

10. 지정자(modifier; 한정자, 제한자, 수정자)

자바에서 지정자는 크게 다음 두 종류로 구분할 수 있다. 이들은 조합하여 사용할 수 있다.

접근 지정자 : public, protected, private

기타 지정자 : final, abstract, static, native, transient, synchronized, volatile

- 접근 지정자 : 클래스, 속성, 메서드 등에 대해 접근을 어떻게 허용할 것인지를 결정
- 기타 지정자 : 선언하는 대상들의 특성을 결정한다.

◆ Java에서 클래스 멤버의 접근지정자에 따른 다른 클래스에서 접근 가능 유무

접근지정자	같은 패키지에 있는		다른 패키지에 있는	
	자식클래스	일반클래스	자식클래스	일반클래스
private(전용)	×	×	×	×
생략(default)	○	○	×	×
protected(보호)	○	○	○	×
public(공용)	○	○	○	○

private : private으로 선언된 것은 선언된 클래스 내에서만 접근 가능(외부 접근 불가)

default : 지정자를 생략한 경우, 같은 패키지에서 정의된 클래스에서 접근 가능하다.

protected : 같은 패키지에서 정의된 클래스 및 모든 자식클래스에서 접근 가능하다.

public : public으로 선언된 것은 모든 클래스에서 접근 가능하다.

- ① Java에서 접근지정자가 생략된 경우는 default 또는 friend 접근지정자가 선언되었다고 하며, 같은 패키지에 포함되어 있는 클래스들은 서로 접근을 허용한다.
- ② 자바에서 패키지(package)는 클래스(인터페이스 포함)들을 조직적으로 관리하기 위해 연관된 여러 클래스들을 묶어 놓은 하나의 단위를 지칭하며, 디렉터리(폴더)와 같다.
- ③ C++에서는 접근지정자를 생략하면 'private'이 생략된 것이다.(Java와 차이점)
- ④ 특정 패키지에 있는 클래스를 사용하려면 import문을 사용해야 한다.

`import java.lang.System;` → lang 패키지의 'System 클래스'를 사용하겠다는 뜻

↳ 그런데, 자바에서 이 패키지는 유일하게 생략할 수 있다.

◆ 접근지정자 사용 프로그래밍 예제

다음은 자바에서 접근지정자 사용에 대한 프로그램이다.

```
import java.lang.System; //생략 가능(자바에서 유일하게 생략할 수 있는 패키지)
class A
{
    private int a = 1;      //A 클래스 내에서만 접근 가능
    int b = 2;             //접근지정자 생략(default), 같은 패키지 내에서는 접근 가능
    protected int c = 3;  //모든 하위클래스에서 접근 가능(패키지 내부, 외부 무관)
    public int d = 4;      //모든 클래스에서 접근 가능(패키지 내부, 외부 무관)
    public int get()
    {
        return a;
    }
}
public class Test
{
    public static void main(String args[])
    {
        A p = new A();

        //System.out.println(p.a);    //실행시키면, 오류 발생
        System.out.println(p.get());  //출력 1
        System.out.println(p.b);     //출력 2
        System.out.println(p.c);     //출력 3
        System.out.println(p.d);     //출력 4
    }
}
```

[Tip 1] Java에서는 하나의 원시코드에는 하나의 클래스를 두는 것이 원칙이다. 여러 개가 있어도 번역하게 되면 컴파일러는 각 클래스를 각 파일(*.class)로 분리 저장시킨다. 만약, 여러 개의 클래스가 있는 경우는 public은 하나만 존재할 수 있다.

[Tip 2] Java에서 클래스를 정의할 때 public을 생략하면 패키지 내부에서만 서로 접근이 가능하다. (구체적인 내용은 이전 페이지를 참조)

기출문제 분석

1. 다음 자바코드를 컴파일할 때, 문법 오류가 발생하는 부분은? [2016년 지방 9급]

```

class Person
{
    private String name;
    public int age;
    public void setAge(int age){ this.age = age; }
    public String toString(){ return("name: " + this.name + ", age : " + this.age); }
}
public class PersonTest
{
    public static void main(String[] args)
    {
        Person a = new Person(); // ㉠
        a.setAge(27);           // ㉡
        a.name = "Gildong";     // ㉢
        System.out.println(a);  // ㉣
    }
}

```

- ① ㉠ ② ㉡
 ③ ㉢ ④ ㉣

☞ 자바 코드 - 문법 오류

- private String name;
 - **private**으로 선언되어 있어서 속성 name은 외부에서 직접 접근할 수 없다.
 - The field Person.name is not visible 오류 발생

정답 : ③