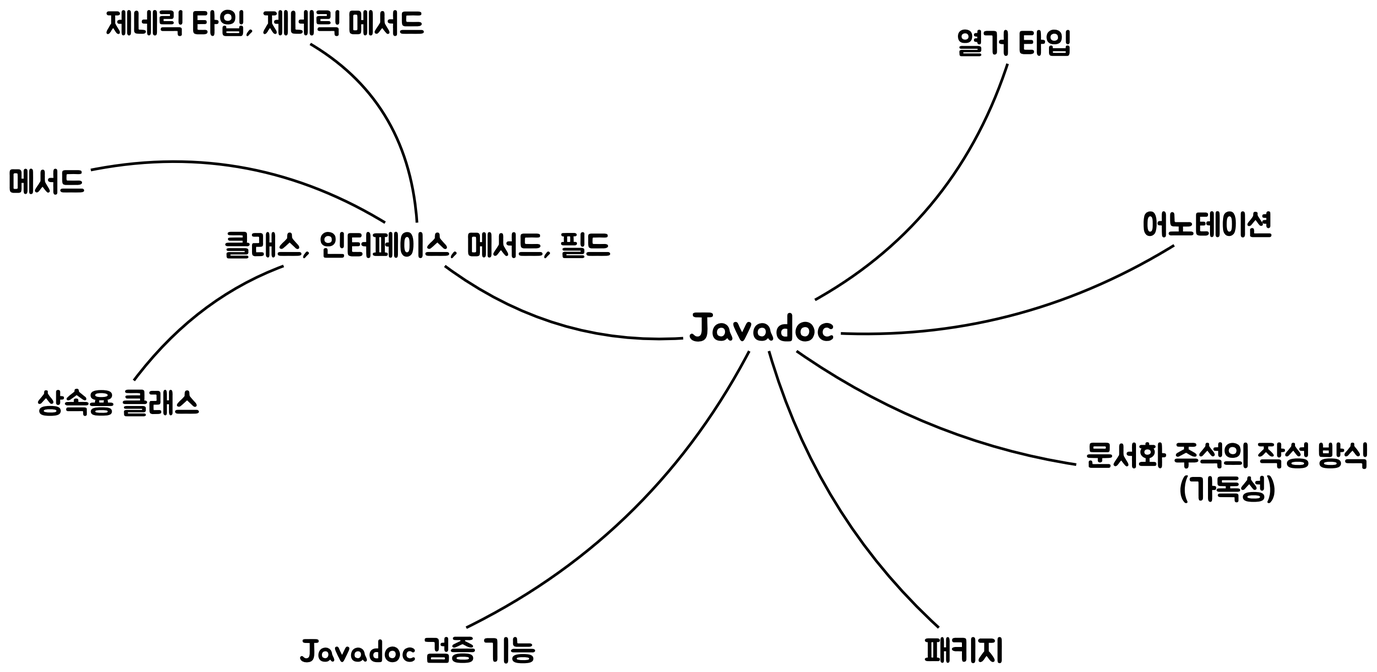
아이템 56
'공개된 API 요소에는 항상 문서화 주석을 작성하라'

'공개된 API 요소에는 항상 문서화 주석을 작성하라'

참고

- Effective Java 3/E - item 56
- Oracle Javadoc
- Oracle Java API Document Generator
- The package-info.java File

'공개된 API 요소에는 항상 문서화 주석을 작성하라'



- 문서화 주석(Javadoc)

1. Javadoc의 태그들

2.메서드용 문서화

3.클래스용 문서화

4.제네릭 타입이나 메서드의 문서화

5.열거 타입의 문서화

6.어노테이션 타입의 문서화

Javadoc의 태그들

@param
- 매개변수의 설명에 대한 태그

@return
- 반환 타입이 void가 아닌 경우 반환타입에 대한 설명을 기술하는 태그

@throws
- 비검사 예외를 선언하여 암시적으로 기술한다.
- 발생할 가능성이 있는 모든 예외에 사용하는 태그

{@code ... }
- JDK 5
- 태그로 감싼 내용을 코드용 폰트로 렌더링하는 태그
- 태그로 감싼 내용에 HTML 요소나 다른 자바독 태그를 무시하는 태그

{@implSpec ... }
- 해당 메서드와 하위 클래스 사이의 계약을 설명하여,
  하위 클래스들이 그 메서드를 상속하거나 super 키워드를 이용해 호출할 때
  그 메서드가 어떻게 동작하는 지를 명확하게 인지하고 사용하도록 해줘야 한다.

{@literal ... }
- JDK 5
- HTML 메타문자를 포함하기 위한 태그

{@summary ... }
- JDK 10
- 요약 설명 전용 태그
- 메서드와 생성자의 요약 설명은 기능에 대한 동사구여야 한다.

{@index ... }
- JDK 9
- API에서 중요한 용어를 추가로 색인하기 위한 태그

{@inheritDoc}
- 상위 타입의 문서화 주석을 일부 상속하기 위한 태그

메서드용 문서화 주석을 사용할 때의 주의사항

메서드와 클라이언트 사이의 규약을 명료하게 기술하기 위한 조건

- 해당 메서드가 **무엇을 하는지**를 기술해야 한다. (what)

- 클라이언트가 해당 메서드를 호출하기 위한 **전제조건**을 나열해야 한다.
  - 일반적으로 전제조건은 @throws 태그로 비검사 예외를 선언하여 암시적으로 기술
  - @param 태그를 이용해 그 조건에 영향받는 매개변수에 기술

- 메서드가 성공적으로 수행된 뒤에 만족해야 하는 **사후조건**을 나열해야 한다.

- **부작용**에 대한 문서화도 필요하다.

- @param, @return 태그는 매개변수가 뜻하는 값이나 반환하는 값을 설명하는 **명사구**로 또는 **산술 표현식**으로 쓴다.
- @param, @return, @throws 태그의 설명에는 **마침표를 붙이지 않는다.**

- **{@inheritDoc}** 태그를 사용하는 경우 상위 타입의 문서화 주석 일부를 상속할 수 있다.
  - 하지만 이는 사용하기 까다롭고 제약이 있다.

메서드용 문서화 주석을 사용할 때의 주의사항

메서드와 클라이언트 사이의 규약을 명료하게 기술하기 위한 조건

- {@code ... } 태그
  - 태그로 감싼 내용을 코드용 폰트로 렌더링한다.
  - 태그로 감산 내용에 포함된 HTML 요소나 다른 자바독 태그를 무시한다.
  - 여러 줄로 된 코드를 예시로 넣기 위해서는 <pre>{@code ...}</pre> 태그로 감싸야 코드의 줄바꿈이 그대로 유지된다.

- {@literal ... } 태그
  - HTML 마크업이나 자바독 태그를 무시하게 해주지만, 코드 폰트로 렌더링 하지 않는다.

클래스용 문서화 주석을 사용할 때의 주의사항

클래스를 상속용으로 설계하는 경우 자기사용 패턴(self-use pattern)에 대해 재정의하는 방법 설명

- @implSpec
  - 해당 메서드와 클라이언트 사이의 계약을 설명
  - 하위 클래스들이 그 메서드를 상속하거나 super 키워드를 이용해 호출하는 경우 동작을 명확히 인지하고 사용하도록 가이드
  - JDK 11 에서 javadoc 명령어를 사용하지 않으면 활성화되지 않는다.

문서화 주석의 가독성

- 첫 번째 문장은 해당 요소의 요약 설명으로 간주한다.
  - 요약 설명은 반드시 대상의 기능을 고유하게 기술해야 한다.
  - 요약 설명에는 마침표를 주의해야 한다. (마침표가 요약 설명의 끝을 나타내어 뒷 내용이 나타나지 않는다.)

- 메서드와 생성자의 요약 설명
  - 해당 메서드와 생성자의 동작을 설명하는 동사구여야 한다.

- 클래스, 인터페이스, 필드의 요약 설명
  - 대상을 설명하는 명사절이어야 한다.

- {@literal ... }
  - 마침표로 인한 문제를 처리하기 위해 마침표를 포함한 문자를 감싸는 임시적인 방법
  - {@literal Mrs. Peacok}

- {@summary ... }
  - JDK 10
  - 요약 설명 전용 태그
  - 위 @literal 의 문제점을 해결 가능

- {@index ... }
  - JDK 9
  - HTML 문서에 검색(색인) 기능이 추가
  - 클래스, 메서드, 필드 같은 API 요소의 색인은 자동으로 만들어진다.
  - 원하는 경우 {@index} 태그를 사용하여 API에서 중요한 용어를 추가로 색인 할 수 있다.

제네릭 타입의 문서화 주석 사용시 주의사항

모든 타입의 매개변수에 대해 주석이 필요하다.

```
All arguments to all task methods must be non-null.

This class is a member of the Java Collections Framework.

Since:          1.5
Author:         Doug Lea
Type parameters: <K> – the type of keys maintained by this map
                 <V> – the type of mapped values
```

```
public class ConcurrentHashMap<K,V> extends AbstractMap<K,V>
    implements ConcurrentMap<K,V>, Serializable {
    private static final long serialVersionUID = 7249069246763182397L;
```

```
    Returns the value to which the specified key is mapped, or defaultValue if this map contains no mapping
    for the key.
    Params:                     key – the key whose associated value is to be returned
                                defaultValue – the default mapping of the key
    Returns:                    the value to which the specified key is mapped, or defaultValue if this map
                                contains no mapping for the key
    Throws:                     ClassCastException – if the key is of an inappropriate type for this map
                                (optional)
                                NullPointerException – if the specified key is null and this map does not
                                permit null keys (optional)
    Implementation Requirements: The default implementation makes no guarantees about synchronization or
                                atomicity properties of this method. Any implementation providing atomicity
                                guarantees must override this method and document its concurrency
                                properties.
    Since:                      1.8
```

```
    default V getOrDefault(Object key, V defaultValue) {
        V v;
        return (((v = get(key)) != null) || containsKey(key))
            ? v
            : defaultValue;
    }
```

- 문서화 주석(Javadoc)

1. Javadoc의 태그들

2.메서드용 문서화

3.클래스용 문서화

4.제네릭 타입이나 메서드의 문서화

5.열거 타입의 문서화

6.어노테이션 타입의 문서화

열거 타입의 문서화 주석 사용시 주의사항

- 상수들에 주석을 달아야 한다.
- 열거 타입 자체, 열거 타입의 public 메서드에도 주석을 달아야 한다.
- 설명이 짧은 경우 주석 전체를 한 문장으로 사용해도 된다.

```
Enumeration of common database drivers.
Since:  1.4.0
Author: Phillip Webb, Maciej Walkowiak, Marten Deinum, Stephane Nicoll
public enum DatabaseDriver {

    Unknown type.
    UNKNOWN( productName: null,  driverClassName: null),

    Apache Derby.
    DERBY( productName: "Apache Derby",  driverClassName: "org.apache.derby.jdbc.EmbeddedDriver",  xaDataSourceClassName: "org.apache.derby.jdbc.EmbeddedXADataSource",
            validationQuery: "SELECT 1 FROM SYSIBM.SYSDUMMY1"),

    H2.
    H2( productName: "H2",  driverClassName: "org.h2.Driver",  xaDataSourceClassName: "org.h2.jdbcx.JdbcDataSource",  validationQuery: "SELECT 1"),

    HyperSQL DataBase.
    HSQLDB( productName: "HSQL Database Engine",  driverClassName: "org.hsqldb.jdbc.JDBCDriver",  xaDataSourceClassName: "org.hsqldb.jdbc.pool.JDBCXADataSource",
            validationQuery: "SELECT COUNT(*) FROM INFORMATION_SCHEMA.SYSTEM_USERS"),

    SQL Lite.
    SQLITE( productName: "SQLite",  driverClassName: "org.sqlite.JDBC"),

    MySQL.
    MYSQL( productName: "MySQL",  driverClassName: "com.mysql.cj.jdbc.Driver",  xaDataSourceClassName: "com.mysql.cj.jdbc.MysqlXADataSource",  validationQuery: "/* ping */ SELECT 1"),
```

어노테이션 타입의 문서화 주석 사용시 주의사항

- 어노테이션 타입 자체, 어노테이션의 모든 멤버에 주석을 달아야 한다.
- 필드의 설명은 명사구로 작성한다.
- 어노테이션 타입의 요약 설명은 동사구로 작성한다.

```
Since:    1.2
See Also: org.springframework.transaction.interceptor.TransactionAttribute, org.springframework.transaction.
          interceptor.DefaultTransactionAttribute, org.springframework.transaction.interceptor.
          RuleBasedTransactionAttribute
Author:   Colin Sampaleanu, Juergen Hoeller, Sam Brannen
```

```java
@Target({ElementType.TYPE, ElementType.METHOD})
@Retention(RetentionPolicy.RUNTIME)
@Inherited
@Documented
public @interface Transactional {

    Alias for transactionManager.
    See Also: transactionManager

    @AliasFor("transactionManager")
    String value() default "";

    A qualifier value for the specified transaction.
    May be used to determine the target transaction manager, matching the qualifier value (or the bean name)
    of a specific TransactionManager bean definition.
    Since:    4.2
    See Also: value, org.springframework.transaction.PlatformTransactionManager, org.springframework.
              transaction.ReactiveTransactionManager

    @AliasFor("value")
    String transactionManager() default "";

    The transaction propagation type.
    Defaults to Propagation.REQUIRED.
    See Also: org.springframework.transaction.interceptor.TransactionAttribute.getPropagationBehavior()

    Propagation propagation() default Propagation.REQUIRED;
```

- 문서화 주석(Javadoc)

1. Javadoc의 태그들
2. 메서드용 문서화
3. 클래스용 문서화
4. 제네릭 타입이나 메서드의 문서화
5. 열거 타입의 문서화
6. 어노테이션 타입의 문서화

그 외

API 문서화의 주의사항

- 스레드 안전성, 직렬화 가능에 대한 정보
- 클래스 혹은 정적 메서드의 Thread 안전 수준을 반드시 API 설명에 포함해야 한다. [아이템 82]
- 직렬화 할 수 있는 클래스의 경우 직렬화 형태도 API 설명에 기술해야 한다. [아이템 87]
- 모든 API 요소에 문서화 주석을 달았더라도 전체 아키텍처를 설명하는 별도의 설명이 필요한 경우 링크를 활용하면 좋다.

정리

정리

- 문서화 주석은 API를 문서화하는 가장 훌륭하고 효과적인 방법이다.
- 공개 API의 경우 빠짐없이 설명을 달아야 한다.
- 표준 규약을 일관되게 지켜야 하고, 문서화 주석에 임의의 HTML 태그를 사용할 수 있다.
- Spring Rest Docs, Swagger, APIDoc, Asciidoctor, slate