JAVA Programming

Basic Class Concepts

BeomSeok Kim
Department of Computer Engineering
KyungHee University
passion0822@khu.ac.kr

Contents

- Introduction to class
- Understanding structure of class
- Package and class

Introduction to class

- 절차지향 vs. 객체지향 프로그래밍 언어
  - ✓ 절차지향 프로그래밍 언어
    - ➢ 프로시저(Procedure) :
      - – 프로그램 처리 절차
      - – 루틴, 하위(서브) 프로그램, 함수라고 불림
    - ➢ 프로시저의 호출 개념이 기반
      - – 내부는 *순차적*으로 처리할 수 있도록 구성
  - ✓ 객체지향 프로그래밍 언어
    - ➢ 애플리케이션을 독립된 *객체(Object)* 의 모임으로 보고 이들 간의 상호작용을 정의해 애플리케이션을 설계하는 방법

[표 2-1] 객체지향과 절차지향 프로그래밍 언어의 주요 요소 비교

구분	객체지향 프로그래밍 언어	절차지향 프로그래밍 언어
호출 단위	메소드	함수
처리 단위	객체	모듈
데이터를 저장하는 곳	속성	변수
확장	라이브러리, 상속, 추상 클래스, 인터페이스	라이브러리

Introduction to class

- 절차지향 vs. 객체지향 프로그래밍 언어
  - ✓ 객체를 무엇으로, 또 어떻게 구현해야 할까?
    - ➢ 객체는 구체적인 사물을 지칭하는 단어, 세상에 있는 모든 것은 객체가 될 수 있다.

[그림 2-1] 자전거의 구조

- ➢ 각 부품들이 객체에 해당함.
  (안장, 프레임, 핸들, 앞바퀴, 체인, 뒷바퀴 등)

Introduction to class

- 절차지향 vs. 객체지향 프로그래밍 언어
  - ✓ 클래스를 사용하여 자전거부품 객체를 도식화

[그림 2-2] 자전거 부품의 클래스 표기

Introduction to class

- 클래스(Class)란
  - ✓ 속성이 동일한 객체들을 포괄하는 상위 개념
  - ✓ 클래스는 객체를 만들어 내는 틀
  - ✓ 클래스는 객체를 표현하는 추상 데이터 타입(Abstract Data Type)

Understanding structure of class

- 클래스의 구성요소
  - ✓ 클래스명(Class Name)
    - ➢ 파일명을 클래스명과 일치시켜야 하는 경우: public으로 설정했을 때, main 메소드가 포함되어 있을 때
  - ✓ 속성(Attribute)이나 필드(Field)인 변수(Variable)
  - ✓ 메소드(Method)
- 추상과 추상 데이터 타입
  - ✓ 추상 : 실제로 존재하는 않음, 데이터 타입 : 값의 집합
  - ✓ 추상 데이터 타입 : 다양한 객체를 프로그램에서 사용하기 전에 미리 선언한 자료형

```
[접근 제한자][지정 예약어]class[클래스명]extends[상위 클래스]implements[상위 인터페이스] {
[Attribute 또는 Field]
내용부;
내용부;
...
[Method]
내용부;
내용부;
...
}
```

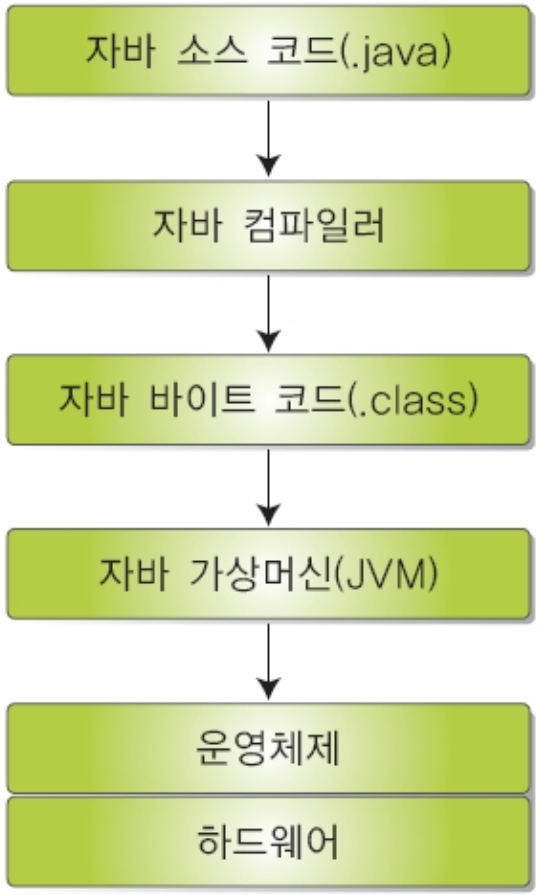
Understanding structure of class

- 클래스의 구성요소
  - ✓ 자전거 클래스를 파일명과 같게 선언한 예
    - ➢ 파일명이 Bike.java 이어야 함 .java 는 확장자

```
01 public class Bike {                          클래스명을 Bike로 지정
02
03              private byte bId;               속성과 필드변수 지정
04              private String name;
05              private int iPosX;
06              private int iPosY;
07
08              public void SetId(byte bId) {
09              …                                메소드 지정
10              }
11
12              public byte GetId() {
13              …
14              }
15
16              public void SetName(String name) {
17              …
18              }
19
20              public String GetName() {
21              …
22              }
23
24              public void Move(int iPosX, int iPosY) {
25              …
26              }
27
28 }
```

Understanding structure of class

- 자바 프로그램의 개발 단계
  1. 편집기를 이용하여 소스코드를 작성
  2. 소스코드를 컴파일함
  3. 바이트 코드를 실행(가상머신이 해석)



[그림 2-3] 자바 가상머신과 자바 소스 실행 단계

Understanding structure of class

- HelloJava.java 소스 분석
  - ✓ main()
    - ➢ 가장 기본이 되는 메소드, 코드의 진입점(Entry Point)
    - ➢ 하나의 자바 프로그램은 하나의 main()을 갖는다
      - public : 외부 객체가 객체 내에 있는 메소드에 접근할 수 있다.
      - static : 객체가 생성되지 않아도 해당 메소드를 실행할 수 있다.
      - void : 메소드를 수행한 뒤에 반환하는 값의 형태를 지정하지 않는다.
  - ✓ System.out.println()
    - ➢ 메시지를 출력하는 명령어.
      - 객체 내에 속성이나 메소드를 호출할 때 (.) 연산자를 이용함
      - 세미콜론(;) 명령문이 끝났음을 알림.

파일명과 클래스명이 같음

```
package javabook.ch01;

public class HelloJava {

        public static void main(String[] args) {
                System.out.println("Hello Java!");
        }

}
```

Package and class

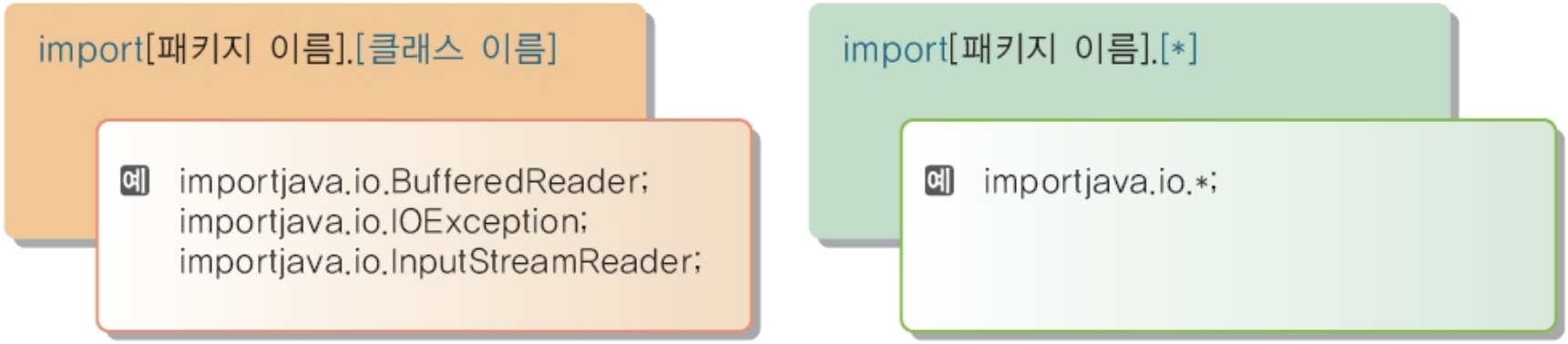
- 자바 기본 패키지
  - ✓ 서로 관련 있는 클래스나 인터페이스를 하나의 이름으로 묶은 것
  - ✓ 패키지 단위로 계층적으로 분류하여 클래스를 좀더 체계적으로 관리
  - ✓ 자바가 패키지를 사용해서 얻을 수 있는 이점
    - ➢ 다양한 클래스를 체계적으로 관리할 수 있다.
    - ➢ 클래스명을 패키지명으로 사용할 때 발생할 수 있는 혼란을 막을 수 있다.
    - ➢ 다양한 방법으로 접근 권한을 제어할 수 있다.
    - ➢ 신규 패키지를 기존 패키지에 쉽게 추가할 수 있다.

[표 2-2] 자바 라이브러리의 주요 패키지

패키지명	설명
java.applet	애플릿을 생성하는 데 필요한 클래스, 애플릿과 상호작용을 하는 애플릿 클래스와 인터페이스를 담고 있다.
java.awt	사용자 인터페이스를 생성하고, 그래픽과 이미지를 사용하는 클래스를 담고 있다.
java.awt.event	이벤트를 처리하는 인터페이스와 클래스를 담고 있다.
java.awt.image	이미지를 생성하고 수정하는 클래스를 담고 있다.
java.io	데이터 입 · 출력과 파일 시스템을 담고 있다.
java.lang	자바 프로그램 언어를 사용하는 기본 클래스를 담고 있다.
java.net	애플리케이션끼리 통신(인터넷/인트라넷)할 수 있게 해주는 클래스를 담고 있다.
java.rmi	원격 객체의 메소드를 호출하고, 분산 자바 프로그램을 제작하는 패키지와 인터페이스를 담고 있다.
java.security	보안(데이터 암호화와 복호화, 접근 권한)을 위한 클래스와 인터페이스를 담고 있다.
java.util	시간 조작, 난수 발생, 문자열 토큰화 등 유틸리티 클래스를 담고 있다.
java.util.zip	표준 ZIP, GZIP 파일 형식을 읽고 쓸 수 있는 클래스를 담고 있다.

Package and class

■ 자바 기본 패키지



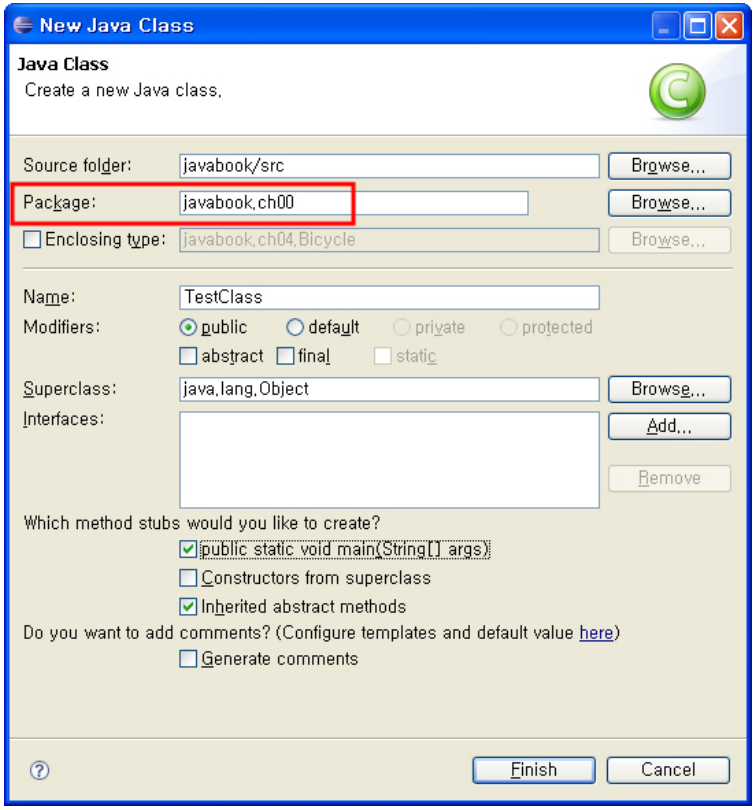
[그림 2-4] 패키지 선언방법과 예

- ✓ 패키지명과 클래스명을 모두 지정해 주어야 함
- ✓ 패키지에 있는 클래스를 와일드카드(*)를 입력.

```
import java.awt.*; // 패키지에 들어 있는 클래스를 포함할 때
import java.awt.event.*; // 패키지에 들어 있는 클래스와 하위 클래스를 모두 포함할 때
```

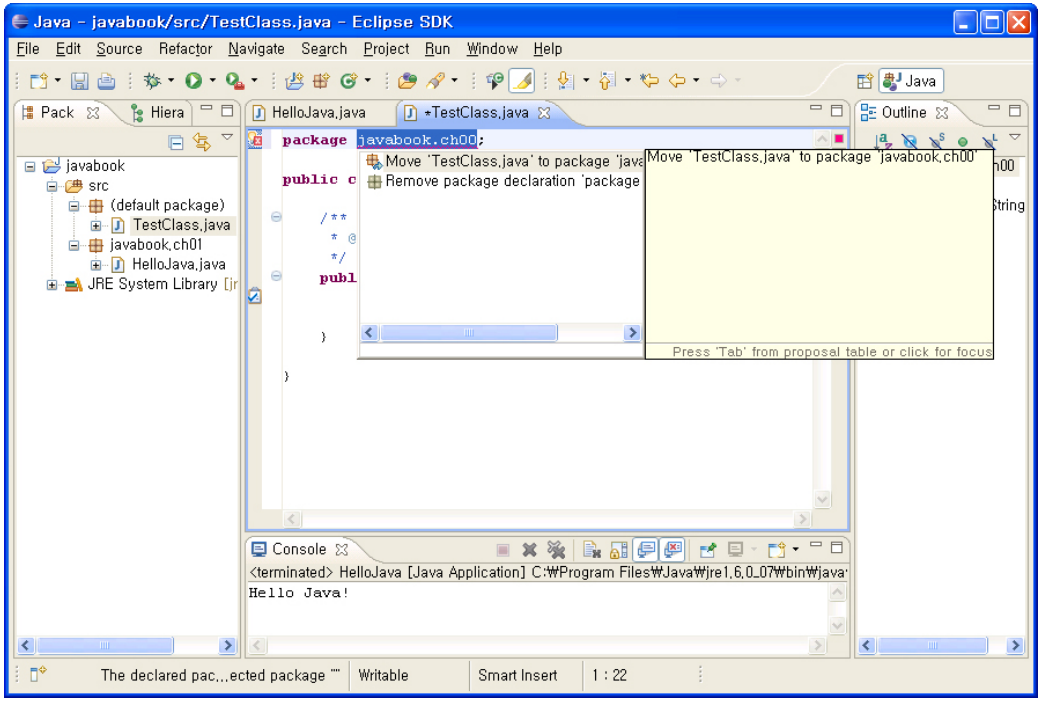
Package and class

- 사용자 정의 패키지의 패키지명 만드는 요령
  - ✓ java로 시작하거나 기타 JDK에 포함된 기본 패키지명은 피함
  - ✓ 역도메인 방식을 사용
  - ✓ 패키지명은 소문자, 클래스명 첫 글자는 대문자로 작성
  - ✓ 첫 행에‘package 패키지명;’을 넣어 줌
  - ✓ package explorer view에서 오른쪽 버튼 클릭 후 [New]-[Package]
  - ✓ 소스를 드래그 해당 패키지로 이동 가능
  - ✓ 기존 패키지명을 F2 키를 눌러 변경 가능



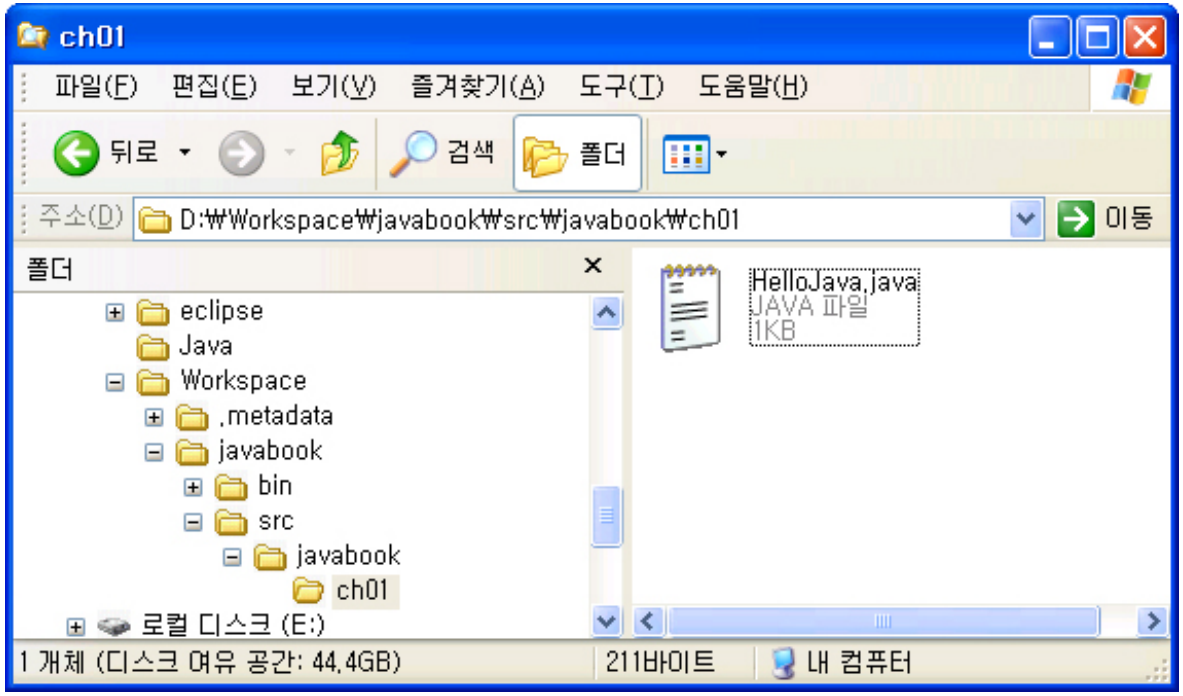
Package and class

- 사용자 정의 패키지
  - ✓ 추후에 패키지를 지정할 시, default package에 들어있는 클래스의 패키지 변경으로 오류가 발생하게 됨
  - ✓ 행 앞에 빨간점을 클릭 후 move ○○○○ class to ○○○○ package 항목을 선택



Package and class

- 사용자 정의 패키지
  - ✓ HelloJava 클래스의 소스구조 실제 폴더
  - ✓ Workspace\javabook\src\javabook\ch01 에 위치
  - ✓ 소스를 생성할 때 패키지를 지정하지 않았다면, Workspace\javabook\src에 위치



Package and class

- 클래스
  - ✓ 객체지향 프로그래밍 언어에서 기본 단위
- 클래스의 구조

```
클래스 {
  속성;
  행동;
}
```

- 자바 프로그램 개발의 3단계
  - ✓ 소스 코드 작성 -> 컴파일 -> 바이트 코드 실행
- 패키지와 클래스
  - ✓ 자바 클래스를 효율적으로 관리하는 구조
  - ✓ 패키지 구조에 따라 서브 폴더를 생성한다
  - ✓ 패키지명이 중복되는 것을 막기위한 역도메인 방법

Thank You!
Q&A