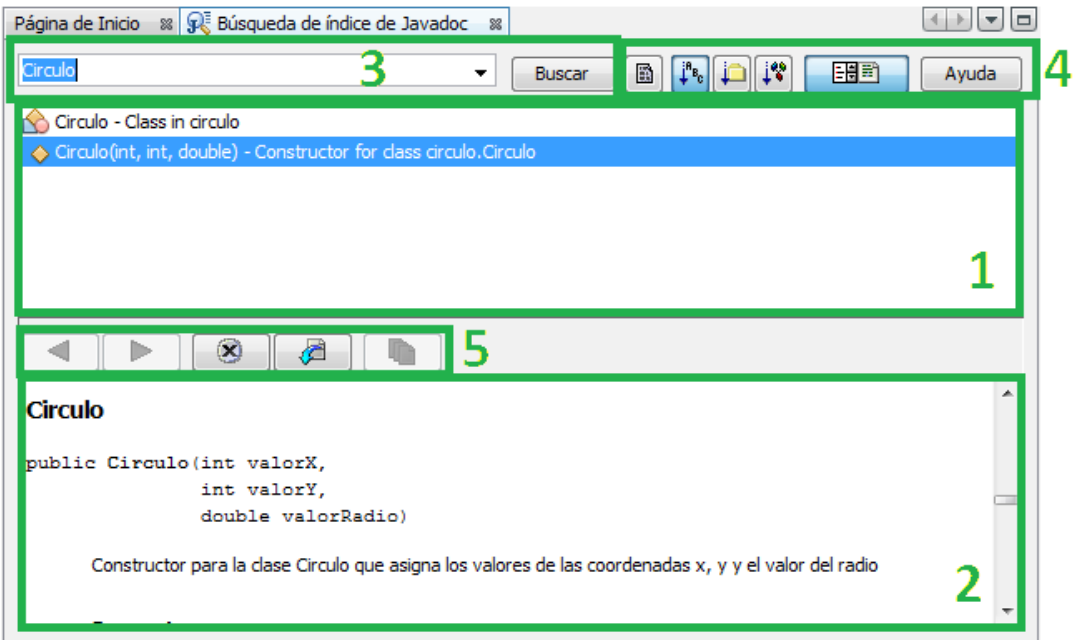
  Colocar o cursor sobre o elemento do código fonte do que se quere ver a documentación e aparecerá na ventá de Javadoc a documentación do elemento.



Ver documentación Javadoc dende o índice de busca de Javadoc

NetBeans permite facer buscas rápidas e fáciles na documentación Javadoc xerada sen saír de NetBeans. Para iso necesítase elixir Ayuda->Búsqueda de índice de Javadoc (Mayusculas-F1) no menú principal. Aparece a pestana Búsqueda de índice de Javadoc.

Se previamente o cursor estivese sobre un elemento do código fonte do que se quere ver a documentación, a pestana Búsqueda de índice de Javadoc aparecería cos resultados da busca dese elemento.



Na que se distinguen as seguintes zonas:

- Zona de aparición dos resultados da busca. Pulsando dobre clic sobre un dos resultados, aparece a documentación no navegador.
- Visor HTML. Anaco de páxina HTML de documentación relativa ó resultado resaltado na zona 1.
- Permite teclear un texto ou seleccionar entre as buscas anteriores e premer en Buscar para facer unha nova busca.
- Utilizar a barra de ferramentas para:
  - Mostrar orixe do resultado resaltado na zona 1 no código fonte. Ábrese a pestana có código fonte.
  - Ordenar os resultados alfabeticamente.

  - Ordenar os resultados alfabeticamente polo nome do paquete.
  - Ordenar os resultados alfabeticamente por entidade (clase, interface, método, construtor ou campo)
  - Activar ou desactivar o visor HTML.
  - Axuda.
- Barra de ferramentas para moverse dentro do visor HTML:
  - Atrás. Retorna á páxina previa.
  - Vai á páxina visitada antes de premer en Atrás.
  - Para a carga da páxina.
  - Recargar a páxina actual.
  - Visualiza o historial das páxinas.

Etiquetas e anotacións

As etiquetas Javadoc non se deben de confundir coas anotacións Java tamén chamados metadatos Java.

Semellanzas:

- colócanse xusto antes da clase, método ou campo á que afectan,
- empezan por @,
- poden ter os mesmos nomes aínda que nas anotación empezarán por maiúsculas e nas etiquetas en minúsculas,
- poden utilizarse para describir as mesmas situacións,
- no código fonte aparecerá o nome do elemento desaprobado tachado.

Diferencias:

- as etiquetas Javadoc van dentro de comentarios Javadoc e as anotación xusto antes do elemento seguidos dun espazo ou unha nova liña e fóra de comentarios,
- as etiquetas Javadoc serven para xerar documentación e as anotacións dan información aos analizadores e compiladores,
- as anotacións Java aportan información a Javadoc para xerar documentación,
- poden complementarse

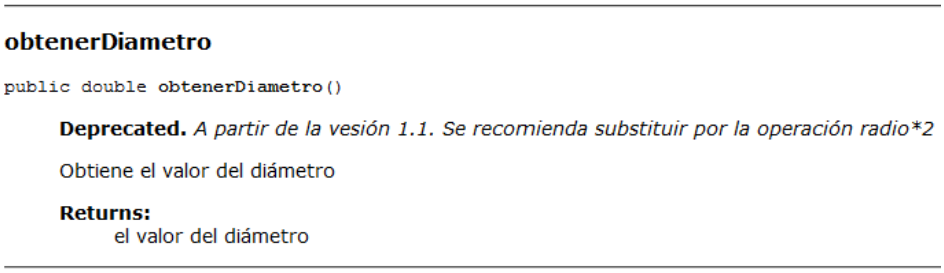
Exemplo @deprecated

```java
/**
 * Obtiene el valor del diámetro
 * @return el valor del diámetro
 * @deprecated A partir de la vesión 1.1.
 * Se recomienda substituir por la operación radio*2
 */
public double obtenerDiametro() {
    return radio * 2;
}

/**
 * Obtiene el valor de la circunferencia
 * @return el valor de la circunferencia
 */
public double obtenerCircunferencia() {
    return Math.PI * obtenerDiametro();
}
```

No código fonte aparece tachado o método e as referencias ao método:

Na documentación HTML xerada, o método obtenerDiametro() sae como elemento desaprobado, en itálica e precedido por un warning en letra grosa.



obtenerDiametro

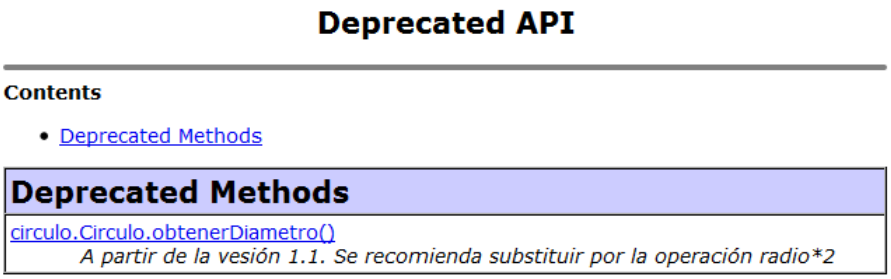
```
public double obtenerDiametro()
```

Deprecated. *A partir de la vesión 1.1. Se recomienda substituir por la operación radio*2*

Obtiene el valor del diámetro

Returns:
el valor del diámetro

Na documentación HTML dos elementos desaprobados aparece o método obtenerDiametro().



Deprecated API

Contents

- Deprecated Methods

Deprecated Methods

circulo.Circulo.obtenerDiametro()
*A partir de la vesión 1.1. Se recomienda substituir por la operación radio*2*

Exemplo @Deprecated

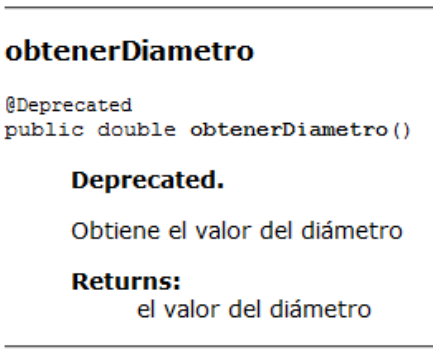
A anotación @Deprecated provoca que os compiladores (polo menos os de Sun) emitan unha mensaxe warning cando se utiliza o método indicando que o método está declarado como desaprobado e ademais xera na documentación un elemento desaprobado.

No código fonte:

```
/**
 * Obtiene el valor del diámetro
 * @return el valor del diámetro
 */
@Deprecated
    public double obtenerDiametro() {
    return radio * 2;
}

/**
 * Obtiene el valor de la circunferencia
 * @return el valor de la circunferencia
 */
public double obtenerCircunferencia() {
    return Math.PI * obtenerDiametro();
}
```

Na documentación da clase Circulo:



obtenerDiametro

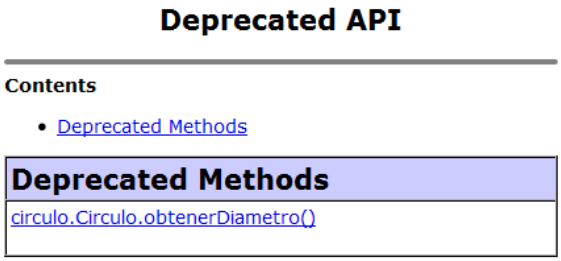
```
@Deprecated
public double obtenerDiametro()
```

Deprecated.

Obtiene el valor del diámetro

Returns:
el valor del diámetro

Na documentación dos elementos desaprobados:



Deprecated API

Contents

- Deprecated Methods

Deprecated Methods
circulo.Circulo.obtenerDiametro()

Exemplos

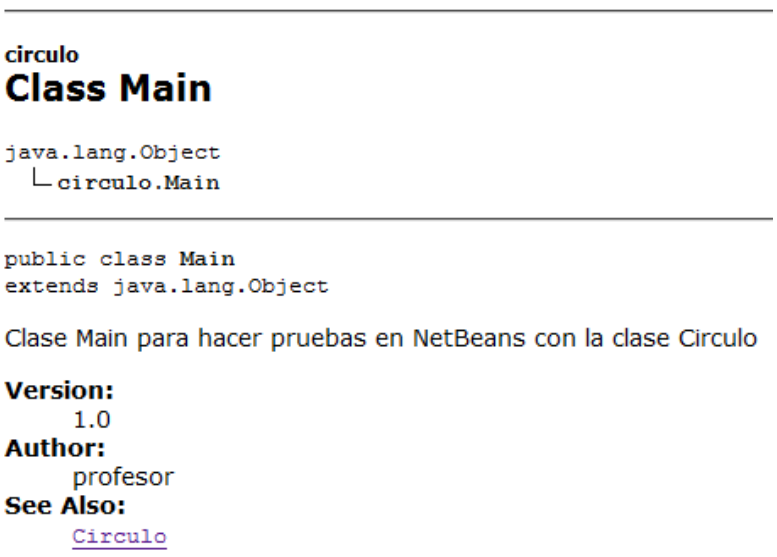
@see

Exemplo de colocación de enlace á documentación da clase Circulo na clase Main.java do proxecto Circulo utilizando @see circulo.Circulo

```java
package circulo;

import java.text.DecimalFormat;

/**
 * Clase Main para hacer pruebas en NetBeans con la clase Circulo
 * @author profesor
 * @version 1.0
 * @see circulo.Circulo
 */
public class Main {
```



circulo

Class Main

```
java.lang.Object
  └ circulo.Main
```

```
public class Main
extends java.lang.Object
```

Clase Main para hacer pruebas en NetBeans con la clase Circulo

Version:
1.0
Author:
profesor
See Also:
Circulo

{@link}

Exemplo de colocación de enlace á documentación da clase Circulo na clase Main.java do proxecto Circulo utilizando {@link circulo.Circulo}:

```
package circulo;

import java.text.DecimalFormat;

/**
 * Clase Main para hacer pruebas en NetBeans con la
 * clase {@link circulo.Circulo círculo}
 * @author profesor
 * @version 1.0
 */
public class Main {
```

---

circulo

Class Main

```
java.lang.Object
  └ circulo.Main
```

```
public class Main
extends java.lang.Object
```

Clase Main para hacer pruebas en NetBeans con la clase círculo

Version:
1.0
Author:
profesor

@docRoot

Exemplo de colocación dun logo na cabeceira da documentación:

- Na carpeta raíz do proxecto circulo, Circulo\dist\javadoc, crear a carpeta imaxes co logo lagarto.jpg.
- Nas propiedades do proxecto circulo, na categoría documentando, colocar nas opcións adicionais para Javadoc:

  -header ""

Package **Class** **Use** **Tree** **Deprecated** **Index** **Help**

PREV CLASS NEXT CLASS FRAMES NO FRAMES

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

Exemplos: http://www.oracle.com/technetwork/java/javase/documentation/index-137868.html#examples

{@value}

Exemplo de visualización do valor da constante PI de java.lang.Math:

```java
/**
 * Obtiene el valor de la circunferencia
 * y utiliza el valor de PI={@value java.lang.Math#PI}
 * @return el valor de la circunferencia
 */
public double obtenerCircunferencia() {
    return Math.PI * obtenerDiametro();
}
```

---

obtenerCircunferencia

```
public double obtenerCircunferencia()
```

Obtiene el valor de la circunferencia y utiliza el valor de PI=3.141592653589793

Returns:

el valor de la circunferencia
