

Javadoc

HTML dokumentációkészítő program

Írta: Angster Erzsébet (2001, 2003)

1. Dokumentációs megjegyzés

A dokumentációs megjegyzés a Java programban a következő:

```
/** Ez a Javadoc számára készült megjegyzés */
```

A javadoc megjegyzésnek csak a következő helyeken van hatása:

- ◆ Osztály és interfész definíciók előtt;
- ◆ Tag metódusok és konstruktorok előtt;
- ◆ Metódusok lokális változói előtt.

A javadoc megjegyzés más helyeken egyszerű Java megjegyzésként viselkedik.

A javadoc a forráskódból html dokumentációkat készít, a neki szóló megjegyzéseket figyelembe véve. A javadoc program a JDK része, a bin könyvtárban található.

A JavaDoc a JBuilder 8.0 Enterprise környezetből is meghívható.


Példaként nézzük a Mikro projekt forráskódját, és a javadoc által generált index.html állományt (1. ábra)!

```
/**  
 * Ezt a megjegyzést a javadoc figyelmen kívül hagyja.  
 */
```

```
import java.awt.*;  
import java.awt.event.*;  
import javax.swing.*;  
import javax.swing.border.*;
```

```
/**  
 * A mikrohullámú sütő főablaka.  
 * Az osztály állapotának meghatározói:  
 * {@link Ajto}, {@link Motor}  
 */
```

Javadoc megjegyzés



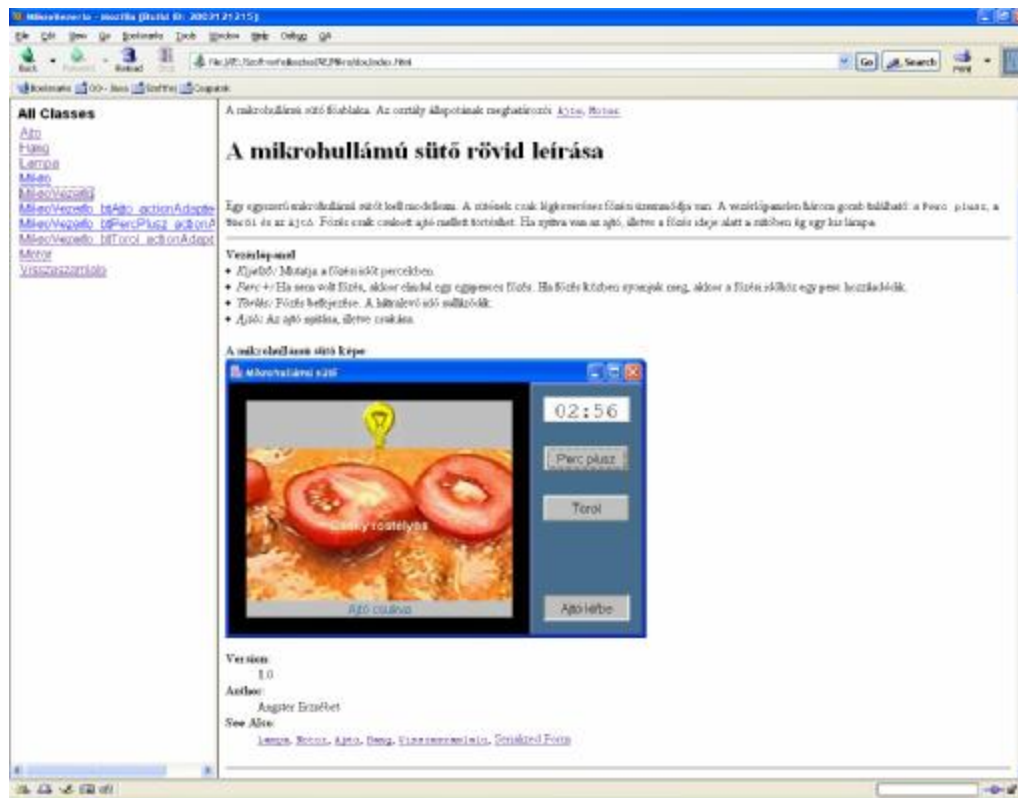
```

* <p><h1>A mikrohullámú sütő rövid leírása</h1><br>
* Egy egyszerű mikrohullámú sütőt kell modellezni.
* A sütőnek csak légkeveréses főzési üzemmódja van.
* A vezérlőpanelen három gomb található: a
* <code>Perc plusz</code>, a <code>Töröl</code> és az
* <code>Ajtó</code>.
* Főzés csak csukott ajtó mellett történhet. Ha nyitva van az
* ajtó, illetve a főzés ideje alatt a sütőben ég egy kis
* lámpa.<br>
* <hr>
* <b>Vezérlőpanel</b>
* <li><i>Kijelző:</i> Mutatja a főzési időt percekben.
* <li><i>Perc+:</i> Ha nem volt főzés, akkor elindul egy
* egyperces főzés. Ha főzés közben nyomjuk meg, akkor a főzési
* időhöz egy perc hozzáadódik.</li>
* <li><i>Töröl:</i> Főzés befejezése. A hátralevő idő
* nullázódik.</li>
* <li><i>Ajtó:</i> Az ajtó nyitása, illetve csukása.</li>
* <br>
* <b>A mikrohullámú sütő képe:</b><br>
* 
*
* @author Angster Erzsébet
* @version 1.0
* @see Lampa
* @see Motor
* @see Ajto
* @see Hang
* @see Viasszaszamlalo
*/

public class MikroVezerlo extends JFrame {
    /** Kapcsolatok.*/
    private Hang rovidsip = new Hang("sound/rovidsip.wav");
    private Hang hosszusip3 = new Hang("sound/hosszusip3.wav");
    ...

    /** Értesítés jött a viasszaszámlálótól, hogy letelt
     * egy másodperc.
     */
    public void mpLetelt() {
        kijelez();
    }
}

```



1. ábra. Az index.html állomány a böngészőben

2. Javadoc megjegyzés

Ajánlatos minden osztály és interfész forráskódját külön állományban tárolni, mert a privát osztályok csak akkor jelennek meg, ha az összes privát deklarációt megjelenítjük (lásd dokumentációkészítési opciók).

Szabályok:

- ◆ A megjegyzés szöveget a `/**` vezeti be, és a `*/` zárja le.
- ◆ A sorkezdő fehér szóközök és az azt követő tetszőleges számú `*` karakter figyelmen kívül marad.
- ◆ Az első mondat (az első pontig) összesítő szöveg, amely csak a tömör leírásban fog szerepelni.
- ◆ A szöveg `html` parancsokkal formázható, grafikák, képek illeszthetők be (néhány parancs kerülendő, mint például a header megadása).

HTML parancsok:

<code><i>...</i></code>	italic	dőlt
<code>...</code>	bold	vastag
<code><u>...</u></code>	underlined	aláhúzott
<code><code>...</code></code>	code style (courier)	kód stílus
<code><p>...</p></code>	paragraph	paragrafus
<code><hr></code>	horizontal line	vízszintes vonal beillesztése
<code></code>	image	Kép beillesztése – src: URL – alt: ezt írja ki, ha nem képes a képet megjeleníteni – align: top/middle/bottom/texttop

A megjegyzés-szöveg utolsó soraiként toldalékok adhatók meg. Egy toldalékot a @ jel vezet be. A toldalékok közé normál szöveg már nem iktatható be. Egy féle toldalékból több is megadható. A toldalékok szöveg végi szekciókat generálnak, mint például:

```
Athor:
Version:
See also:
Returns:
Throws:
Parameters:
Since:
Depricated:
```

Toldalékok:

@<toldaléknév> <szöveg>

Tagnév	Mi előtt használható?	Lerírás
@author	Osztály, Interfész	Szerző megadása.
@deprecated	Osztály, Interfész, Metódus, Lokális változó	A deklaráció elavult.
@exception	Metódus	A metódus által dobott kivételek megadása.
{@link}	Osztály, Interfész, Metódus, Lokális változó	Létrehoz egy inline hiperkapcsolatot.
@param	Metódus	Megadja a metódus formális paraméterét.
@return	Metódus	
@see	Osztály, Interfész, Metódus, Lokális változó	Készít egy hiperlink hivatkozást (lásd később)

@since	Osztály, Interfész, Metódus, Lokális változó	Megadja a kód első olyan verziószámát, amelyben ez a deklaráció már létezett.
@throws	Metódus	mint a @exception toldalék.
@version	Osztály, Interfész	

3. Hiperlinkek – @see és {@link} toldalékok

Osztályokhoz, interfészekhez

```
@see <name> <label>
{@link <name> <label>}
```

- ◆ @see a **See also** szekcióba kerül bele. A {@link} a szövegben levő inline hivatkozás.
- ◆ name: teljes vagy részleges azonosító
- ◆ label: A hivatkozás helyén megjelenített szöveg. Ha nem adjuk meg, akkor az azonosító jelenik meg.

Tagokhoz

```
@see <name>#method <label>
{@link <name>#method <label>}
```

- ◆ @see a *See also* szekcióba kerül bele. A {@link} a szövegben levő inline hivatkozás.
- ◆ name: teljes vagy részleges osztályazonosító
- ◆ label: A hivatkozás helyén megjelenített szöveg. Ha nem adjuk meg, akkor az azonosító jelenik meg.

Például az insert metódus megjegyzésében:

```
@deprecated Version 1.1 replaced by the {@link #push(Object)
<i>push</i>} method. The insert name was misleading.
@see #push(Object)
```

A stylesheet.css állomány tartalmazza a html állomány stílusát (betűk, színek stb.), amely szöveg formátumú és szerkeszthető.

4. Dokumentációkészítés, opciók

A javadoc direkt futtatása:

```
javadoc <opciók>
```

Opciók

<code>-sourcepath <path></code>	A forrásállományokat tartalmazó könyvtárak és állományok listája. Szeparátor: ; (Windows) : (Unix)
<code>-classpath <path></code>	Az előző lista folytatása.
<code>-d <directory></code>	Célkönyvtár, ez lesz a html dokumentumok gyökérkönyvtára, ide kerül az <code>index.html</code> . A célkönyvtárnak léteznie kell. Alapért: aktuális könyvtár.
<code>-public</code>	Csak <code>public</code> deklarációkra készül dokumentáció.
<code>-protected</code>	Csak <code>public/protected</code> deklarációkra készül dokumentáció. Ez az alapértelmezés.
<code>-package</code>	Csak a <code>package/public/protected</code> deklarációkra készül dokumentáció.
<code>-private</code>	Bármely láthatóságú deklarációra készül dokumentáció.
<code>-author</code>	A dokumentációban a <code>@author</code> paragrafusnak akkor van hatása, ha ez az opció itt is szerepel.
<code>-version</code>	A dokumentációban a <code>@version</code> paragrafusnak akkor van hatása, ha ez az opció itt is szerepel.
<code>-nodeprecated</code>	A dokumentáció <code>@deprecated</code> paragrafusai nem kerülnek bele a html dokumentációba.

Példa

Készítsünk egy forráskódból javadoc dokumentációt úgy, hogy az tartalmazzon minden deklarációt, a megfelelő helyeken szerepeljen a verziószám, de a szerző ne legyen feltüntetve! A forráskód az aktuális könyvtárban van, a dokumentációt a `f:\javadoc` könyvtárba kérjük:

```
javadoc Forras.java -d f:\javadoc -private -version
```

A Mikro program dokumentációját a `makeJavadoc.bat` állítja elő, tartalma a következő:

```
%JAVA_HOME%\bin\javadoc src\*.java -d doc\javadoc -author  
-version
```

Az `src` könyvtárban található összes `java` fájlból készít dokumentációt. Az eredményt az `doc\javadoc` könyvtárba teszi. A dokumentációban megjelenik a szerző- és verzióinformáció.

A Javadoc a JBuilder 8.0 Enterprise környezetből is meghívható.