

# Chapter 08 자바 API 패키지, 예외 처리, 모듈

---

## 8.1 패키지란?

### 패킷지의 개념

- 패키지: 관련 있는 클래스들을 하나로 묶은 것

### 왜 패키지가 필요할까?

1. 서로 관련된 클래스들을 하나의 단위로 모아 쉽게 유지 관리할 수 있다.
2. 패키지 안에 있는 클래스들은 패키지 안에서만 사용할 수 있도록 선언하여 캡슐화를 구현할 수 있다.
3. 패키지가 다르면 동일한 클래스의 이름을 사용할 수 있다.

## 8.2 패키지 선언하기

- 패키지 선언문은 반드시 소스 파일의 첫 번째 문장이어야 한다.
- 여러 개의 소스 파일에 동일한 패키지 선언문을 넣을 수 있다.
- 패키지는 계층 구조를 가질 수 있다.

### 이클립스에서 패키지 만들기

### 소스 파일의 위치

### 클래스 파일의 위치

## 8.3 패키지 사용하기

1. 완전한 이름으로 참조 - 패키지 이름을 클래스 앞에 적는다. -> 클래스 이름만 가지고 클래스 참조 or import
2. 패키지 안에서 우리가 원하는 클래스만을 포함한다.
3. 패키지 안의 모든 클래스를 포함한다.

### 계층 구조의 패키지 포함하기

### 정적 import 문장

- 일반적으로 클래스 안에 정의된 정적 상수나 정적 메소드를 사용하는 경우에 클래스 이름을 앞에 적어야 한다.
- 하지만 정적 import 문장을 사용하면 클래스의 이름을 생략할 수 있다.

## 8.4 클래스 파일은 언제 로드될까?

### 언제 클래스 파일이 로드되는가?

- Lazy Class Loading(지연 클래스 로드): 요청 시 클래스 파일은 JVM에 로드된다. -> JVM의 메모리를 절약할 수 있다.
- 시작 시 애플리케이션 코드를 구성하는 파일은 로드된다.

### 자바 가상 머신이 클래스를 찾는 순서

1. 부트스트랩 클래스: 자바 플랫폼을 구성하는 핵심적인 클래스
2. 확장 클래스: 자바 확장 메커니즘을 사용하는 클래스
3. 사용자 클래스: 확장 메커니즘을 활용하지 않는 개발자 및 타사에 의해 정의된 클래스

## CLASSPATH

- 사용자가 직접 작성하였거나 외부에서 다운로드 받은 클래스를 찾기 위하여 가상 머신이 살펴보는 디렉토리들을 모아둔 경로

클래스 경로를 지정하는 3가지 방법

1. 현재 디렉토리
2. CLASSPATH에 설정된 환경 변수 설정
3. -classpath 옵션

## JAR 압축 파일

이클립스를 사용하는 경우

JAR 형태의 라이브러리 사용하기

## 8.5 자바 API 패키지

## 8.6 Object 클래스

- java.lang 패키지에 들어 있으며, 자바 클래스 계층 구조에서 맨 위에 위치하는 클래스

Object 클래스 안에 정의된 메소드

| 이름                                | 기능                     |
|-----------------------------------|------------------------|
| public boolean equals(Object obj) | obj가 이 객체와 같은지 검사한다.   |
| public String toString()          | 객체의 문자열 표현을 반환한다.      |
| protected Object clone()          | 객체 자신의 복사본을 생성하여 반환한다. |
| public int hashCode()             | 객체에 대한 해쉬 코드를 반환한다.    |
| protected void finalize()         | 가비지 컬렉터에 의하여 호출된다.     |
| public final Class getClass()     | 객체의 클래스 정보를 반환한다.      |

### getClass() 메소드

- 객체가 어떤 클래스로 생성되었는지에 대한 정보를 반환한다.
- reflection(리플렉션): 객체가 자신을 만든 클래스에 대하여 물어보는 것

### toString() 클래스

- 객체의 상태를 몇 줄의 문자열로 요약하여 반환하는 함수
- 일반적으로 재정의하여야 하는 메소드

## equals() 메소드

- 동등 비교
- '==' : 참조값 비교
- equals() : 내용 비교

## 8.7 래퍼 클래스

- 기초 자료형을 객체로 포장하는 클래스

### 오토박싱

- 래퍼 객체와 기초 자료형 사이의 변환을 자동으로 해주는 기능

## 8.8 String 클래스

### String 클래스

#### 객체 생성

문자열 객체를 생성하는 두 가지 방법

```
String s1 = new String("java");  
  
String s2 = "java";
```

#### 문자열의 기초 연산들

- length()
- charAt()
- concat()

#### 문자열 비교하기

#### 문자열 안에서 단어 찾기

#### 문자열을 단어로 분리할 때

#### StringBuffer 클래스

## 8.9 기타 유용한 클래스