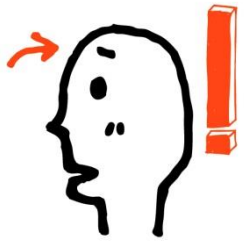
Android Java

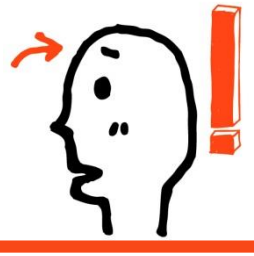
7장 패키지의 이해와 Getter/Setter

package와 import 키워드로 명시적 선언하기



- **패키지** : 서로 관련 있는 클래스 파일들을 하나의 폴더로 모아서 관리하기 위한 묶음
- 클래스를 사용하면 기능별 혹은 의미별로 패키지를 관리하기 용이
- 같은 패키지 안에서 클래스 이름은 중복해서 사용할 수 없음
- 패키지는 디렉토리와 같은 계층적 구조
- 패키지명을 쓸 때는 상위 패키지 이름부터 하위 패키지 이름순으로 표기
- 상위 디렉토리와 하위 디렉토리는 점 (.)을 사용하여 구분

package와 import 키워드로 명시적 선언하기



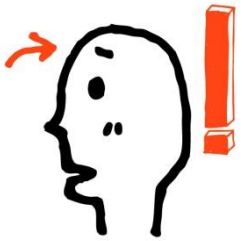
소속을 밝히는 package 키워드

- package는 이 클래스가 어떤 패키지에 포함되는지 명시적으로 표현하기 위해서 사용
- 클래스 중 가장 처음에 선언

```
사용법 : package [패키지 이름];
사용예 : package com.gilbut.chapter5;
```

- 만약 실수로 package 키워드를 사용하지 않고 클래스를 선언하더라도
  JVM에서는 기본적으로 default 패키지에 해당 클래스를 포함

package와 import 키워드로 명시적 선언하기



"이 클래스는 orange.area 패키지에 묶겠다!"

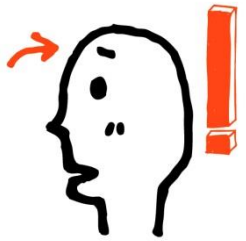
이 클래스를 orange\area 디렉터리에 저장하고,
이 경로를 명시해서 인스턴스를 생성하겠다!

```
package orange.area; // 패키지 선언

public class Circle
{

   . . . .

}
```

인스턴스의 생성방법

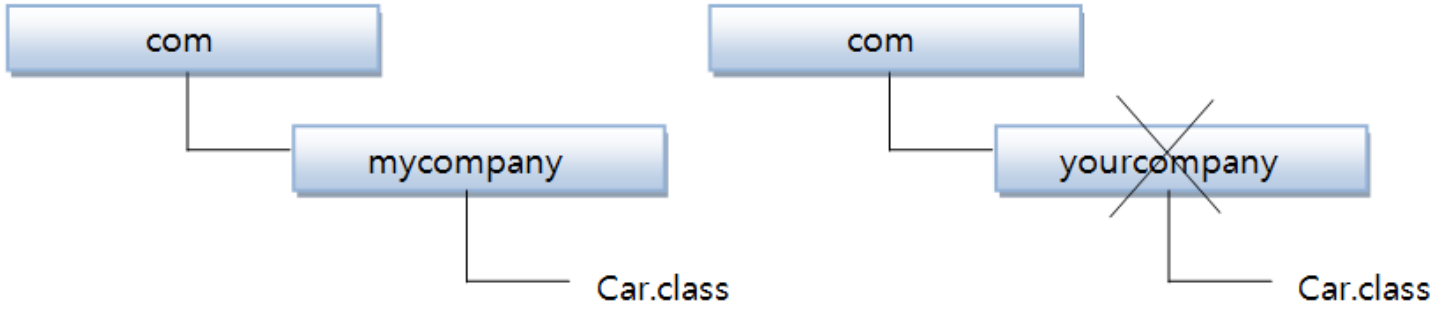
```
orange.area.Circle c1=new orange.area.Circle(1.5);
System.out.println("반지름이 1.5인 원의 넓이 : "+c1.getArea());
```

package와 import 키워드로 명시적 선언하기

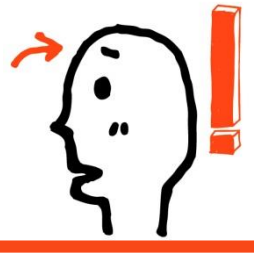


클래스 선언할 때 패키지 결정

- 클래스 선언할 때 포함될 패키지 선언
- 클래스 파일은(~.class) 선언된 패키지와 동일한 폴더 안에서만 동작
- 클래스 파일은(~.class) 다른 폴더 안에 넣으면 동작하지 않음



package와 import 키워드로 명시적 선언하기

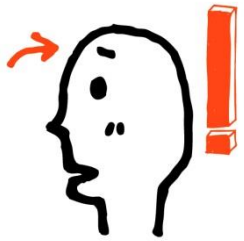


참조를 위한 import 키워드

- **import**는 다른 클래스의 메소드나 속성을 참조할 때, 명시적으로 표현하기 위해서 사용
- 클래스의 이름만으로는 어떤 패키지에 속해 있는지 모르기 때문에 중복된 것을 참조할 수 있는데, 이때 사용하는 구문이 import 키워드
- import 구문을 사용하는 방법
  : 클래스를 직접 명시
  : 사용하고자 하는 클래스가 포함된 패키지 전체를 포함

```
사용법 : import [패키지 이름].[클래스 이름] ;        //클래스 이름을 직접 명시
         import [패키지 이름].* ;                   //클래스가 포함된 패키지 전체를 포함
사용예 : import java.util.Random;                  //Random 클래스만 import
         import java.util.*;                       //java.util package의 모든 클래스를 import
```

package와 import 키워드로 명시적 선언하기

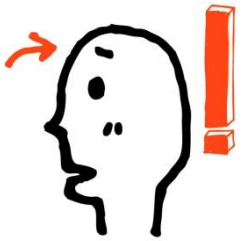


```
package com.mycompany;

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

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

Source>Organize imports (단축키: Ctrl+Shift+O)

1. Window>Preference>Java>Code Style>Organize imports 를 선택
2. Number of imports needed for .*의 99 를 1 로 변경하고 [OK] 버튼을 클릭한다.
3. 다시한번 Ctrl+Shift+O 를 클릭한다.

Getter와 Setter 선언하기



- **클래스 선언할 때 필드는 일반적으로 private 접근 제한**
  - 읽기 전용 필드가 있을 수 있음 (Getter의 필요성)
  - 외부에서 엉뚱한 값으로 변경할 수 없도록 (Setter의 필요성)

- **Getter**
  - private 필드의 값을 리턴 하는 역할 - 필요할 경우 필드 값 가공
  - getFieldName() 또는 isFieldName() 메소드
    - 필드 타입이 boolean 일 경우 isFieldName()

- **Setter**
  - 외부에서 주어진 값을 필드 값으로 수정
    - 필요할 경우 외부의 값을 유효성 검사
  - setFieldName(타입 변수) 메소드
    - 매개 변수 타입은 필드의 타입과 동일

THANK YOU

실무에서 알아야 할 기술은 따로 있다! 자바를 다루는 기술