

**객체**

# 목차

- ✓ Chap01. 객체지향언어
- ✓ Chap02. 클래스
- ✓ Chap03. package와 import
- ✓ Chap04. 필드
- ✓ Chap05. 생성자
- ✓ Chap06. 메소드

# **Chap01.**

## **객체지향언어**

## ▶ 객체 지향 언어

### ✓ 객체 지향 언어란 ?

현실 세계는 사물이나 개념처럼 독립되고 구분되는 각각의 객체로 이루어져 있으며, 발생하는 모든 사건들은 객체간의 상호작용이다.  
이 개념을 컴퓨터로 옮겨 놓아 만들어낸 것이 객체지향 언어이다.

### ✓ 자바에서 객체(Object)란 ?

컴퓨터가 인식하게 클래스를 통한 new 연산자를 통해 메모리 영역에 생성한 것

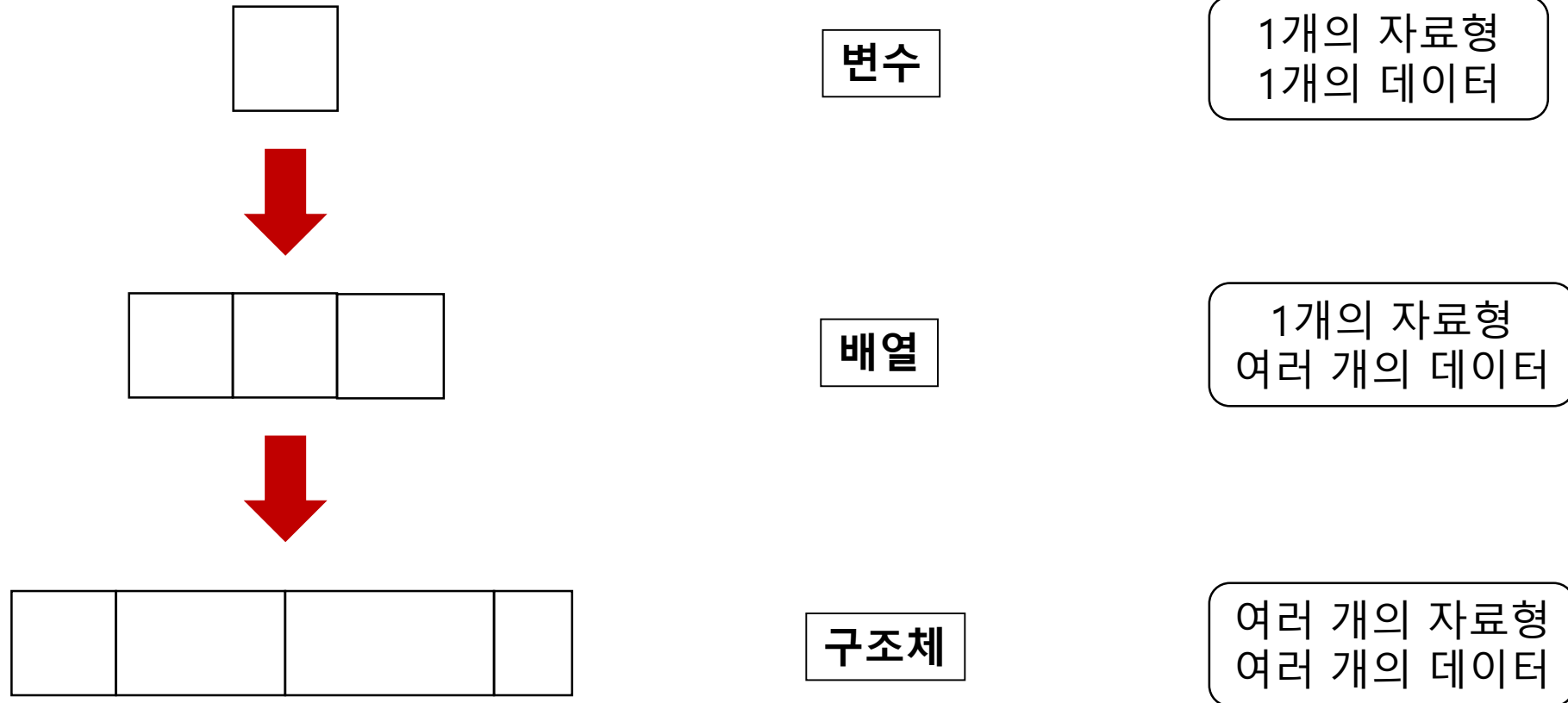
## ▶ 객체 지향 언어 - 클래스

### ✓ 클래스

메모리 영역에 생성 할 객체를 찍어내는 설계도, 붕어빵 틀  
객체의 특성에 대한 정의를 한 것으로 캡슐화를 통해 기능을 포함한 개념,  
프로그램에 필요한 사물이나 개념의 공통 요소를 추상화(abstraction)하여 정의  
ex) 회원, 자동차, 도서

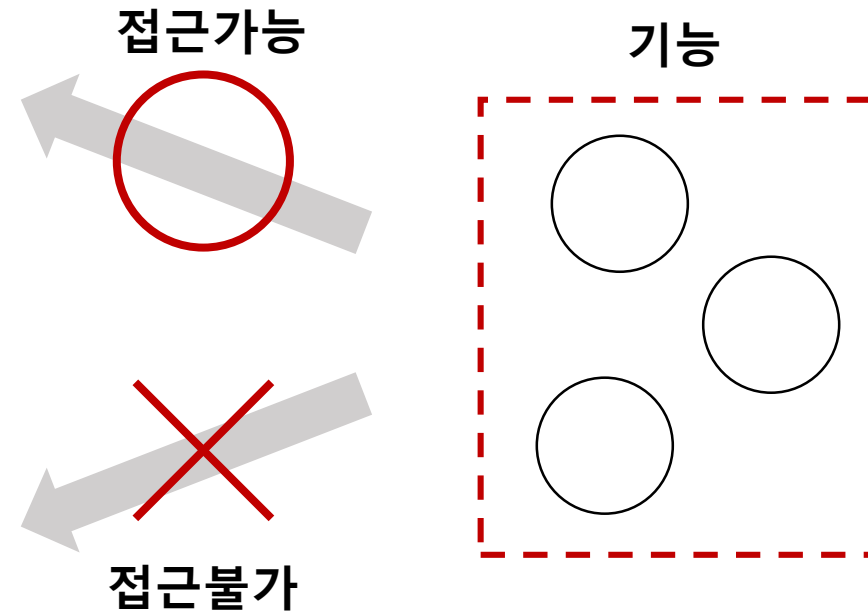
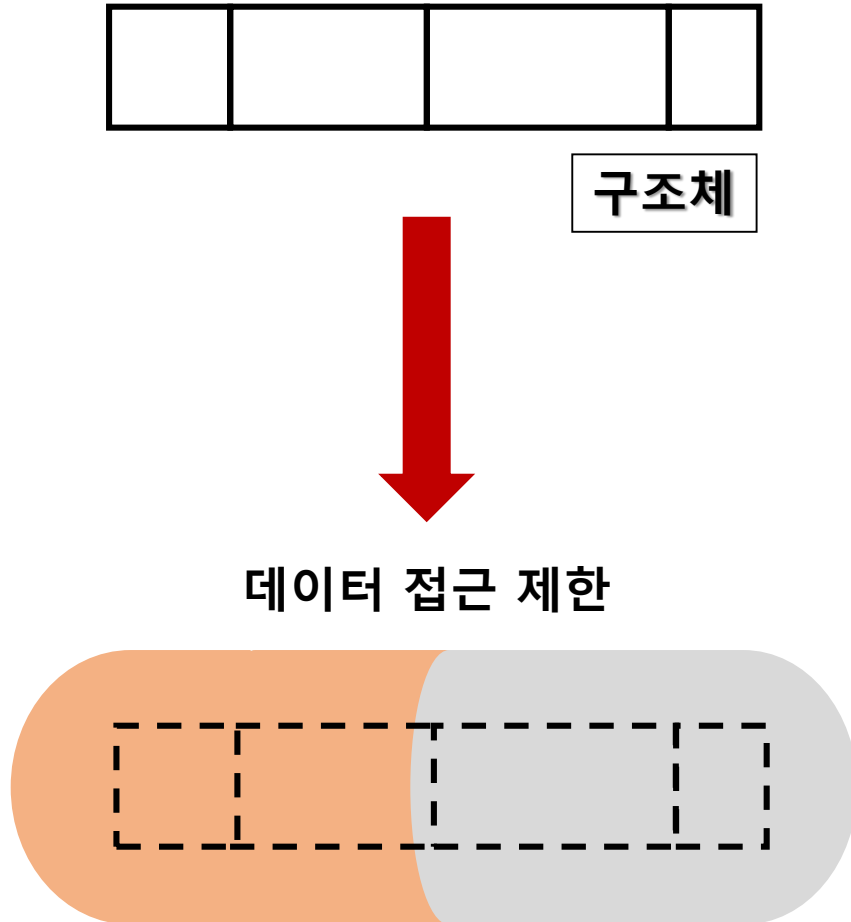
# ▶ 객체 지향 언어 - 클래스

## ✓ 클래스의 등장 배경



# ▶ 객체 지향 언어 - 클래스

## ✓ 클래스의 등장 배경



클래스 내부의 데이터를  
연산 처리하는 기능을 포함하게 됨

## ▶ 객체 지향 언어 - 추상화

유연성을 확보하기 위해 구체적인 것은 제거한다는 의미

프로그램에서 필요한 공통점을 추출하고, 불필요한 공통점을 제거하는 과정

ex) 국가에서 국민 정보 관리용 프로그램을 만들려고 할 때,

프로그램에서 요구되는 "국민 한 사람"의 정보를 추상화 한다면?





## ▶ 객체 지향 언어 - 추상화

### ✓ 추상화(abstraction) 예시

앞 페이지에서 추상화한 결과물을 객체 지향 프로그래밍 언어를 사용해서  
변수명(데이터 이름)과 자료형(데이터 타입) 정리

항목	변수명	자료형(type)
주민등록번호	pNo	String
이름	name	String
성별	gender	char
주소	address	String
전화번호	phone	String
나이	age	int

## ▶ 객체 지향 언어

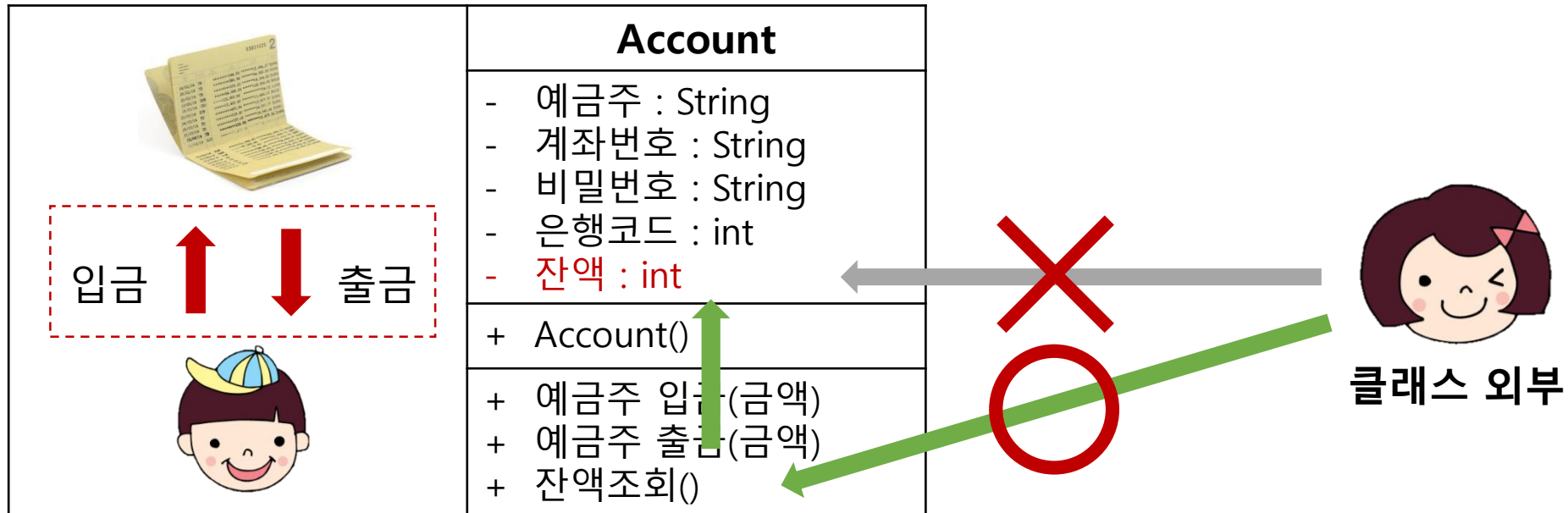
### ✓ 캡슐화란 ?

추상화를 통해 정리된 데이터들과 기능을 하나로 묶어 관리하는 기법  
클래스의 가장 중요한 목적인 데이터의 접근제한을 원칙으로 하여  
클래스 외부에서 데이터의 직접 접근을 막고,  
대신 데이터를 처리하는 함수들을 클래스 내부에 작성하는 방법

## ▶ 객체 지향 언어

### ✓ 캡슐화 원칙

1. 클래스의 멤버 변수에 대한 접근 권한은 private을 원칙으로 한다.
2. 클래스의 멤버 변수에 대한 연산처리를 목적으로 하는 함수들을 클래스 내부에 작성한다.
3. 멤버 함수는 클래스 밖에서 접근할 수 있도록 public으로 설정한다.



Account 클래스로 생성된 김철수 학생 명의의 계좌 객체

# ▶ 객체 지향 언어

## ✓ 객체

현실에 존재하는 독립적이면서 하나로 취급되는 사물이나 개념으로  
객체 지향 언어에서 객체의 개념은 클래스에 정의된 내용대로 메모리에 할당된 결과물(Object)

### 클래스 (Class)

학생 (Student)

학생이 가지는 공통적인  
요소를 추상화 하여  
클래스를 정의

### 객체 (Instance)

인스턴스화 instantiation



김철수



김영희

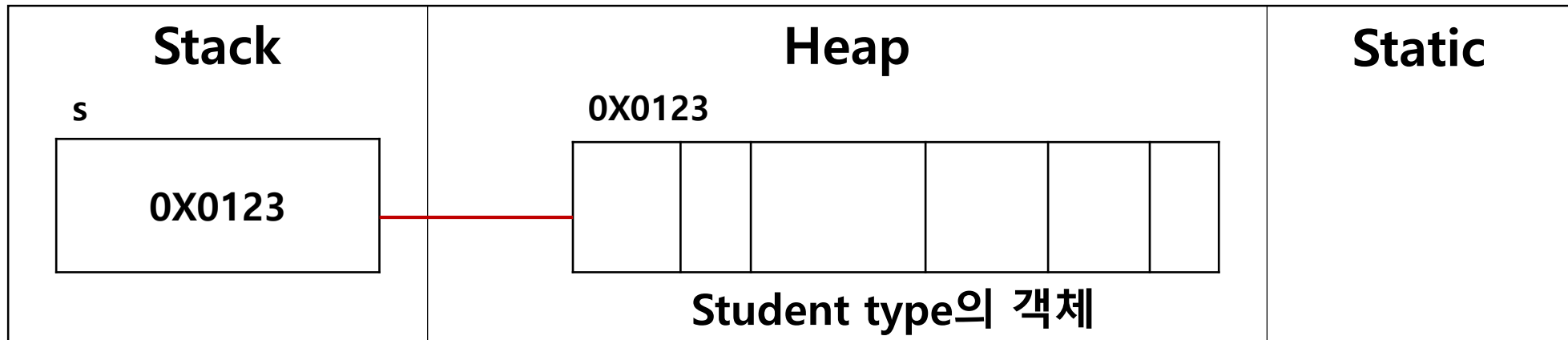
현실세계에 존재하는  
고유 객체를  
메모리에 할당

## ▶ 객체 지향 언어

### ✓ 객체(Instance)의 할당

new 연산자와 생성자를 사용하여 객체 생성 시 heap 메모리 공간에 서로 다른 자료형의 데이터가 연속으로 나열/할당된 객체 공간

예) Student s = **new** Student();



클래스



인스턴스(객체)

## ▶ 객체 지향 3대 특징

1. 캡슐화 (Encapsulation)
2. 상속 (Inheritance)
3. 다형성 (Polymorphism)

# **Chap02.**

## **클래스 (Class)**

# ▶ 클래스 (Class)

## ✓ 클래스 선언

[접근제한자] [예약어] class 클래스명 {

[접근제한자] [예약어] 자료형 변수명;  
[접근제한자] [예약어] 자료형 변수명;

속성값  
설정

[접근제한자] 생성자명() { }

[접근제한자] 반환형 메소드명(매개변수) {  
    // 기능 정의  
}

기능정의  
설정

}



## ▶ 클래스 (Class)

### ✓ 클래스 접근제한자

구분		같은 패키지 내	전체
+	public	○	○
-	(default)	○	

예)

```
public class 클래스명 {  
    // .....  
}  
class 클래스명 {  
    // .....  
}
```

## ▶ 클래스 (Class)

✓ 클래스 예시

```
public class Member {  
    private String name;  
    private int age;  
  
    public Member() {}  
  
    public String getName() {  
        return name;  
    }  
    public void setName(String name) {  
        this.name = name;  
    }  
    public int getAge() {  
        return age;  
    }  
    public void setAge(int age) {  
        this.age = age;  
    }  
}
```

# **Chap03.**

## **package와 import**

## ▶ 소스파일

### ✓ 소스파일 구성 순서

1. package문
2. import문
3. 클래스 선언

### ✓ 소스파일 구성 예시

```
package kh.academy; //package문

import java.util.Date; //import문

public class ImportTest { //클래스 선언
    public static void main(String[] args) {

    }

}
```

## ▶ package

### ✓ 패키지란 ?

서로 관련된 클래스 혹은 인터페이스의 묶음으로 패키지는 폴더와 비슷  
패키지는 서브 패키지를 가질 수 있으며, '.'으로 구분

예) Scanner 클래스의 full name은 패키지명이 포함된 `java.util.Scanner` 이다.

### ✓ 패키지의 선언

소스파일 첫 번째 문장에 단 한번 선언하며  
하나의 소스파일에 둘 이상의 클래스가 포함된 경우, 모두 같은 패키지에 속함  
모든 클래스는 하나의 패키지에 속하며, 패키지가 선언되지 않은 클래스는 자동적으로  
이름없는 패키지(default)에 속하게 됨

예) `package java.util;`

# ▶ import

## ✓ import란 ?

사용할 클래스가 속한 패키지를 지정하는데 사용  
import문을 사용하면 클래스를 사용할 때 패키지 명 생략 가능  
java.lang 패키지의 클래스는 import하지 않고도 사용 가능

\* java.lang 패키지 내의 클래스 → String, Object, System....

## ✓ import문의 선언

import문은 패키지문과 클래스 선언의 사이에 선언함.  
지정된 패키지에 포함된 클래스는 import 가능하지만 서브 패키지에 속한 모든 클래스까지는 불가능

예) `import java.util.Date;`  
      `import java.util.*;`                   // java.util 패키지 내의 모든 클래스 (단, 서브클래스는 X)  
      `import java.*;`                    // 불가능

# ▶ import

## ✓ import문 주의사항

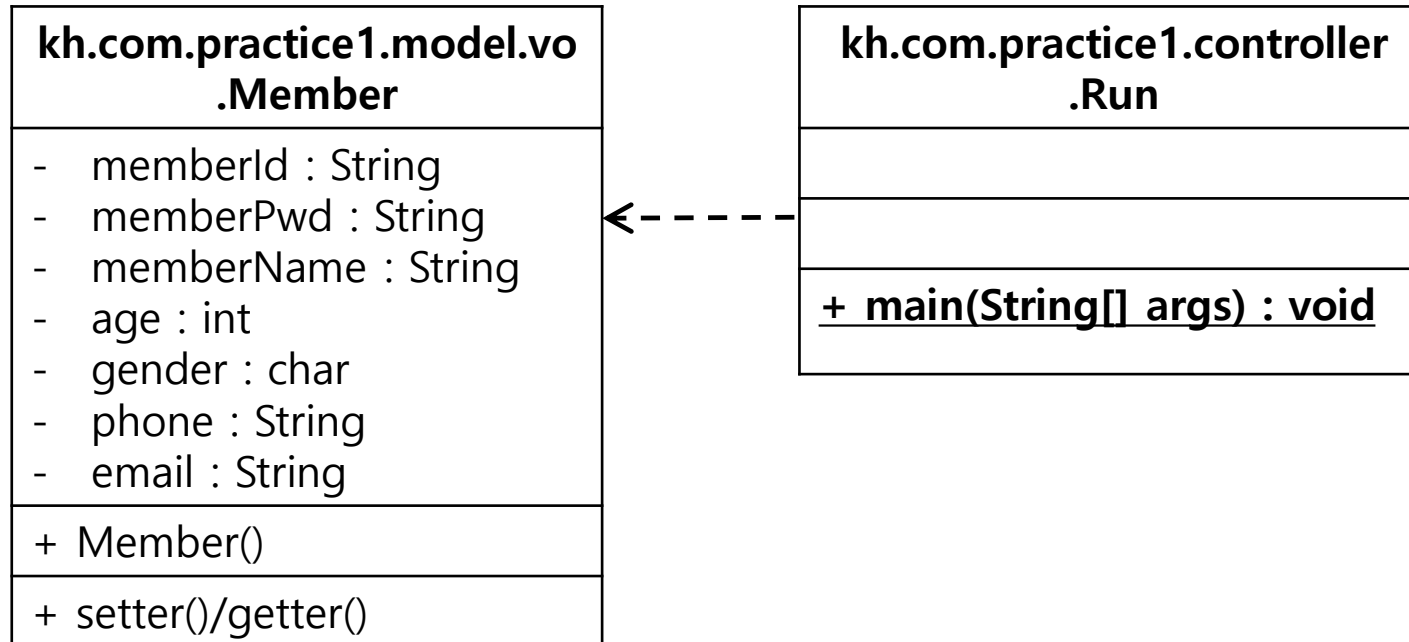
이름이 같은 클래스가 속한 두 패키지를 import 할 때는 클래스 앞에 패키지 명을 붙여 구분해 주어야 한다.

예) `package kh.academy;`

```
import java.sql.Date;
```

```
public class ImportTest {  
    public static void main(String[] args) {  
        java.util.Date today = new java.util.Date();  
    }  
}
```

## ▶ 실습문제1



package	class	method	설명
kh.com.practice1.controller	Run	<u>main(args:String[] ) : void</u>	실행용 메소드 기본 생성자로 객체를 만들고 Setter를 이용해 값 변경 후 Getter를 이용해 출력