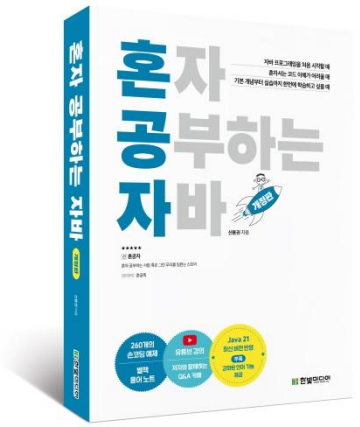
Chapter 06

클래스



06-6. 패키지와 접근 제한자

혼자 공부하는 자바(개정판) (신용권 저)

목차

시작하기 전에

[핵심 키워드] : 패키지 선언, import문, 접근 제한자, Getter/Setter

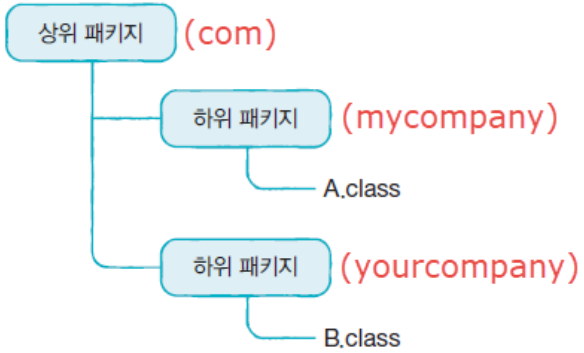
[핵심 포인트]

프로젝트 개발 시 클래스를 체계적으로 관리하기 위해 패키지를 사용한다.
클래스와 클래스의 멤버를 사용 범위에 맞게 접근 제한자를 활용한다.

❖ 패키지

- 패키지의 물리적인 형태는 파일 시스템의 폴더
- 패키지는 클래스의 일부분으로, 클래스를 유일하게 만들어주는 식별자 역할
- 클래스 이름이 동일하더라도 패키지가 다르면 다른 클래스로 인식
- 클래스의 전체 이름은 패키지+클래스 사용해서 다음과 같이 표현

  - 상위패키지.하위패키지.클래스
  - com.mycompany.A
  - com.yourcompany.B



패키지 선언

❖ 패키지 선언

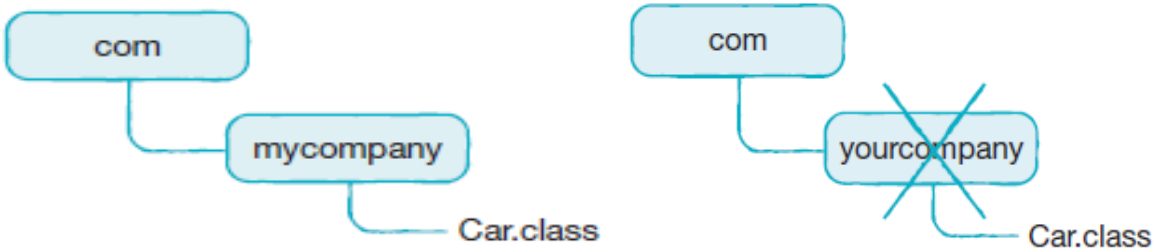
- 클래스 작성 시 해당 클래스가 어떤 패키지에 속할 것인지를 선언

```
package 상위패키지.하위패키지;

public class ClassName { … }
```

```
package com.mycompany;

public class Car { … }
```



- 패키지 이름 규칙
  - - 숫자로 시작 불가
  - - _ 및 $ 제외한 특수문자 사용 불가
  - - java로 시작하는 패키지는 자바 표준 API 에서만 사용하므로 사용 불가
  - - 모두 소문자로 작성하는 것이 관례

패키지 선언

❖ import문

- 사용하고자 하는 클래스 또는 인터페이스가 다른 패키지에 소속된 경우
- 해당 패키지 클래스 또는 인터페이스 가져와 사용할 것임을 컴파일러에 통지

```
import 상위패키지.하위패키지.클래스이름;
import 상위패키지.하위패키지.*;
```

-> 이 패키지 안의 모든 클래스를 가져오겠다는 뜻

```
package com.mycompany;

import com.hankook.Tire;
[ 또는 import com.hankook.*; ]

public class Car {
 Tire tire = new Tire();
}
```

- 패키지 선언과 클래스 선언 사이에 작성

- 하위 패키지는 별도로 import를 해야함

```
import com.hankook.*;
import com.hankook.project.*;
```

- 다른 패키지에 동일한 이름의 클래스가 있을 경우

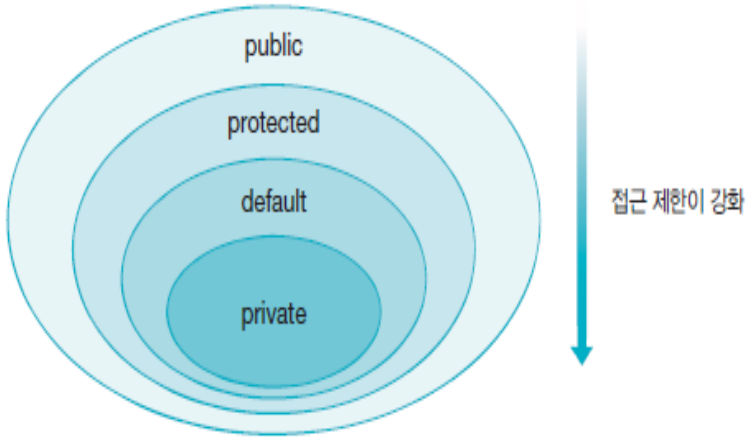
  import와 상관없이 클래스 전체 이름을 기술

접근 제한자

❖ 접근 제한자 (access modifier)

- 클래스와 인터페이스 및 이들이 가진 멤버의 접근 제한

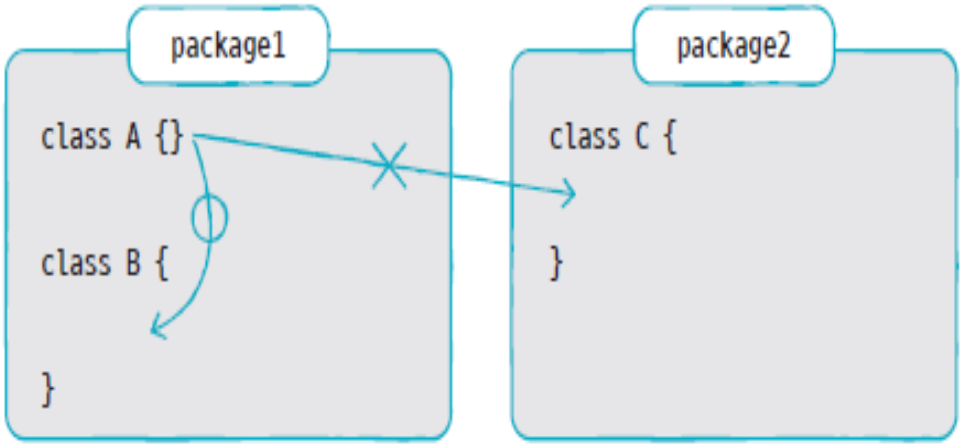
- public 접근 제한자
  - 외부 클래스가 자유롭게 사용할 수 있도록 함
- protected 접근 제한자
  - 같은 패키지 또는 자식 클래스에서 사용할 수 있도록 함
- private 접근 제한자
  - 외부에서 사용할 수 없도록 함
- default 접근 제한
  - 같은 패키지에 소속된 클래스에서만 사용할 수 있도

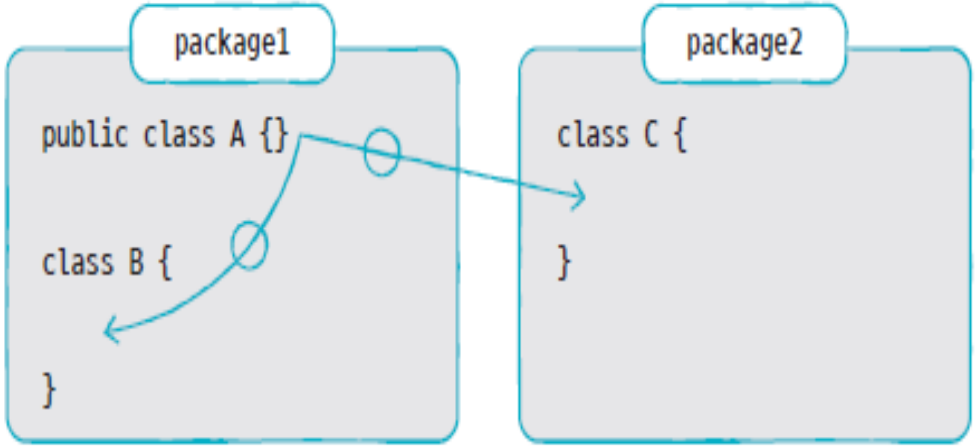


클래스의 접근 제한

❖ 클래스 접근 제한

- 같은 패키지 내에서만 사용할 것인지 다른 패키지 내에서도 사용할 수 있도록 할 것인지 결정





생성자의 접근 제한

❖ 생성자 접근 제한에 따라 생성자 호출 가능 여부 결정

```
public class ClassName {
  //public 접근 제한
  public ClassName(…) { … }

  //protected 접근 제한
  protected ClassName(…) { … }

  //default 접근 제한
  ClassName(…) { … }

  //private 접근 제한
  private ClassName(…) { … }
}
```

필드와 메소드의 접근 제한

❖ 필드와 메소드의 접근 제한

```
//필드 선언
[ public ¦ protected ¦ private ] [static] 타입 필드;

//메소드 선언
[ public ¦ protected ¦ private ] [static] 리턴 타입 메소드(…) { … }
```

Getter와 Setter 메소드

❖ 외부에서 객체에 마음대로 접근할 경우 객체의 무결성 깨질 수 있음

❖ Setter 메소드
- 외부의 값을 받아 필드의 값을 변경하는 것이 목적
- 매개값 검증하여 유효한 값만 필드로 저장할 수 있음 - 유효성 검사

```
void setSpeed(double speed) {
  if(speed < 0) {
    this.speed = 0;
    return;
  } else {
    this.speed = speed;
  }
}
```

매개값이 음수일 경우 speed 필드에 0으로 저장하고, 메소드 실행 종료

Getter와 Setter 메소드

❖ Getter 메소드

- 외부로 필드값을 전달하는 것이 목적
- 필드값을 가공해서 외부로 전달할 수도 있음

```
double getSpeed() {
  double km = speed*1.6;
  return km;
}
```

필드값인 마일을 km 단위로 환산 후 외부로 리턴

관례적으로, boolean일 경우에만 get 대신 is가 붙음

키워드로 끝내는 핵심 포인트

- 패키지 선언 : 해당 클래스 또는 인터페이스가 어떤 패키지에 속할 것인지를 선언.

```
package 상위패키지.하위패키지;
```

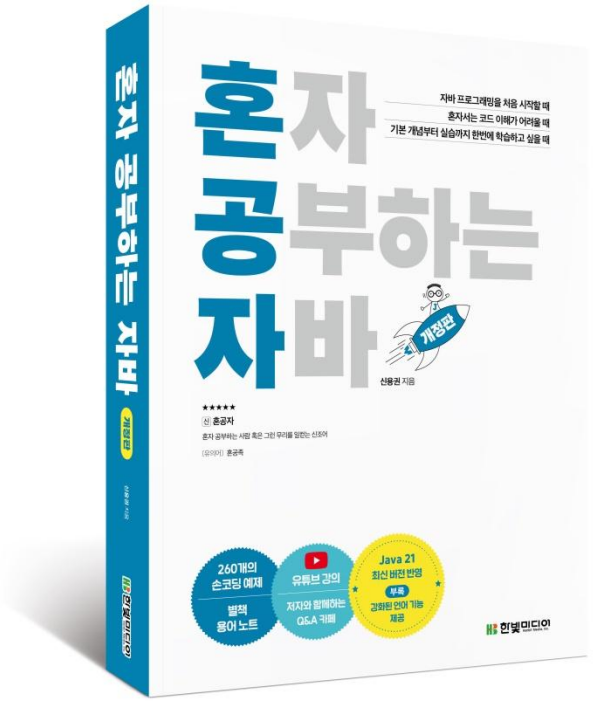
- import문 : 다른 패키지에 소속하는 클래스와 인터페이스를 사용할 경우 필요

```
import 상위패키지.하위패키지.클래스이름;
import 상위패키지.하위패키지.*;
```

- 접근 제한자 : 클래스, 인터페이스, 그리고 멤버들을 사용을 제한할 경우 사용

접근 제한	적용 대상	접근할 수 없는 클래스
public	클래스, 필드, 생성자, 메소드	없음
protected	필드, 생성자, 메소드	자식 클래스가 아닌 다른 패키지에 소속된 클래스
default	클래스, 필드, 생성자, 메소드	다른 패키지에 소속된 클래스
private	필드, 생성자, 메소드	모든 외부 클래스

- Getter/Setter :
  - 필드 값을 외부로 리턴하는 메소드를 Getter (getXXX(), isXXX())
  - 외부에서 값을 받아 필드값을 변경하는 메소드를 Setter (setXXX())


혼자 공부하는 자바
혼자 공부하는 자바
개정판
자바 프로그래밍을 처음 시작할 때
혼자서는 코드 이해가 어려울 때
기본 개념부터 실습까지 한번에 학습하고 싶을 때
개정판
신용권 지음
★★★★★
신 혼공자
혼자 공부하는 사람 혹은 그런 무리를 일컫는 신조어
[유의어] 혼공족
260개의 손코딩 예제
별책 용어 노트
유튜브 강의
저자와 함께하는 Q&A 카페
Java 21 최신 버전 반영
부록 강화된 언어 기능 제공
한빛미디어

Thank You!