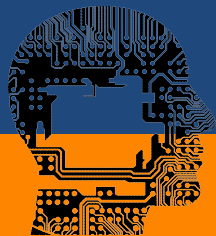




한국IT진흥부설

정보보호교육학원

아이섹



# 자바 프로그래밍 기초

## 패키지와 접근제한자

By SoonGu Hong(Kokono)

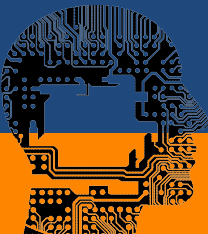


한국IT진흥부설

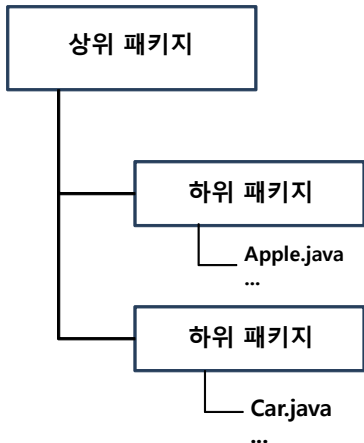
정보보호교육학원 아이섹

# 자바 프로그래밍 기초

## 1. 패키지



## 1-1-1. 패키지란?

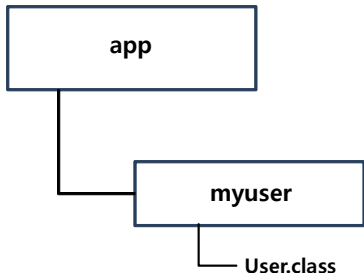


- 프로그램을 개발하다보면 수많은 클래스들을 작성하게 됩니다.

- 클래스들을 체계적으로 관리하지 않으면 클래스 간 관계가 뒤엉켜 유지보수가 난해한 프로그램이 됩니다.

- 우리가 파일들을 폴더에 관리하듯 자바에서는 클래스들을 관리하기 위해 패키지를 사용합니다.

## 1-1-2. 패키지란?



- 예를들어 app패키지 아래 myuser패키지 아래 User클래스가 존재한다면 해당 클래스의 이름은 User가 아닌 **app.myuser.User**가 됩니다.

- 실제 파일시스템에서는 app폴더 밑에 myuser폴더 밑에 User클래스파일이 위치하게 됩니다.

## 1-2. 패키지 생성 방법

- 패키지는 클래스를 컴파일하는 과정에서 자동으로 생성되는 폴더입니다. 컴파일러는 클래스에 포함된 패키지 선언을 보고 파일시스템의 폴더로 자동 생성시킵니다.

```
package 상위패키지.하위패키지;
```

```
public class 클래스이름 { ... }
```

- 패키지 이름은 개발자가 임의로 지으면 되지만 몇가지 규칙이 있습니다.

1. 숫자로 시작 불가, 특수문자 사용 불가(\$, \_ 제외).
2. 최상위 패키지는 java로 시작할 수 없음.
3. 모두 소문자로 작성하는 것이 관례.

## 1-3-1. import문

- 같은 패키지에 속하는 클래스들은 아무런 조건 없이 다른 클래스를 사용할 수 있지만 다른 패키지에 속한 클래스를 사용하려면 2가지 방법중에 하나를 선택해야 합니다. 예를 들면 app.myuser패키지 에 소속된 User클래스를 다른 패키지 app.board에 있는 Board클래스에서 사용하는 첫번째 예시입니다.

```
package app.board;
```

```
public class Board {
```

```
    app.myuser.User user = new app.myuser.User();
```

```
}
```

타입

변수명

생성자

## 1-3-2. import문

- 이전 장에서 소개한 방법은 패키지 이름이 짧다면 불편함이 없으나 패키지 이름이 길거나, 상하위 구조가 더 복잡할 경우 사용할 클래스에 전체 패키지명을 밝히는 방법은 코드를 복잡하게 보이게 합니다. 그래서 import키워드를 사용한 2번째 방법을 주로 사용합니다.

```
package app.board;

import app.myuser.User;

public class Board {

    User user = new User();

}
```

### 1-3-3. import문

- 만일 다른 특정 패키지의 클래스를 여러개 사용한다면 import구문이 많아질 것입니다. 이럴 때에는 \*을 사용하여 import를 처리합니다.
- app.myuser패키지에 User, Auth, Login 클래스가 있다고 가정합니다.

```
package app.board;  
  
import app.myuser.*;  
  
public class Board {  
    User user = new User();  
    Auth auth = new Auth();  
    Login userLogin = new Login();  
}
```

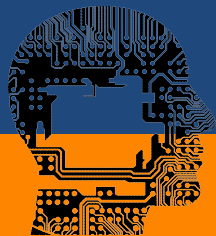




한국IT진흥부설

정보보호교육학원

아이섹

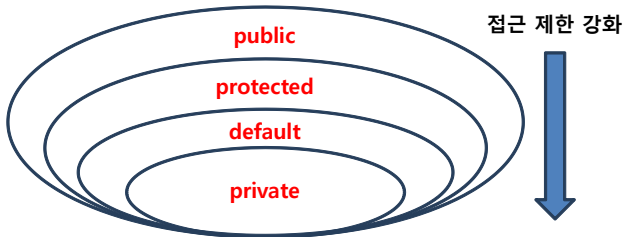


# 자바 프로그래밍 기초

## 2. 접근제한자

## 2-1. 접근제한자(Access Modifier)

- main()이 포함되지 않은 대부분의 클래스들을 외부 클래스에서 사용할 목적으로 설계된 라이브러리 클래스입니다.
- 라이브러리 클래스를 설계할 때는 외부 클래스에서 접근할 수 있는 멤버와 접근할 수 없는 멤버로 구분해서 필드, 생성자, 메서드를 설계하는 것이 좋습니다.



## 2-2. 접근제한자의 종류

Modifier	Same Class	Same Package	Other Package & Subclass	Other Package (Universe)
public	Yes	Yes	Yes	Yes
protected	Yes	Yes	Yes	No
default	Yes	Yes	No	No
private	Yes	No	No	No

- 접근제한자를 붙이지 않으면 default제한으로 설정됩니다.
- 클래스에는 public과 default만 사용가능합니다.
- 필드, 메서드, 생성자에는 4가지 제한자를 모두 사용할 수 있습니다.

**감사합니다**  
**THANK YOU**