|  |  |
| --- | --- |
| |  | | --- | | [**JavaDoc 주석 다는 방법, 기본적인 예**](http://www.gisdeveloper.co.kr/562) | |
|  |
| 여전이 주석이라는 것에 대해 긍정적인 면보다는 부정적인 면이 더 많다고 생각하고 있지만.. 소스 코드와 주석을 통해 멋지게 문서를 자동으로 생성해 주는 개념은 주석에 대한 부정적인 많은 부분을 불식시키고도 남는듯합니다.   Java로 만든 어플리케이션(또는 서비스)의 소스를 협업자 또는 제3자에게 제공하기 위해 주석을 달때 JavaDoc의 도움을 받아 좀더 체계적으로 작업할 수 있도록 하기 위한 주석법입니다. 간단히 제 스스로 개발할때 참고하여 사용할만한 예제 코드로 정리해 봅니다.  먼저 클래스에 대한 주석의 예입니다.  [view plaincopy to clipboardprint?](http://www.gisdeveloper.co.kr/562)   1. /\*\* 2. \* 화면상에 윈도우를 나타내기 위한 클래스 3. \* 사용 방법: 4. \* <pre> 5. \*    Window win = new Window(parent); 6. \*    win.show(); 7. \* </pre> 8. \* 9. \* @author  Sami Shaio 10. \* @version 1.0 11. \* @see     java.awt.BaseWindow 12. \* @see     java.awt.Button 13. \*/ 14. **class** Window extends BaseWindow { 15. ... 16. }   다음은 클래스의 필드에 대한 주석의 예입니다.  [view plaincopy to clipboardprint?](http://www.gisdeveloper.co.kr/562)   1. /\*\* 2. \* 컴포넌트의 X 좌표 3. \* 4. \* @see #getLocation() 5. \*/ 6. **int** x = 1263732;   끝으로 클래스의 매서드에 대한 주석의 예입니다.  [view plaincopy to clipboardprint?](http://www.gisdeveloper.co.kr/562)   1. /\*\* 2. \* 지정된 index에 위치하는 문자 반환. 3. \* index의 범위는 <code>0</code>에서 <code>length() - 1</code>까지임. 4. \* 5. \* @param     index  얻고자 하는 문자의 인덱스 값 6. \* @return      지정된 index의 문자 7. \* @exception StringIndexOutOfRangeException 8. \*     index가 <code>0</code>에서 <code>length()-1</code>의 범위를 벗어남 9. \* @see       java.lang.Character#charValue() 10. \*/ 11. **public** **char** charAt(**int** index) { 12. ... 13. } |

자바에서 주석 처리는 3가지 형태가 있다.  
// 와 /\* \*/ 그리고 문서화 주석이라 하는 것  
  
문서화 주석 는 /\*\* 로 시작해서 \*/ 로 끝나는데 이문서화 주석를  
사용하게 되면 javadoc 라는 유틸을 사용해 HTML 파일로 저장이 가능해 진다  
  
즉 프로그램의 문서화를 편하게 할수 있는 것이다.

javadoc 에서 인식하는 태그는 아래와 같다.  
  
@author 계발자  
@deprecated 클래스나 멤버가 사장되었는지 지정  
{@docRoot} 현 문서의 루트 디렉토리 경로 지정  
@exception 메소드에서의 예외 확인  
{@inhreitDoc} 상위 클래스로부터 주석을 상속  
{@link} 다른 주제에 관한 인라인 링크 삽입  
{@linkplain} 다른 주제에 관한 인라인 링크 삽입 - 링크가 평문 텍스트로 표시  
@param 메소드의 매개변수 문서화  
@return 메소드의 반환값 문서화  
@see 다른 주제에 관한 링크 지정  
@serial 기본 직렬화 필드 문서화  
@serialData writeObject() 나 writeExteral() 메소드로 기록한 데이타 문서화   
@serialField ObjectStreamField 컴포넌트 문서화  
@since 특별한 변경 사항이 소개될때 릴리즈 기록  
@throws 메소드에서의 예외 확인  
{@value} static 필드여야 하는 상수값 표시  
@version 클래스의 버전  
  
예)  
  
/\*\*  
  
\* 이 클래스는 .... 이다  
\* @author 아무개  
\* @version 1.1.2  
  
\*/

**@serialField** : ObjectStreamField 객체를 설명하는데 사용  
**@serial** : 기본적으로 직렬화할 수 있는 클래스의 데이터 멤버를 설명하는데 사용  
**@since** : 어떠한 특성이 코드에 소개된 시점을 가리키는데 사용한다.  
[**{@link**](mailto:%7B@link) **#entity label}** : 소스코드의 또 다른 엔터티의 문서에 대한 내부링크를 추가하는 효과를 나타낸다.  
                                     \* 태그 이름의 대소문자를 구별해야 한다는 것을 유의해야 한다.

**<사용의 예>**  
\* @author 4offline  
\* @version 1.0.1 November 3 2001  
\* @since version 1.0.1  
\* @see <a href=”doc-files/jdbc1.html”> jdbc1 </a>  
\* @throws exceptiontype description  
\* @param parameterName description  
\* @return description  
\* @serial description  
\* @serialField fieldname fieldType description

**[ Javadoc의 실행 ]**

**javadoc [options] [packages] [sourcefiles] [@files]**

**<option>**

- **author** : @author 태그의 출력이 생성된다.  
- **version** : @version 태그의 출력이 생성된다.  
- **public** : 문서출력을 public 클래스와 멤버들로 제한한다.

               기본적으로는 public 클래스와 public/protected 멤버들로 제한.

- **protected** : public 클래스와 protected/public 클래스 멤버로 제한  
- **private** : 모든 클래스와 멤버들이 문서화된다.  
- **package** : 문서 출력을 패키지, protected/public 클래스와 멤버로 제한  
- **use** : 각 클래스와 인터페이스 문서의 일부로서 용법클래스 생성. 특정 클래스 또는

           인터페이스가 참조되는 곳에 대한 리스트

- **overwiew pathfilename** : 개표 페이지에 보이게 될 텍스트를 포함하는 파일의 경로와 이름을 지정한다.  
- **ddirectoryName** : 지정한 디렉토리에 html파일들을 저장한다.  
- **sourcepath pathList** : javadoc이 처리할 패키지들을 찾기 위한 경로를 지정  
- **classpath pathList** : javadoc이 문서화 주석에서 참조할 클래스들을 찾기 위한 경로  
- **extdirs dirList** : 자바 확장 메커니즘을 사용하는 클래스들을 포함하는 디렉토리들의 리스트를 지정  
- **bootclasspath pathlist** : 자바 플랫폼 클래스들을 찾기 위한 경로를 지정  
- **serialwarn** : @serial 태그를 빠트렸을 때 경고 메시지 표시  
- **verbose** : javadoc이 실행되는 동안 더 포괄적인 메시지를 만들도록 한다.

[**JAVA 주석 사용법 - javadoc**](http://blog.bagesoft.com/465)

1) JavaDoc주석을 사용하는 경우  /\*\* \*/

     Interface, Class, 멤버함수와 변수 바로 위에서 선언

     JavaClass의 Document를 자동 생성하여 활용하는 경우 필요한 정보를

     기술한다. 작성된 Document는 이후 이해하기 쉬운 용어를 이용하여 기술한다.

2) 일반Class 주석을 사용하는 경우

     코드 그 자체로 쉽게 이해할 수 없는 부분을 자세히 설명해 준다. 코드 내에서

     사용되는 변수의 의미 설명을 한다

   //  기본적으로 멀티라인 주석을 사용하는 것을 원칙으로 하나 임시로 사용하거나

           간단한주석처리 시 사용

  /\* \*/  더 이상 적용하지 않는 코드 라인들에 코멘트처리를 하거나 Method 안의

              변수 또는 코드에 주석을 달고자 할 때 사용

javadoc 로 컴파일하면 html파일로 만들어 줌

파라미터 사용법  
/\*\*  
 \* @see class\_name : "See also"라는 항목을 만들어 해당 클래스와 연결해준다.  
 \* @see class\_name#method\_name : "See also"라는 항목을 만들어 특정 메소드와 연결해준다.  
 \* @version text : HTML 파일의 버전을 표시하는 항목을 만들어 준다.  
 \* @author text : HTML 파일의 저자를 표시하는 항목을 만들어 준다.   
 \*  
 \* @param name description : 특정 메소드가 취하는 파라메터를 기술할 때 사용된다.  
 \* @return description : 특정 메소드의 리턴값을 기술할 때 사용된다.  
 \* @exception class\_name : 특정 메소드가 발생시킬 수 있는 에외상황을 기술할 때 사용된다.  
\*/  
   
사용예

/\*\*  
 \* String 에 관련된 Util 클래스   
 \* @version : 1.0  
 \* @author  : kim jeng soon ([newtake@naver.com](mailto:newtake@naver.com))  
 \*/

public class StringUtils{  
/\*\*  
  \* 문자를 replace합니다.  
  \* @param str - 문자      
  \* @param pattern  replace전  문자  
  \* @param replace  replace후  문자    
  \* @return 바꿔진 문자      
  \*/  
 public static String replace(String str, String pattern, String replace) {    
    int s = 0;   
    int e = 0;   
    StringBuffer result = new StringBuffer();

    while ((e = str.indexOf(pattern, s)) >= 0) {    
     result.append(str.substring(s, e));    
     result.append(replace);    
     s = e + pattern.length();    
    }    
    result.append(str.substring(s));    
    return result.toString();    
 }

}

사용 예

/\*\*

 \* Title : 제목 .....(Action, Dao, Dto, Form .. class)

 \* <p>Description : 내용 <p>

 \* <p>Copyright: Copyright(c) 2004 soma. All rights reserved.</p>

 \*

 \* @author Jaehak. Hwang

 \* @version $Revision. \* $

 \* $log : $

 \*

 \* @struts.action name="actionForm" path="/exemAction" scope="request"  
 \* @struts.action-forward name="pass" path="/exemAction1.do"

 \*

 \* @struts:form name="actionForm"

 \*/