

패키지(package)

패키지(package)

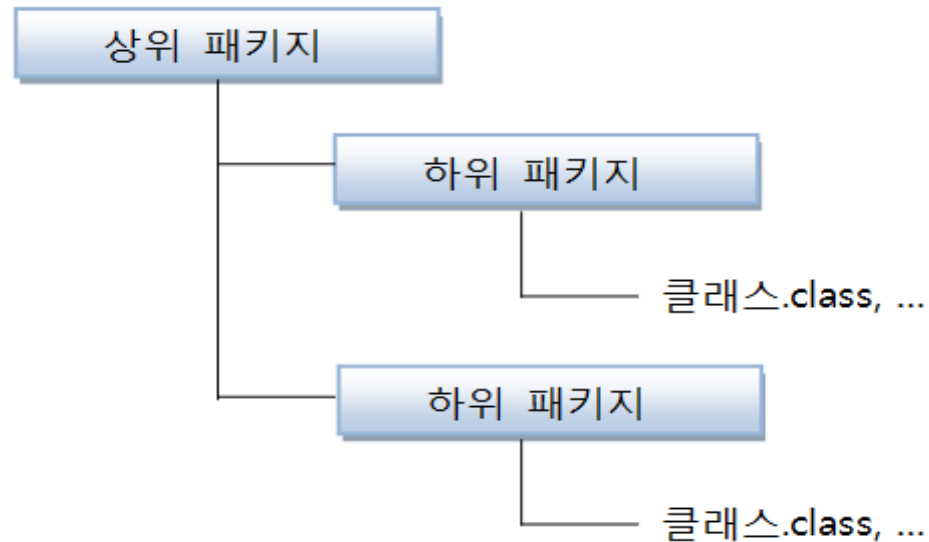
❖ 패키지란?

- 클래스를 기능별로 묶어서 그룹 이름을 붙여 놓은 것
 - 파일들을 관리하기 위해 사용하는 폴더(디렉토리)와 비슷한 개념
 - 패키지의 물리적인 형태는 파일 시스템의 폴더

패키지(package)

❖ 패키지란?

- 클래스 이름의 일부
 - 클래스를 유일하게 만들어주는 식별자
 - 전체 클래스 이름 = 상위패키지.하위패키지.클래스
 - 클래스명이 같아도 패키지명이 다르면 다른 클래스로 취급



패키지(package)

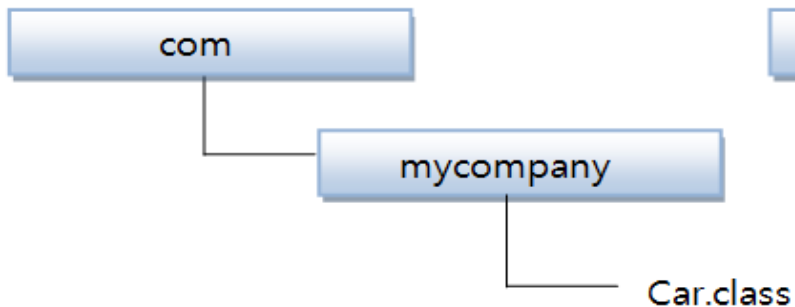
❖ 패키지란?

- 클래스 선언할 때 패키지 결정
 - 클래스 선언할 때 포함될 패키지 선언

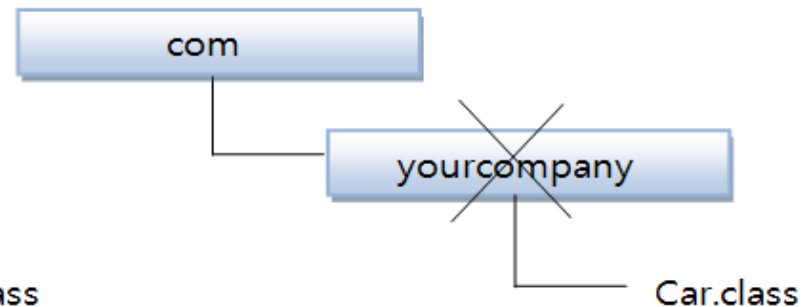
```
package 상위패키지.하위패키지;
```

```
public class ClassName { ... }
```

- 클래스 파일은(~.class) 선언된 패키지와 동일한 폴더 안에서만 동작
- 클래스 파일은(~.class) 다른 폴더 안에 넣으면 동작하지 않음



com.mycompany.Car
→ com\mycompany\Car.class



com.yourcompany.Car
→ com\yourcompany\Car.class

패키지(package)

❖ import 문

- 패키지 내에 같이 포함된 클래스간 클래스 이름으로 사용 가능
- 패키지가 다른 클래스를 사용해야 할 경우
 - 패키지 명 포함된 전체 클래스 이름으로 사용

```
package com.mycompany;
```

```
public class Car {  
    com.hankook.Tire tire = new com.hankook.Tire();  
}  
    타입        필드명        객체 생성
```

패키지(package)

❖ import 문

- import 문으로 패키지를 지정하고 사용

```
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.

패키지: sec12.exam03_import.hankook

【 SnowTire.java 】

```
01 package sec12.exam03_import.hankook;
02
03 public class SnowTire { }
```

【 Tire.java 】

```
01 package sec12.exam03_import.hankook;
02
03 public class Tire { }
```

패키지: sec12.exam03_import.kumho

【 BigWidthTire.java 】

```
01 package sec12.exam03_import.kumho;
02
03 public class BigWidthTire { }
```

【 Tire.java 】

```
01 package sec12.exam03_import.kumho;
02
03 public class Tire { }
```

패키지: sec12.exam03_import.hyundai

【 Engine.java 】

```
01 package sec12.exam03_import.hyundai;
02
03 public class Engine { }
```

Tire 클래스가 두 패키지에
모두 들어가 있다.