Code Conventions for the Java Programming Language

- description : 5 - Comments
- author : 오션
- email : shlim@repia.com
- lastupdate : 2022-04-15 Fri

5. 들여쓰기

Java 프로그램은 구현 주석과 문서화 주석의 두 가지 주석을 가질 수 있습니다. 구현 주석은 /*...*/ 및 // 로 구분되는 C++에서 볼 수 있는 주석입니다. 문서화 주석(“ 문서 주석 ” 이라고 함) Java 전용이며 /**...*/로 구분됩니다. 문서 주석은 javadoc 도구를 사용하여 HTML 파일로 추출할 수 있습니다.

구현 주석은 코드를 주석 처리하거나 특정 구현에 대한 주석을 위한 것입니다. 문서 주석은 구현이 필요 없는 관점에서 코드의 명세를 설명하는 것으로 소스 코드가 없을 수도 있는 개발자가 읽도록 의도된 것입니다.

코드의 개요를 제공하고 코드 자체에서 쉽게 사용할 수 없는 추가 정보를 제공하기 위해 주석을 사용해야 합니다. 주석에는 프로그램을 읽고 이해하는 데 관련된 정보만 포함되어야 합니다. 예를 들어, 해당 패키지의 빌드 방법 또는 패키지가 위치한 디렉토리에 대한 정보는 주석으로 포함되어서는 안 됩니다.

사소하지 않거나 명백하지 않은 디자인 결정에 대한 논의는 적절하지만, 코드에 존재하며 명확한 정보는 중복하지 마십시오. 중복된 주석은 너무 쉽게 구식이 됩니다. 일반적으로 코드가 발전함에 따라 구식이 될 가능성이 있는 주석은 피하십시오.

Note: 주석의 빈도는 때때로 코드의 품질이 좋지 않음을 반영합니다. 주석을 더 추가해야 겠다고 느낄 때, 코드를 재작성하여 더 명확하게 하는 것을 고려하세요

별표(*)나 다른 문자로 그려진 큰 박스에 주석을 넣으면 안 됩니다.
주석에는 양식 피드 및 백스페이스와 같은 특수 문자가 포함되어서는 안 됩니다.

5.1 구현 주석 포맷 (Implementation Comment Formats)

프로그램은 블록(block), 한 줄(single-line), 후행(trailing) 및 라인 끝(end-of-line)의 네 가지 스타일의 구현 주석을 가질 수 있습니다.

5.1.1 블록 주석(Block Comments)

블록 주석은 파일, 메소드, 데이터 구조 및 알고리즘에 대한 설명을 제공하는 데 사용됩니다. 블록 주석은 각 파일의 시작과 각 메소드 전에 사용할 수 있습니다. 메서드 내부와 같은 다른 위치에서도 사용할 수 있습니다. 함수 또는 메소드 내부의 블록 주석은 설명하는 코드와 동일한 수준으로 들여쓰기 되어야 합니다.

블록 주석은 나머지 코드와 구분하기 위해 공백 행을 맨 앞에 만듭니다.

```
/*
 * 이곳이 블록 주석입니다.
 */
```

/*- 들여쓰기

```
/*-
 * 이곳에 들여쓰기를 무시하기를 원하는 특별한
 * 포맷의 블록 주석을 사용합니다.
 *      one
 *          Two
 *              Three
 */
```

5.1.2 한 줄 주석(Single-line Comments)

짧은 주석은 아래의 코드 수준으로 들여쓰기된 한 줄에 작성할 수 있습니다. 주석을 한 줄로 작성할 수 없는 경우, 블록 주석 포맷을 준수해야 합니다. 한 줄 주석 앞에는 빈 줄이 있어야 합니다. 다음은 Java 코드의 한 줄 주석의 예제입니다.

```
if (condition) {

    /* 조건을 처리합니다. */
    ...
}
```

5.1.3 후행 주석 (Trailing Comments)

매우 짧은 주석은 설명하는 코드와 같은 줄에 작성할 수 있지만, 명령문에서 주석을 분리할 수 있을 만큼 충분히 멀리 이동해야 합니다. 코드 덩어리에 둘 이상의 짧은 주석이 표시되면 모두 동일한 탭 설정으로 들여써야 합니다.

다음은 Java 코드에서 후행 주석의 예입니다:

```
if (a == 2) {
    return TRUE;            /* 특별한 경우*/
} else {
    return isPrime(a);      /* a가 홀수인 경우에만 동작 */
```

5.1.4 라인 끝 주석 (End-Of-Line Comments)

// 주석 구분 기호는 전체 행 또는 일부 행만 주석 처리할 수 있습니다. 텍스트 주석에 대해 연속된 여러 줄에 사용하면 안 됩니다. 그러나 코드 섹션을 주석 처리하기 위해 연속된 여러 줄에서 사용할 수 있습니다. 세 가지 스타일의 예는 다음과 같습니다.

```
if (foo > 1) {

    // Do a double-flip.
    ...
}
else {
    return false;          // Explain why here.
}
//if (bar > 1) {
//
//    // Do a triple-flip.
//    ...
//}
//else {
//    return false;
//}
```

5.2 문서화 주석 (Documentation Comments)

Note : 여기에 설명된 주석 포맷의 예제는 "Java 소스 파일 예제 " 를 참조하십시오.

자세한 내용은 문서 주석 태그(@return, @param, @see)에 대한 정보가 포함된 "Javadoc용 문서 주석 작성 방법 " 을 참조하십시오.

문서 주석은 Java 클래스, 인터페이스, 생성자, 메소드 및 필드를 설명합니다. 각 문서 주석은 주석 구분 기호 **…*/** 안에 설정되며 클래스, 인터페이스 또는 멤버당 하나의 주석이 있습니다. 이 주석은 선언 직전에 나타나야 합니다.

```
/**
 * The Example class provides ...
 */
public class Example { ...
```

최상위 클래스와 인터페이스는 들여쓰기가 되어 있지 않지만 그 멤버들은 들여쓰기되어 있습니다. 클래스 및 인터페이스에 대한 문서 주석(/**)의 첫 번째 줄은 들여쓰기되지 않습니다; 후속 문서 주석 줄에는 각각 1개의 들여쓰기 공간이 있습니다(별표를 세로로 정렬하기 위해). 생성자를 포함한 멤버에는 첫 번째 문서 주석 줄에 대한 4개의 공백이 있고 그 이후에는 5개의 공백이 있습니다.

문서화에 적합하지 않은 클래스, 인터페이스, 변수 또는 메소드에 대한 정보를 제공해야 하는 경우, 선언 바로 뒤에 구현 블록 주석 또는 한 줄 주석을 사용하십시오. 예를 들어, 클래스 구현에 대한 세부 정보는 클래스 문서 주석이 아니라 클래스 문 다음의 구현 블록 주석에 들어가야 합니다.

Java는 문서화 주석을 주석 뒤의 첫 번째 선언과 연관시키기 때문에, 문서 주석은 메소드 또는 생성자 정의 블록 내부에 위치해서는 안 됩니다.

Ref Site

Code Conventions for the Java Language 5 - Comments

오션,, Comments

From:
http://125.132.25.164/dokuwiki/ - **문제를 잘 정의하는 것은 문제를 절반 해결한 것이다. - 2023.12**

Permanent link:
http://125.132.25.164/dokuwiki/doku.php?id=wiki:miscellaneous:code_conventions:comments&rev=1650241942

Last update: **2022/04/18 09:32**