



(스마트웹&콘텐츠개발)스마트웹&콘텐츠
웹애플리케이션 개발자 양성과정



JAVA 시작하기

강 사 : 임 정 훈



학 습 내 용

1. JAVA 개요(장점, 특징)
2. 객체지향언어의 4가지 특징
3. JAVA 개발 환경 만들기(Java SE)
4. JAVA API & editor
5. Oracle 국제공인자격증



1. JAVA 개요

● 객체지향언어(Object Oriented Language)

- 자바는 완전한 객체지향 언어(1995년 JAVA 베타2 버전 탄생)
- 객체지향 특징 : 클래스, 상속, 캡슐화, 다형성 등의 개념이 잘 적용된 언어
- 객체지향 프로그래밍 : 우리가 살아가는 실제 세계와 동일한 사고방식의 프로그램
- 자바 언어는 가장 쉽게 이해할 수 있으며, 빨리 배울 수 있는 프로그래밍 언어



1. JAVA 개요

1) Java의 장점

1. . 확장성 (Scalability)

➤ 적용분야광범위

- 기업용(Enterprise급) ~ 핸드폰(Mobile)
- 다양한 분야에 적용 가능한 프로그램 개발

➤ 기존 시스템 확장용이

- 인사 시스템
- 급여 시스템



1. JAVA 개요

2. . 보안성 (Security)

- 자체 강력한 보안 능력보유
 - 클래스의 캡슐화
 - 외부접근자 : public vs private
 - 자바가상머신(JVM) 보안 제공



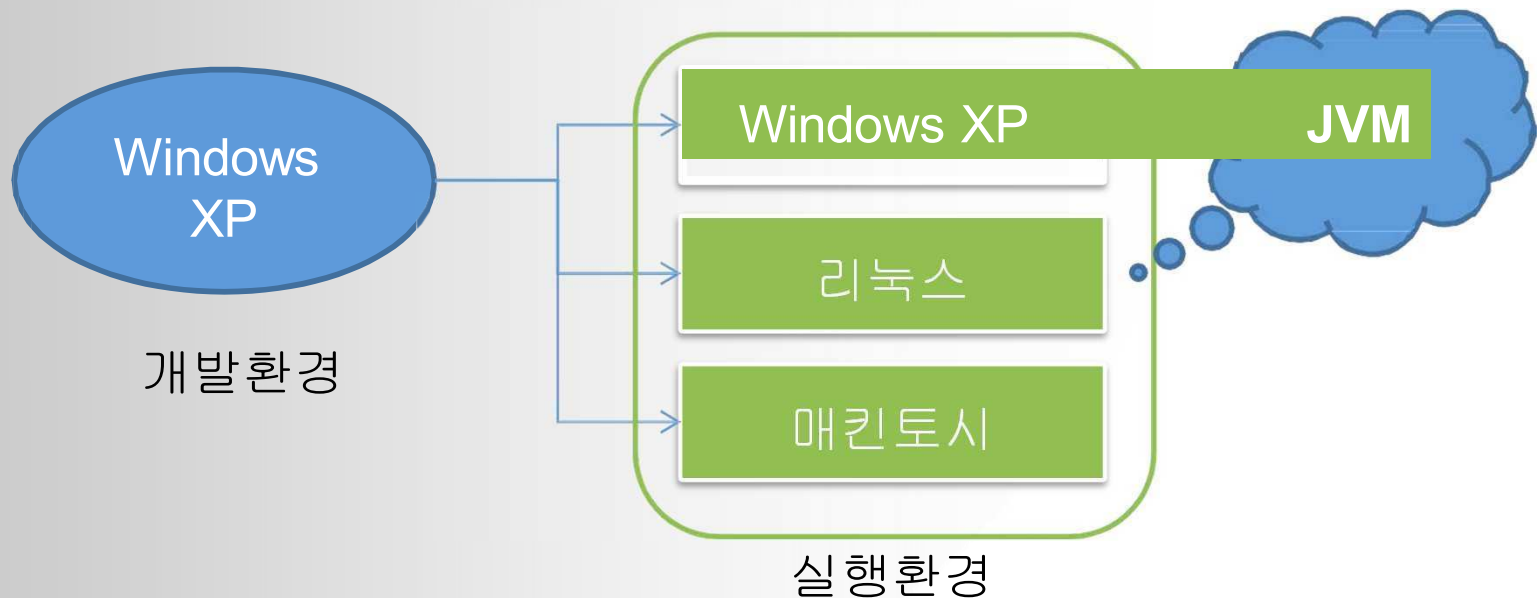


1. JAVA 개요

3. 보편성(Universality) = 이식성, 호환성(Compatible)

➤ 플랫폼(운영체제:OS) 독립성

- JVM(Java Virtual Machine) : java 프로그램 해석 & 실행 기능

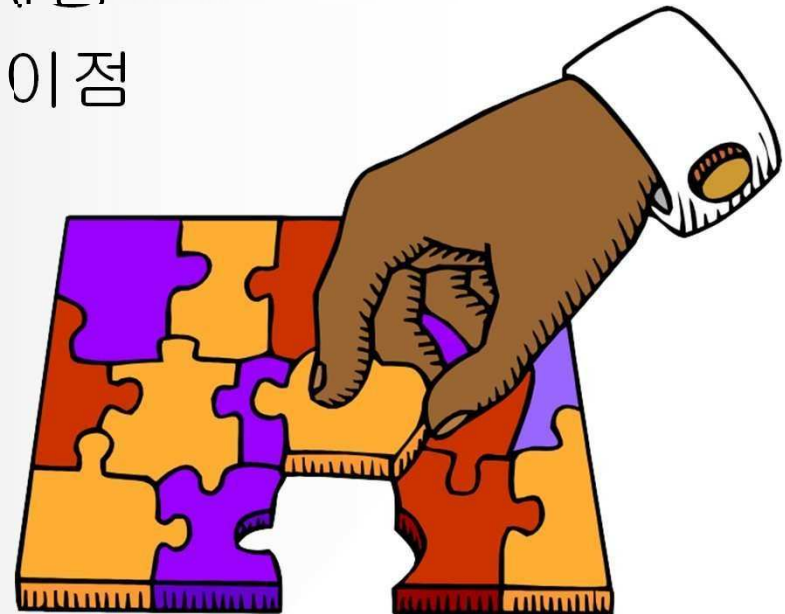




1. JAVA 개요

4. . 재사용성 (Reuse) = Modularity

- **Modularity** : 생산에 규격화된 부품을 사용함
- 기 개발된 java 프로그램 재사용
- 객체지향프로그램의 가장 큰 이점





1. JAVA 개요

2) Java의 특징

1. 쉽고, 편한 프로그램 개발 환경 제공

- 복잡한 부분 제외
 - C언어의 포인터와 다중 상속 기능 제거
- 객체지향 프로그램(Object Oriented Program)
 - CBD(Component Based Development) 개발방법론적용
 - 객체지향 개념 이해 선행
- 풍부한 API(Application Programming Interface) 제공
 - API : 프로그램 개발 시 사용될 수 있는 유용한 응용 프로그램



1. JAVA 개요

2. 플랫폼에 독립적인 환경 제공

➤ 프로그램 언어(해석기로 구분)

- 컴파일러 언어 : 베이직, 포트란, C언어
- 인터프리터 언어 : HTML, Script언어
- 컴파일러 + 인터프리터 언어 : JAVA

구분	컴파일 언어	인터프리터 언어
장점	실행 속도 빠름	OS 독립적
단점	OS 종속적	실행 속도 느림



1. JAVA 개요

플랫폼 종속성(platform dependency)



인텔 CPU를 가진
리눅스 환경에서
개발

C/C++
응용 프로그램

플랫폼 = 하드웨어 플랫폼 + 운영체제 플랫폼

프로그램의 플랫폼 호환성 없는 이유

- 기계어가 CPU마다 다름
- 운영체제마다 API 다름
- 운영체제마다 실행파일 형식 다름

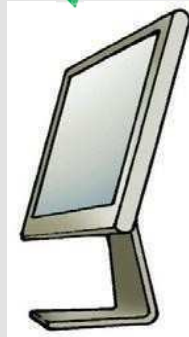
실행

실행되지
않음

실행되지
않음



인텔 CPU + 리눅스



Apple사의 MAC PC



인텔 CPU + 윈도우 노트북



1. JAVA 개요

플랫폼 독립성(platform independency)

Write Once !!



자바 응용 프로그램

Run Anywhere!!

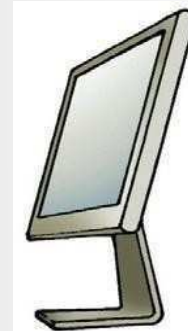
실행



자바 가상 기계

인텔 CPU + 리눅스

실행



자바 가상 기계

Apple사의 MAC PC

실행



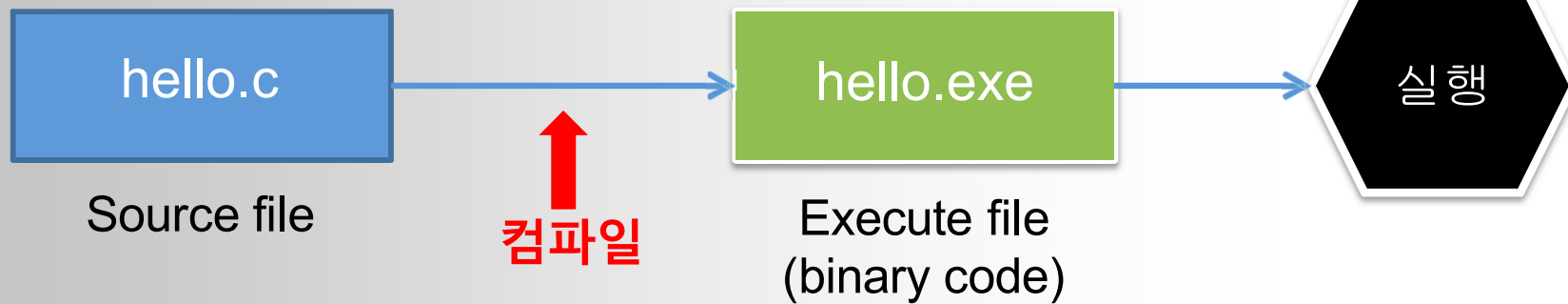
자바 가상 기계

인텔 CPU + 윈도우 노트북

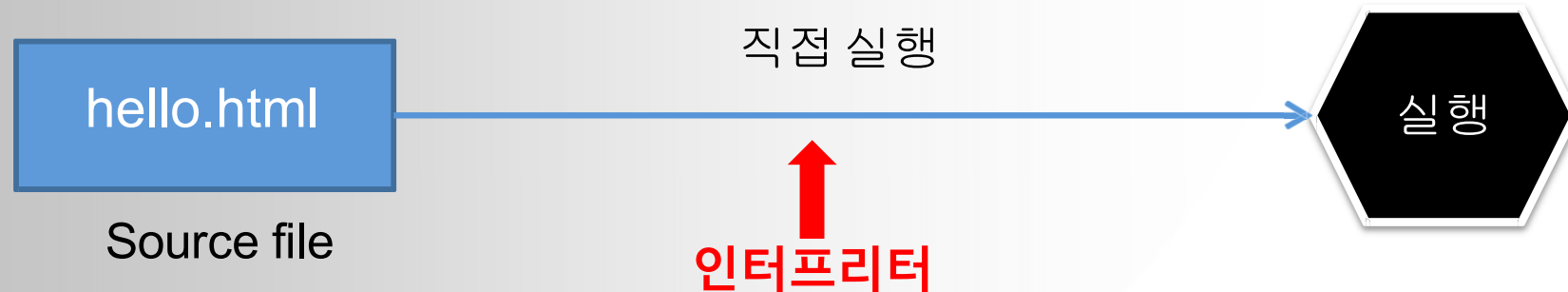


1. JAVA 개요

➤ 컴파일언어



➤ 인터프리터언어





1. JAVA 개요

❖ 컴파일(compile)이란?

- 사용자가 작성한 프로그램을 기계가 인식할 수 있는 기계어로 번역해 주는 소프트웨어

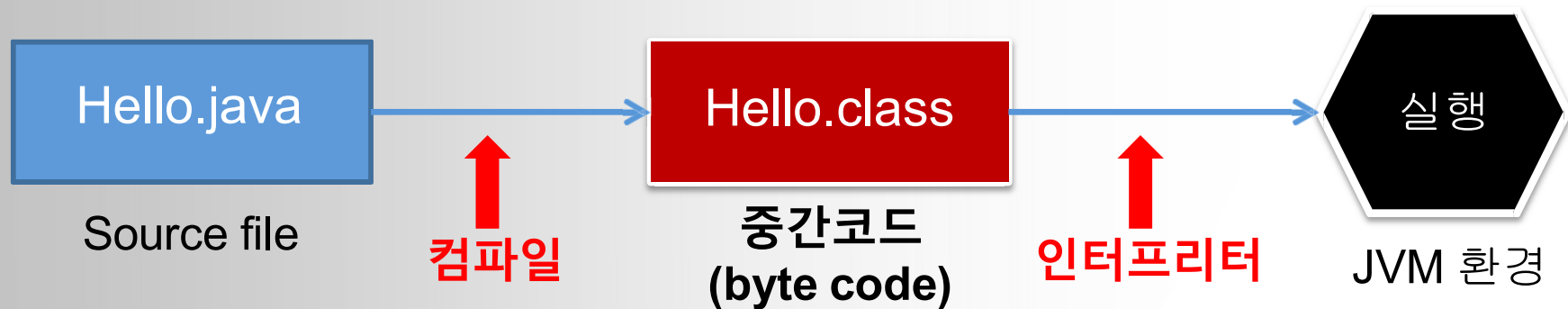




1. JAVA 개요

➤ JAVA 언어

- 컴파일 언어 장점 : 빠른 기계어 생성
- 인터프리터 언어 장점 : 플랫폼 독립성
- JVM : 바이트 코드 해석 및 실행(자체 실행불가)

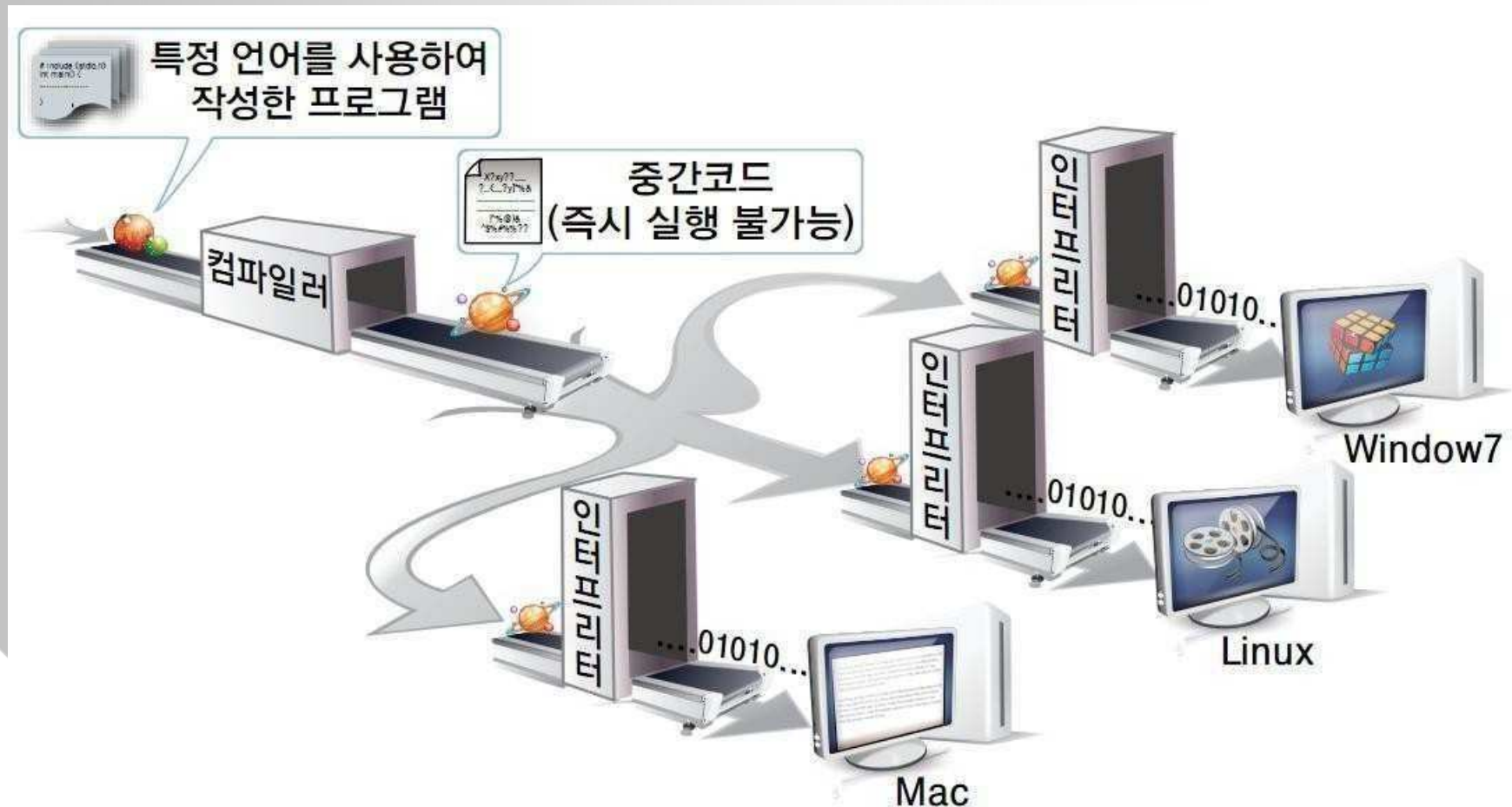




1. JAVA 개요

❖ 바이트코드란?

- 플랫폼에 독립적인 자바 코드

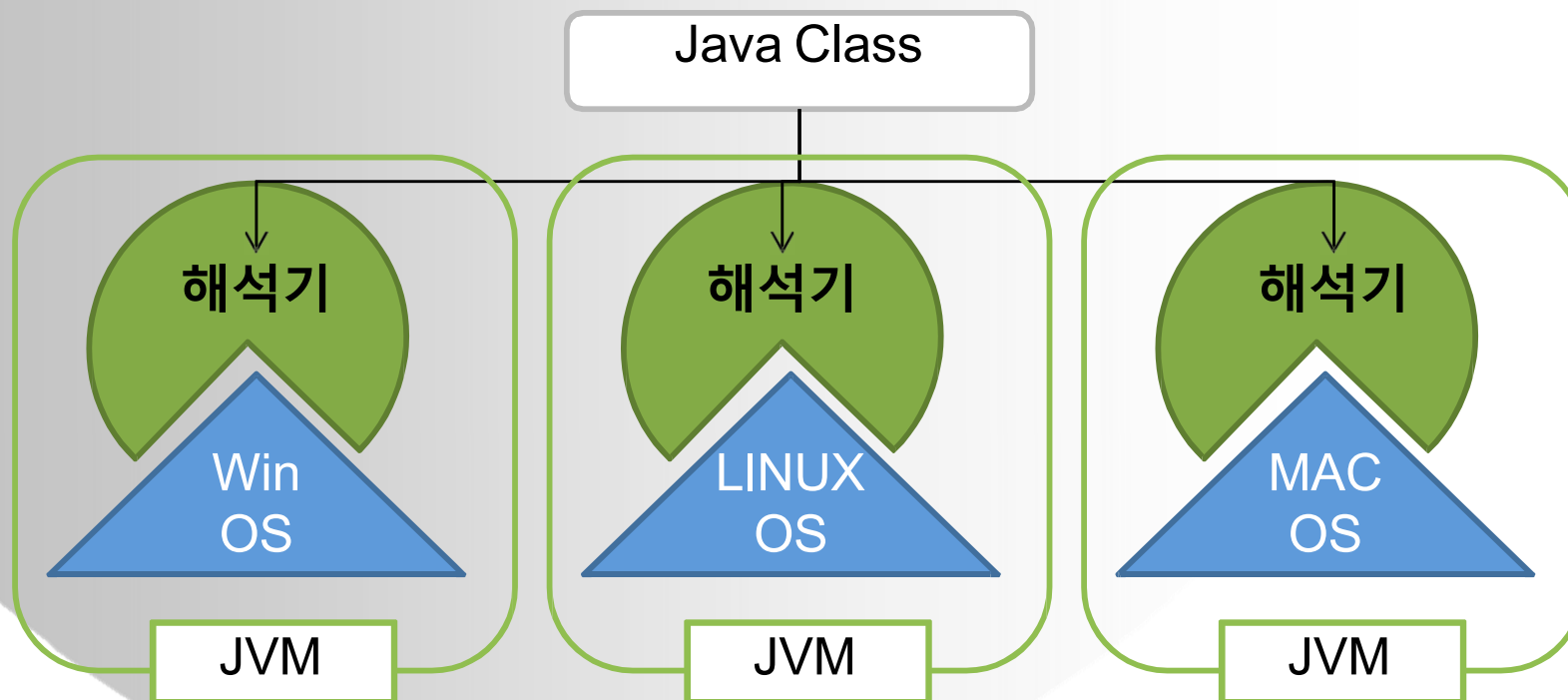




1. JAVA 개요

3. JVM(Java Virtual Machine)

- Byte code(class) 파일 자체 실행 못함
- JVM : Byte code 해석기를 포함한 시스템



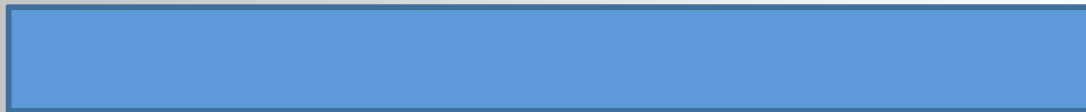


1. JAVA 개요

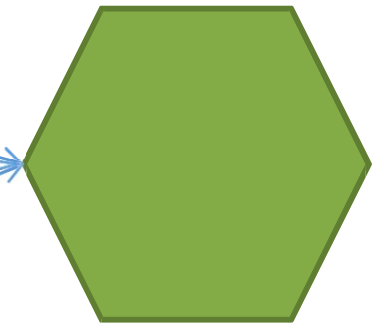
4. Thred기반 멀티태스킹

➤ Process(실행 프로그램)의 작은 모듈

프로세스



쓰레드



프로세서
(CPU)

시스템 부하
(Overhead)



1. JAVA 개요

5. Garbage Collection

- 프로그램 종료 후 사용한 메모리 시스템에 반환
- 효율적인 메모리사용
 - “메모리가 부족하다.” 메시지 해결
- 시기 : 유휴 시간(idle time)





1. JAVA 개요

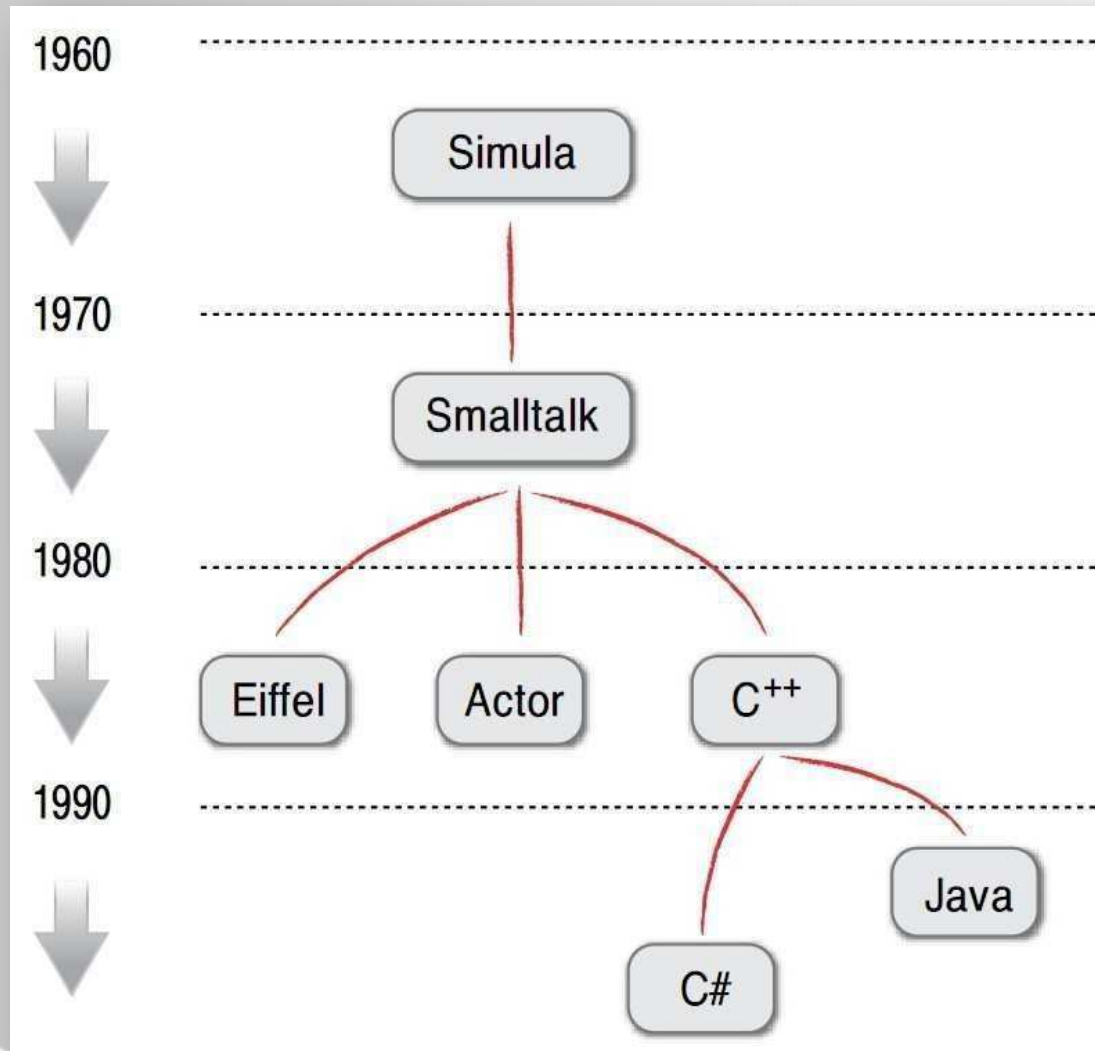
3) Java 응용분야

- 일반 사무용 소프트웨어 개발 : 급여관리, 재고관리
- 임베드디 소프트웨어 개발 : 가전제품 삽입 프로그램
- 웹 소프트웨어 개발 : 일반 웹, 서버/클라이언트 페이지 개발
- 하이브리드 웹 & 앱 컨텐츠 개발 : 안드로이드폰, 아이폰
- 엔터프라이즈 소프트웨어 개발 : 대규모 기업 프로그램 개발



2. 객체지향언어의 4가지 특징

- 객체지향 언어의 역사



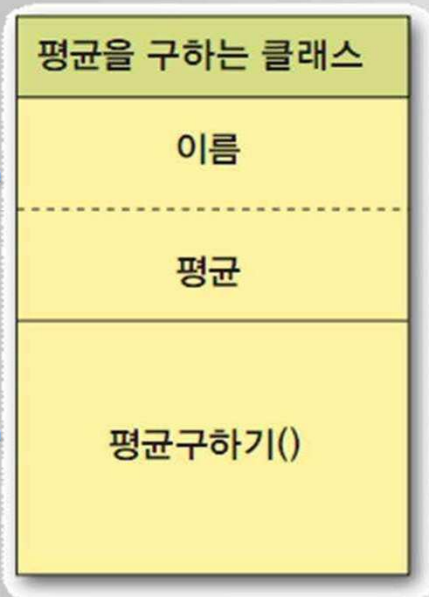


2. 객체지향언어의 4가지 특징

- 클래스와 객체

속성

기능



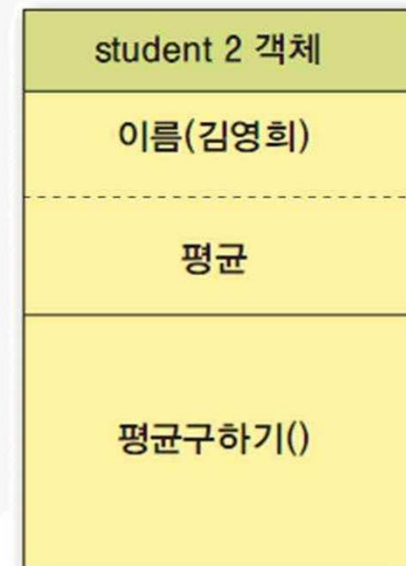
객체생성
(instantiate)

객체생성
(instantiate)



6장

인스턴스
(instance)



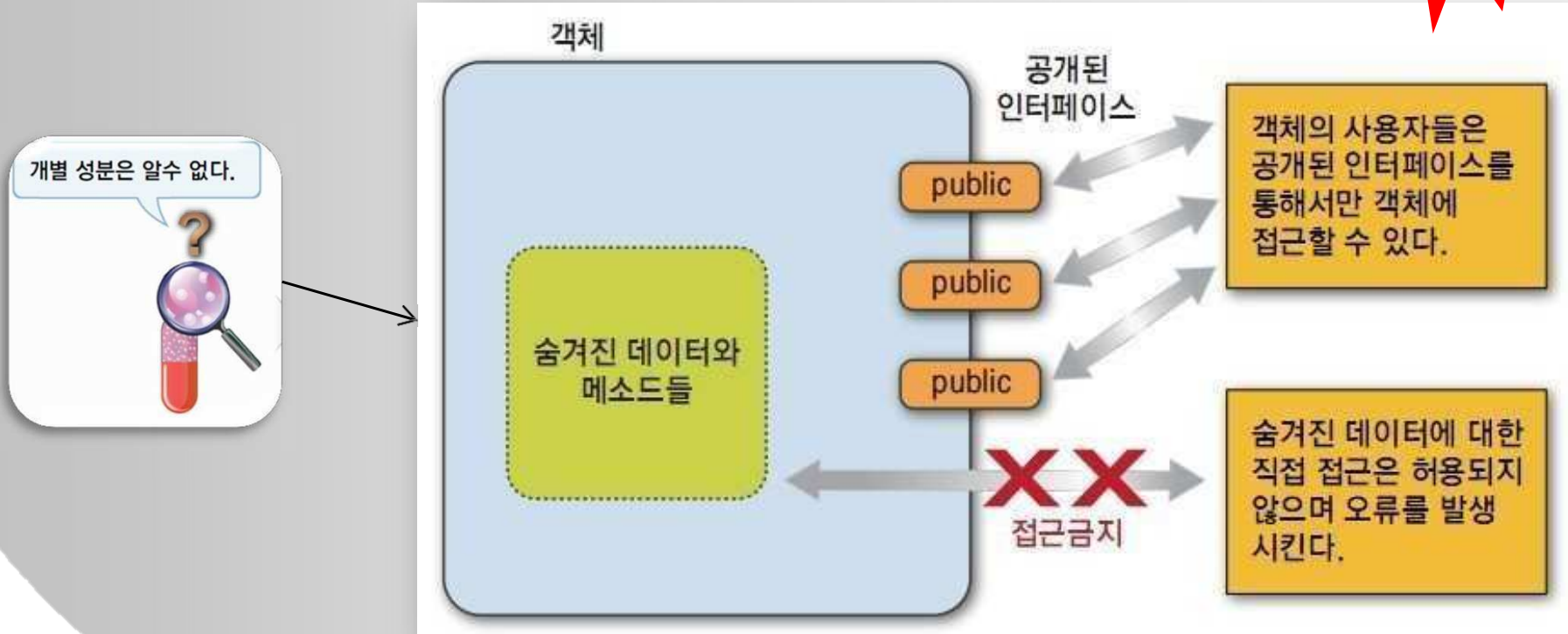


2. 객체지향언어의 4가지 특징

1) 캡슐화(Encapsulation)

- ▶ 캡슐화는 클래스 내에서 접근지정자로 구현됨

7장



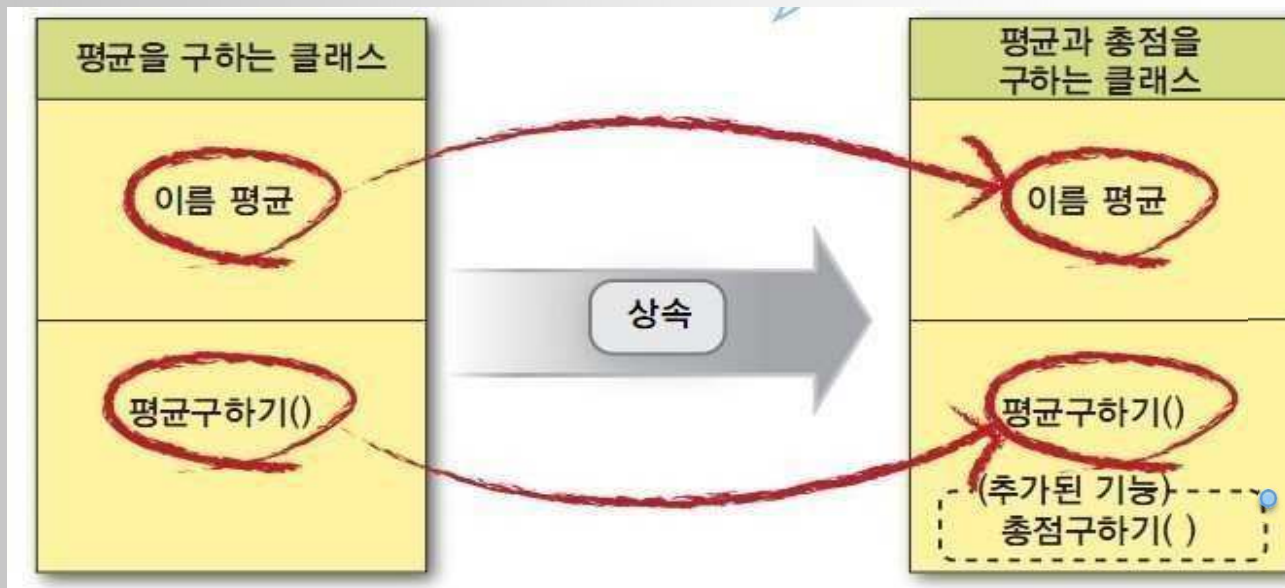


2. 객체지향언어의 4가지 특징

2) 상속(Inheritance)

- 기존 클래스로부터 모든 속성과 메소드 상속
- 필요한 속성과 메소드를 추가하여 새로운 클래스 생성

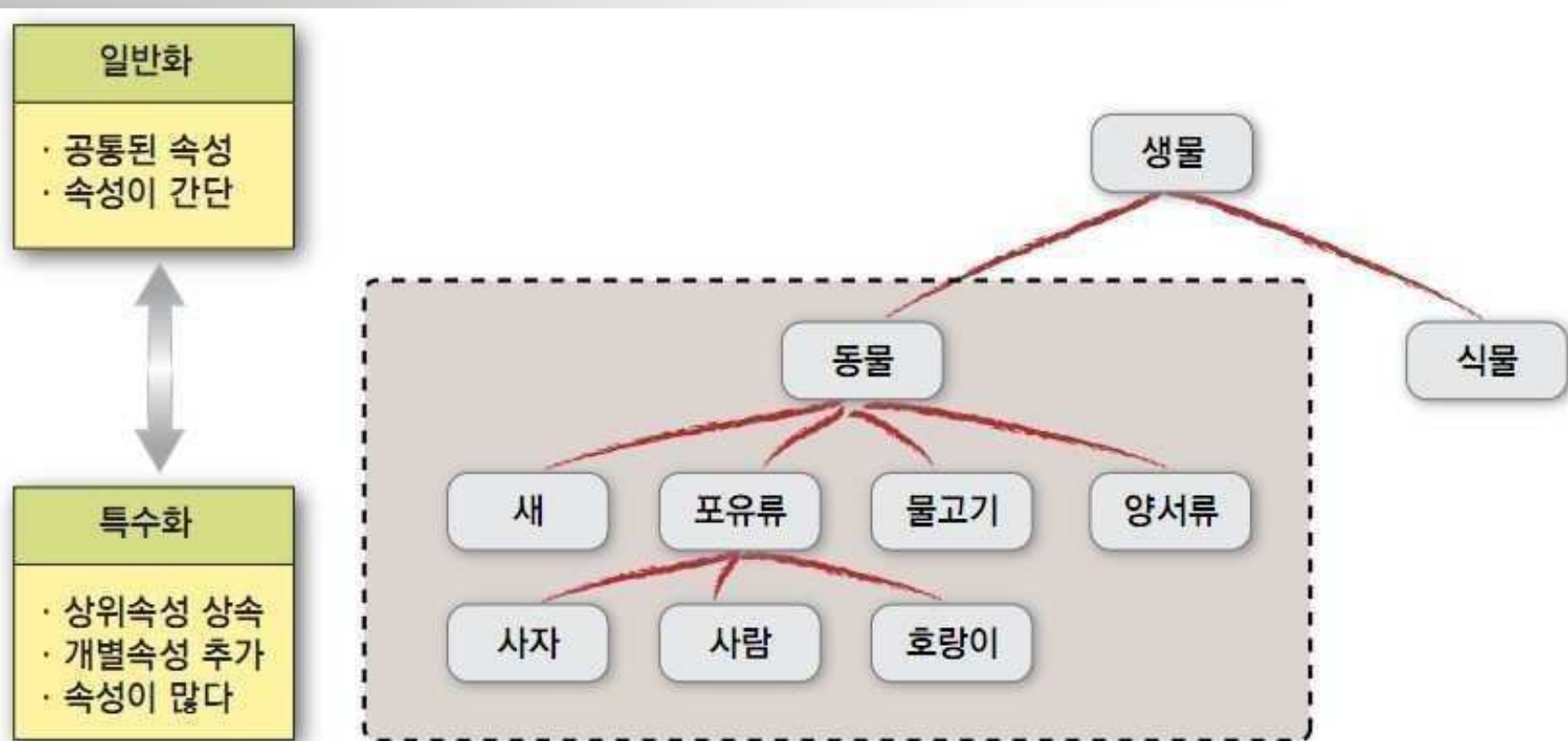
7장





2. 객체지향언어의 4가지 특징

2) 상속(Inheritance)



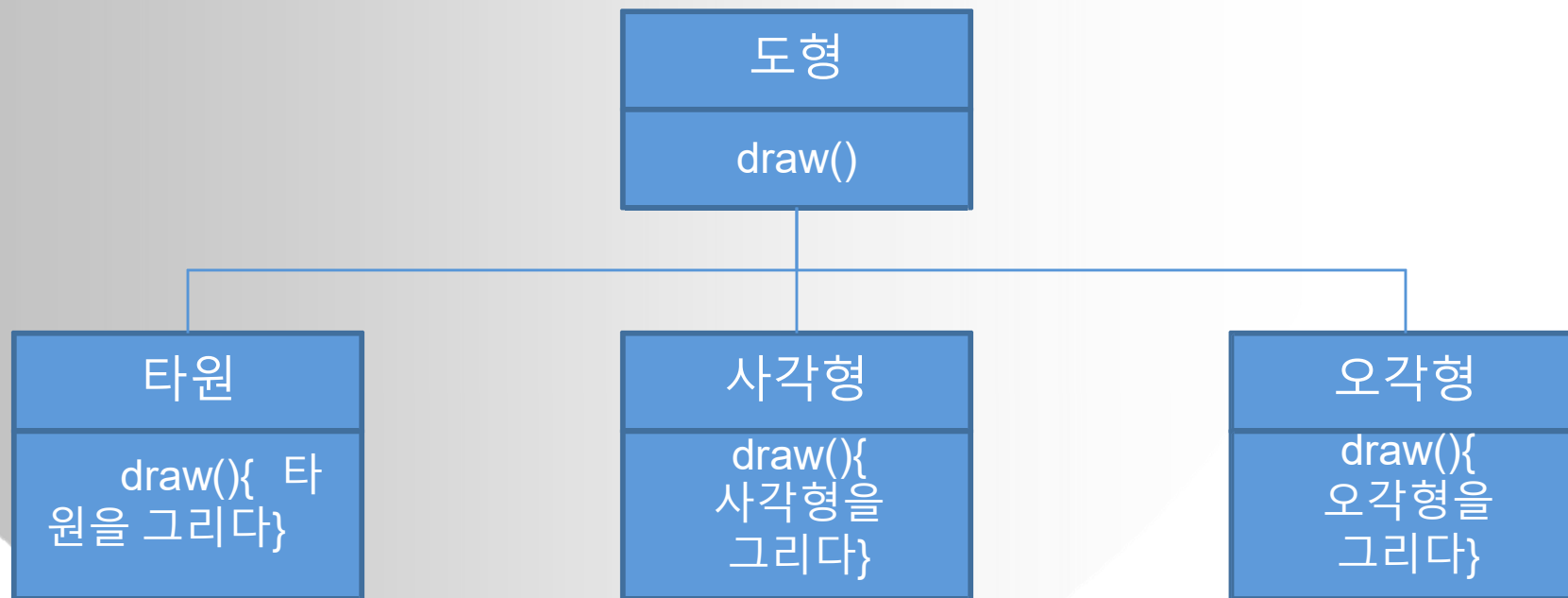


2. 객체지향언어의 4가지 특징

3) 다형성(Polymorphism)

- 서로 다른 객체가 동일한 메시지에 대하여 서로 다른 방법으로 응답할 수 있는 기능(상속을 통해서 구현됨)

7장

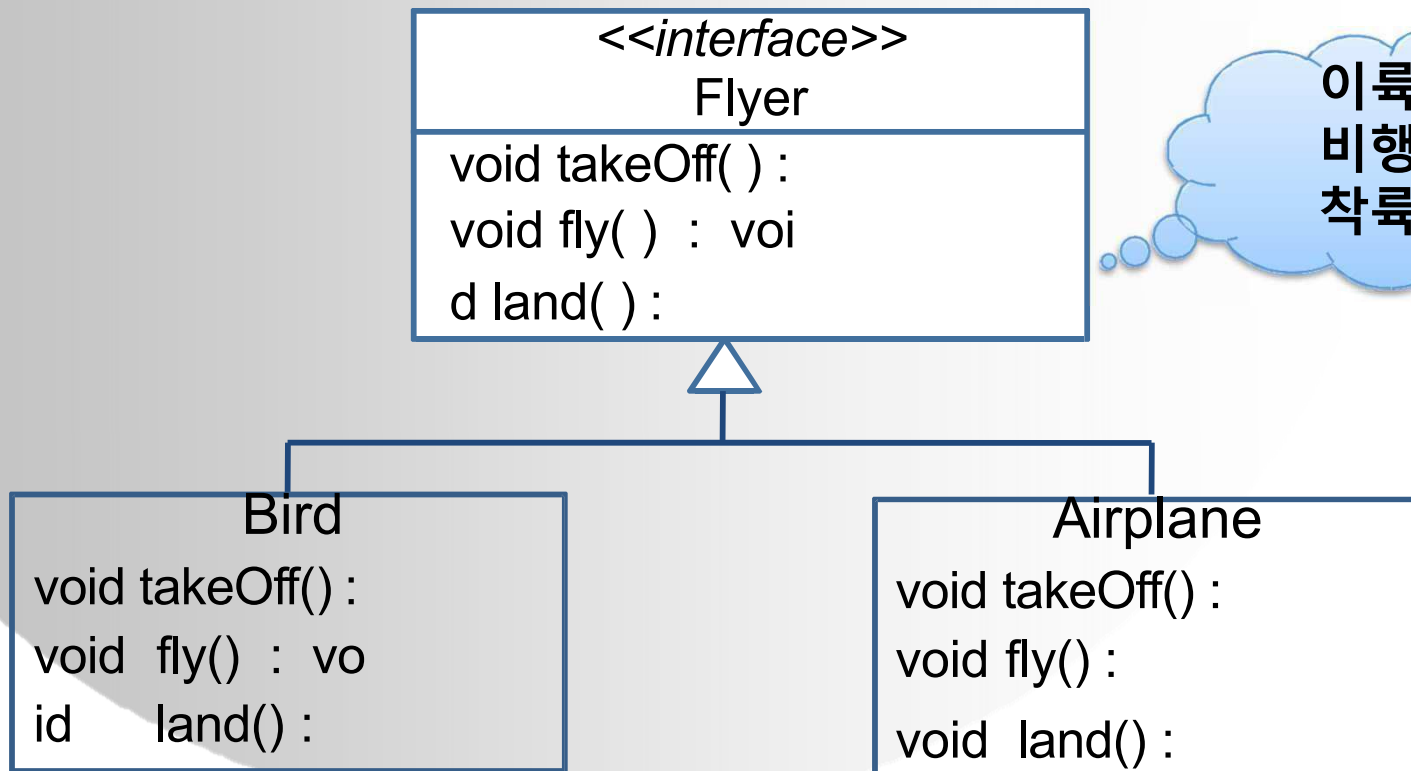




2. 객체지향언어의 4가지 특징

4) 추상화(Abstract)

- ▶ 객체들의 공통적인 특징을 추출하여 구현된 클래스
- ▶ abstract class, interface로 구현됨(활용은 상속을 통해서)





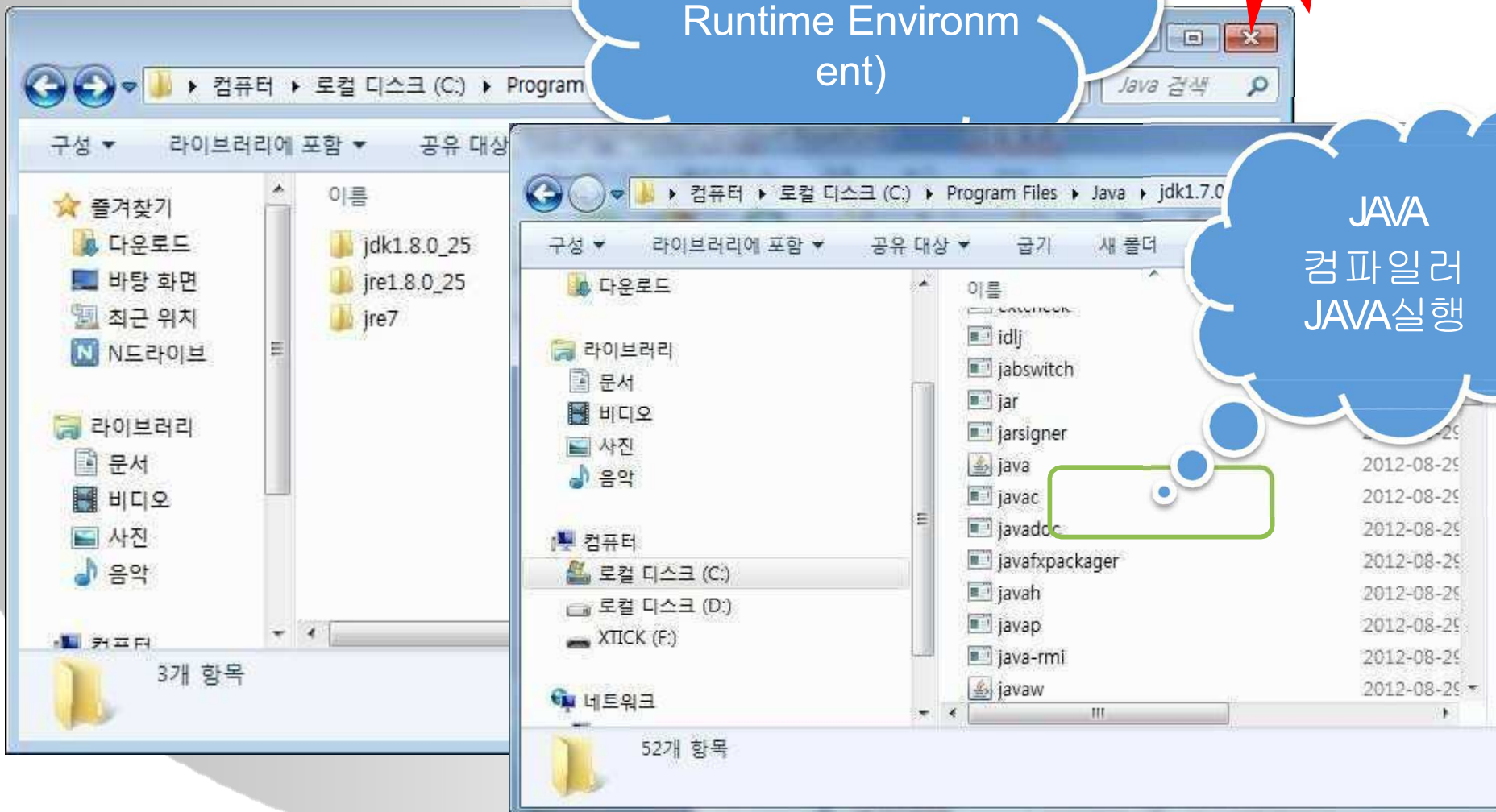
3. 자바 개발 환경 만들기(Java SE)

- Java SE 설치 폴더

JDK(Java Development Kit) JRE(Java Runtime Environment)

JVM

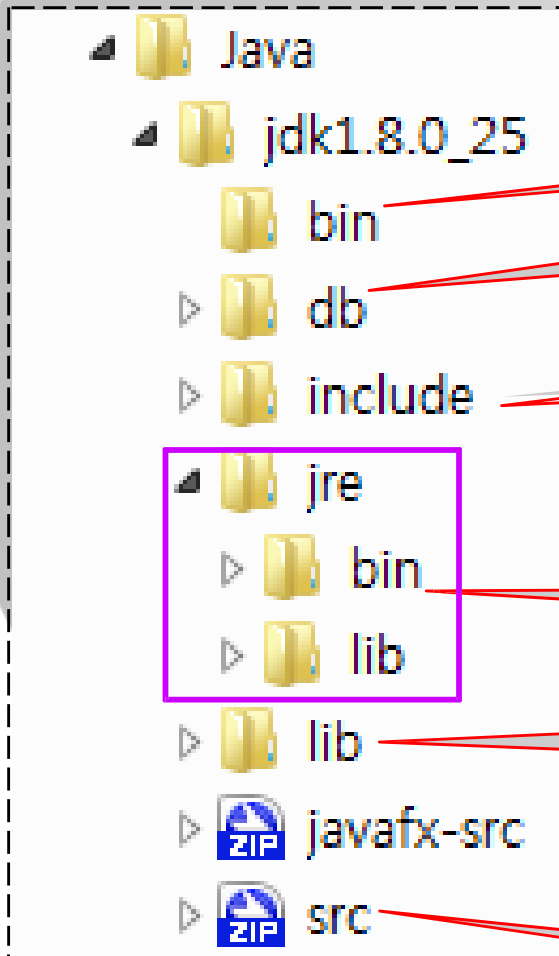
JAVA
컴파일러
JAVA실행





3. 자바 개발 환경 만들기(Java SE)

● JDK 폴더 구성



자바 개발, 실행하는데 필요한 도구와 유틸리티

자바로 DB 응용프로그램을 개발하는데 필요한 도구

네이티브 코드 프로그래밍에 필요한 헤더 파일

자바 런타임 환경. JVM, 클래스 라이브러리 등 실행에 필요한 파일

JDK의 기본 라이브러리 외 추가 클래스 라이브러리와 개발 도구에서 필요로 하는 여러파일

자바API 클래스들에 대한 자바 소스



3. 자바 개발 환경 만들기(Java SE)

- Java 환경변수 설정

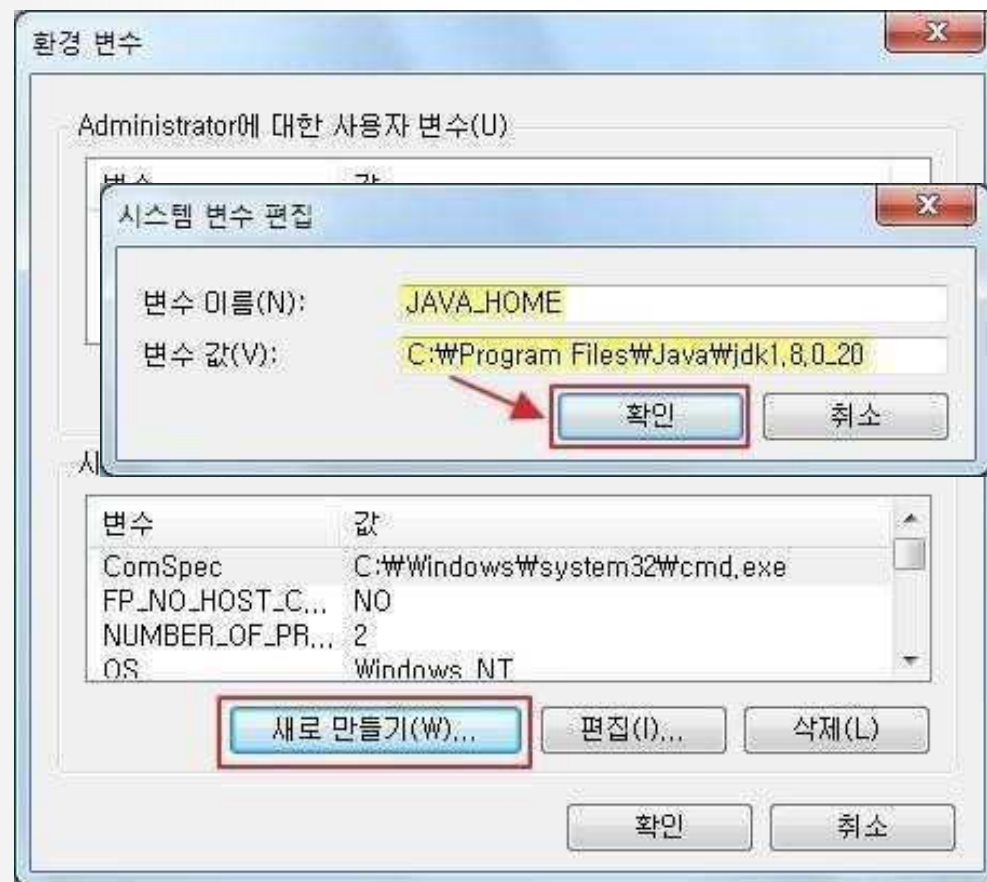
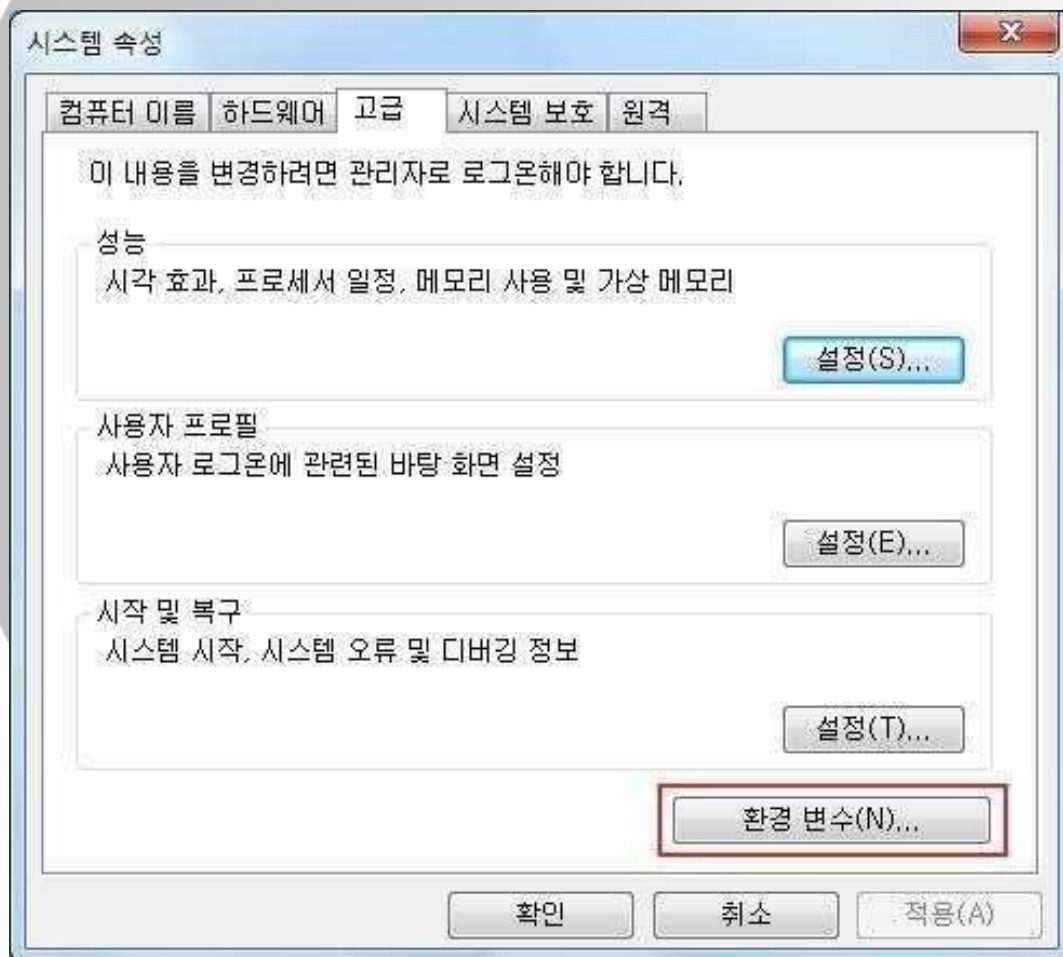
➤ [제어판] → [시스템] → [고급 시스템 설정]





3. 자바 개발 환경 만들기(Java SE)

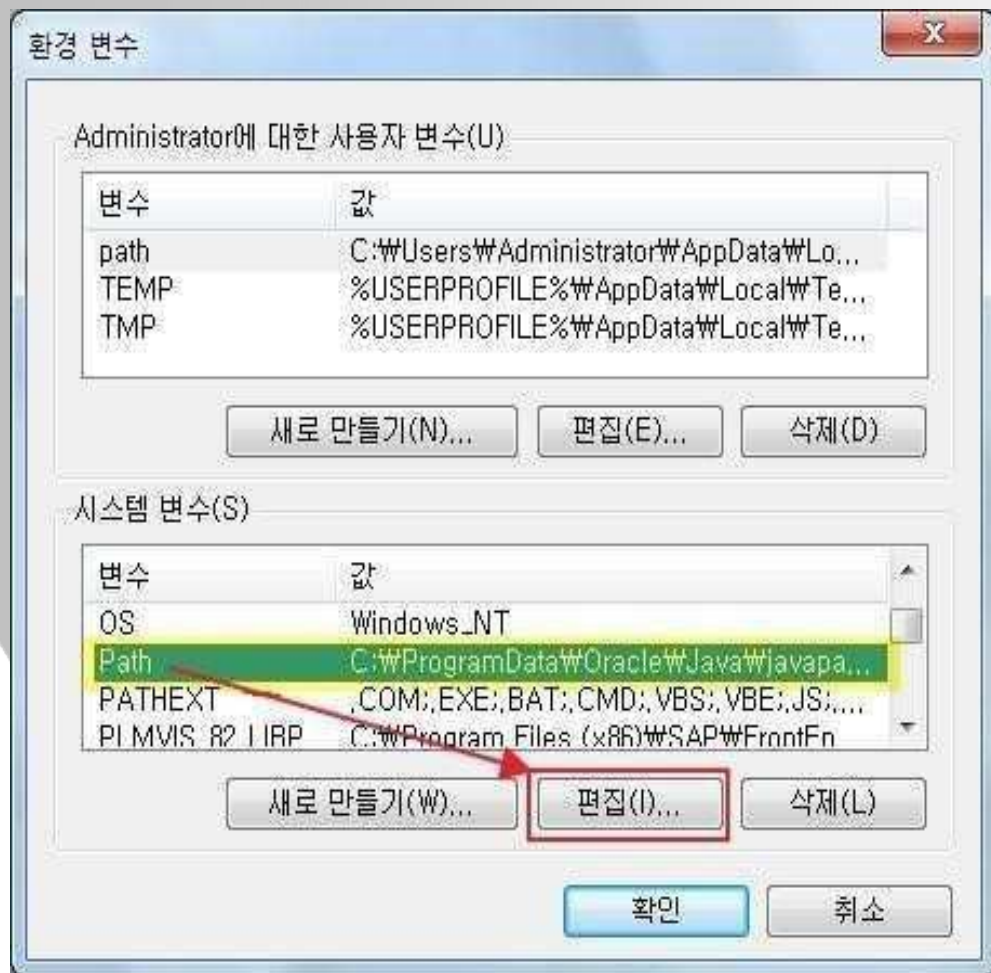
- Java 환경변수 설정





3. 자바 개발 환경 만들기(Java SE)

- Java 환경변수 설정





4. API & IDE

● API Document

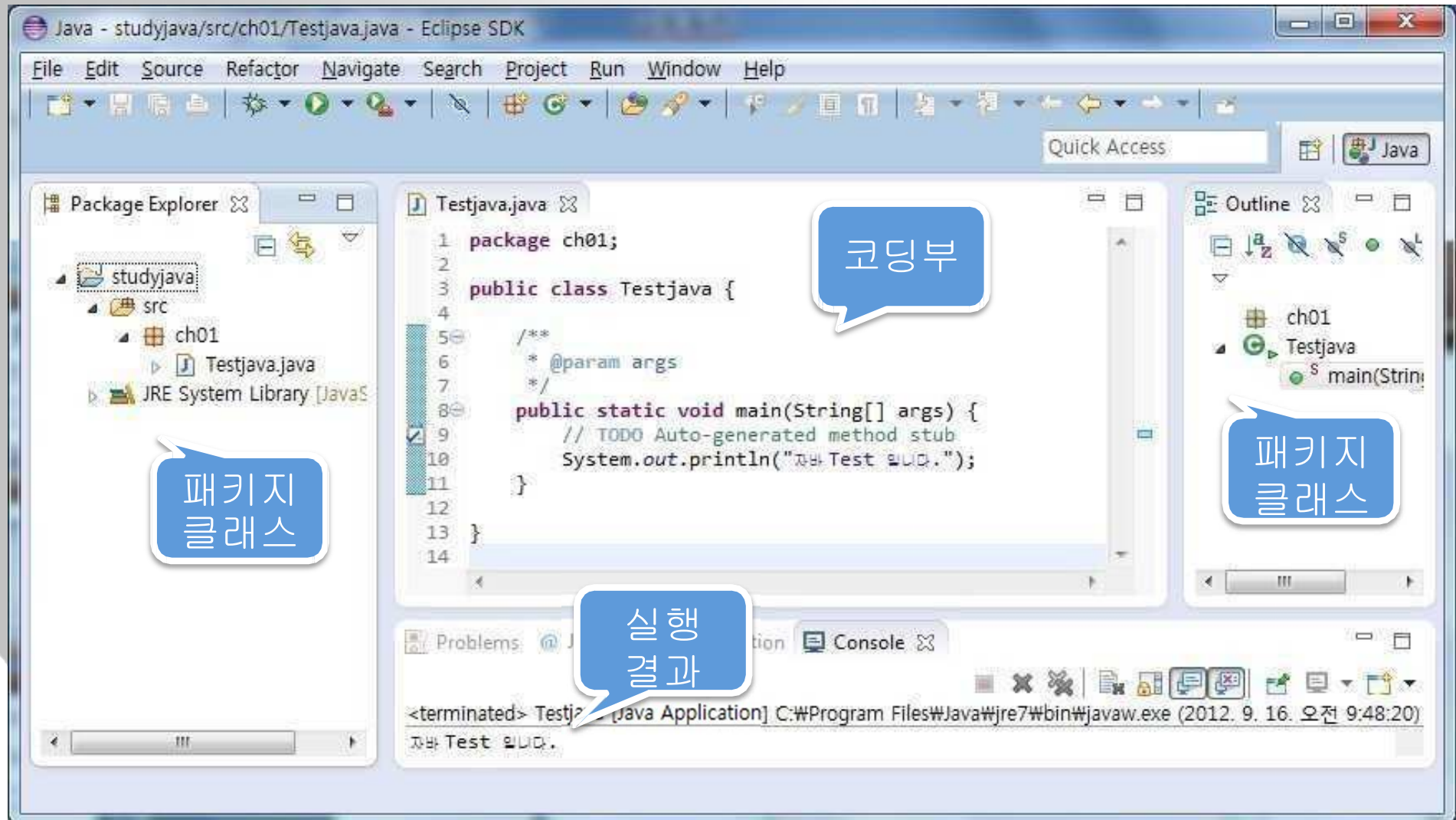
- Class와 Package에 해당하는 정보를 제공하는 참조문서
- 오라클(<http://docs.oracle.com/javase/8/docs/api/>)사 제공





4. API & IDE

- Eclipse에서 클래스와 패키지 작성





4. API & IDE

- Eclipse 활용 연습문제

The screenshot displays the Eclipse IDE interface. On the left, the 'Package Explorer' shows a project named 'StudyJava' with a source folder 'src' containing a package 'ch02' and a class 'SumTest.java'. The 'JRE System Library' is also listed. The main editor window shows the code for 'SumTest.java':

```
package ch02;

public class SumTest {

    /**
     * @param a
     * @param b
     */
    public static void main(String[] args) {
        // TODO
        int a, b;
        a = Integer.parseInt(args[0]);
        b = Integer.parseInt(args[1]);
        sum = a + b;
        System.out.println("Sum: " + sum);
    }
}
```

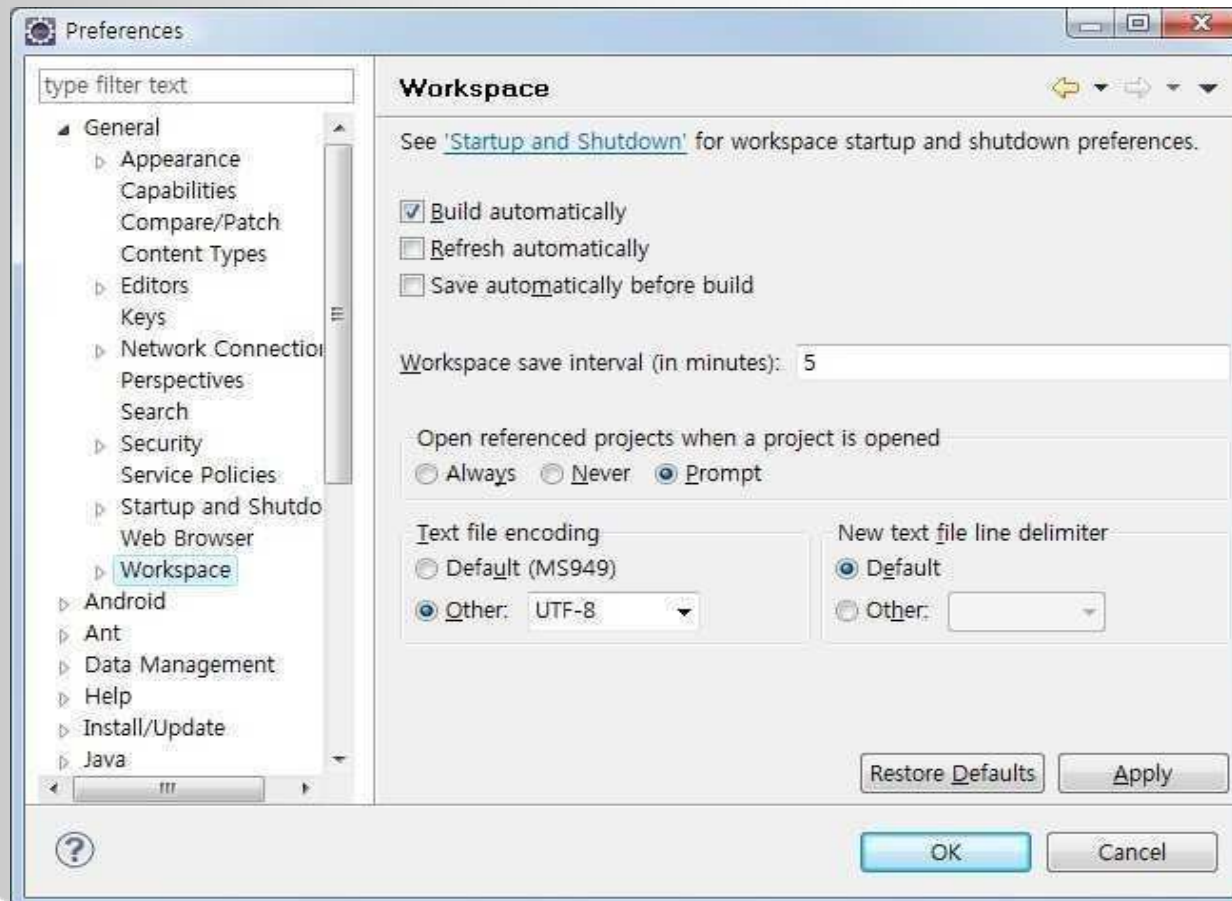
Overlaid on the right is the 'Run Configurations' dialog box. It has a title bar 'Run Configurations' and a subtitle 'Create, manage, and run configurations'. Below the subtitle is the text 'Run a Java application'. The dialog is divided into two main sections. The left section, titled 'type filter text', shows a tree view of the project structure with 'SumTest' selected. The right section contains configuration details: 'Name: SumTest', 'Main' as the selected tab, 'Arguments' as the selected sub-tab, and 'JRE' as the selected JRE. The 'Program arguments:' field contains '100 200'. There is a 'Variables...' button next to it. The 'VM arguments:' field is empty. At the bottom of the dialog are buttons for 'Apply', 'Revert', 'Run', and 'Close'. Below the IDE, the 'Console' window shows the output: '<terminated> SumTest [Java Application] C:\Program Files\Java\jdk1.7.0_07\bin\javaw.exe (2012, 9, 1) 100+200=300'.



4. API & IDE

❖ 옵션 설정

- 실습과 한글 호환성 확보를 위해 몇 가지 옵션을 조정
- 워크스페이스의 문자 셋을 UTF-8로 변경한다.

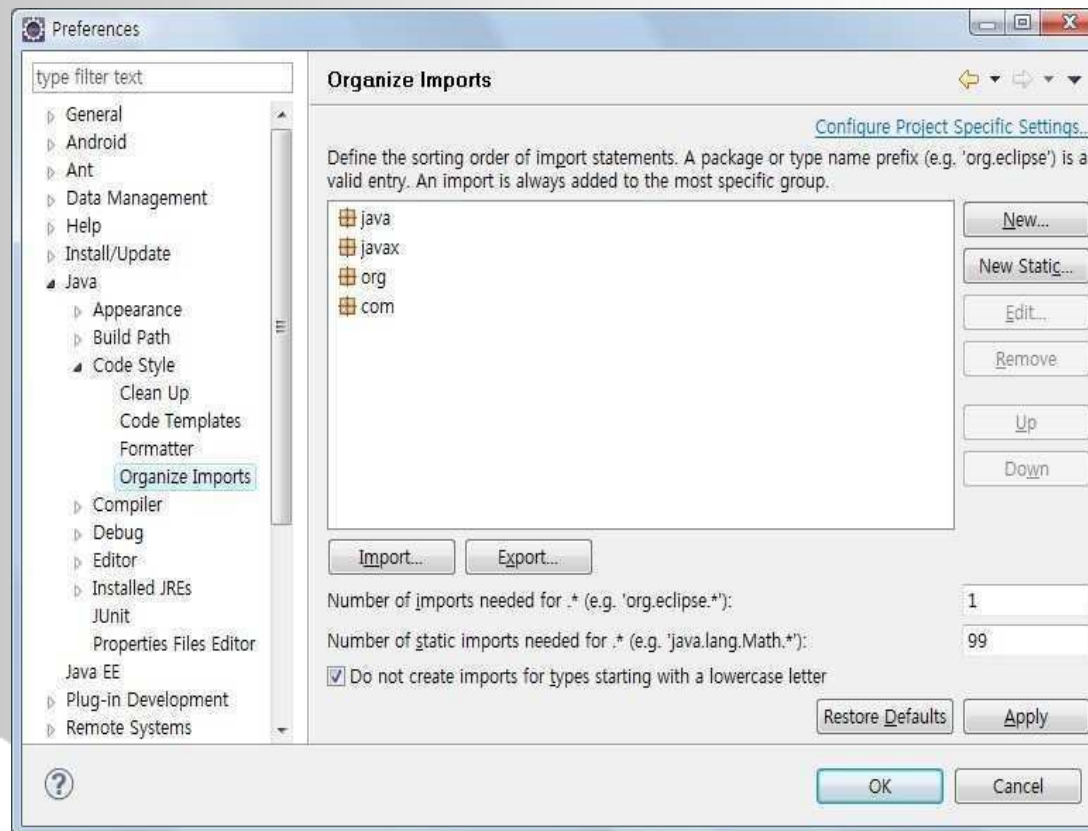




4. API & IDE

❖ 옵션 설정

- [Java-Code Style-Organize Imports] 페이지의 “Number of imports needed for .*” 옵션을 1 변경
 - 이클립스는 import문을 자동으로 정리해 주는 편리한 기능을 제공하는데 매 클래스마다 import문이 작성되어 소스가 너무 길어집니다.
 - 이 값을 1로 바꿔 놓으면 패키지의 모든 클래스를 한꺼번에 임포트 해 주므로 실습할 때 편리하고 결과 소스가 짧아지는 효과가 있습니다.

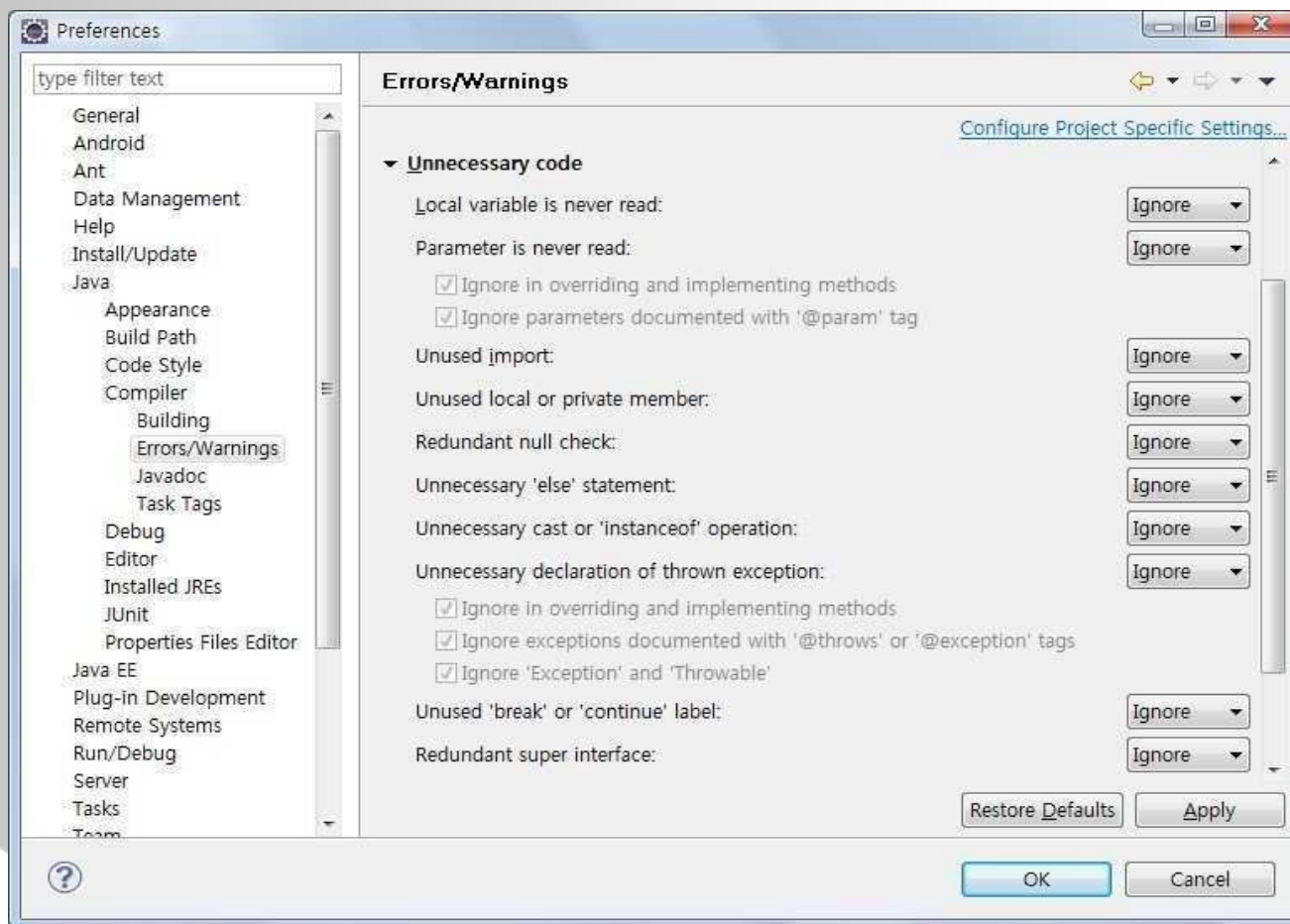




4. API & IDE

❖ 옵션 설정

- [Java-Compiler-Errors-Warnings-Unnecessary code]의 모든 옵션들을 Ignore로 변경한다. 이 옵션을 바꾸지 않으면 실습 중에 수많은 경고가 발생





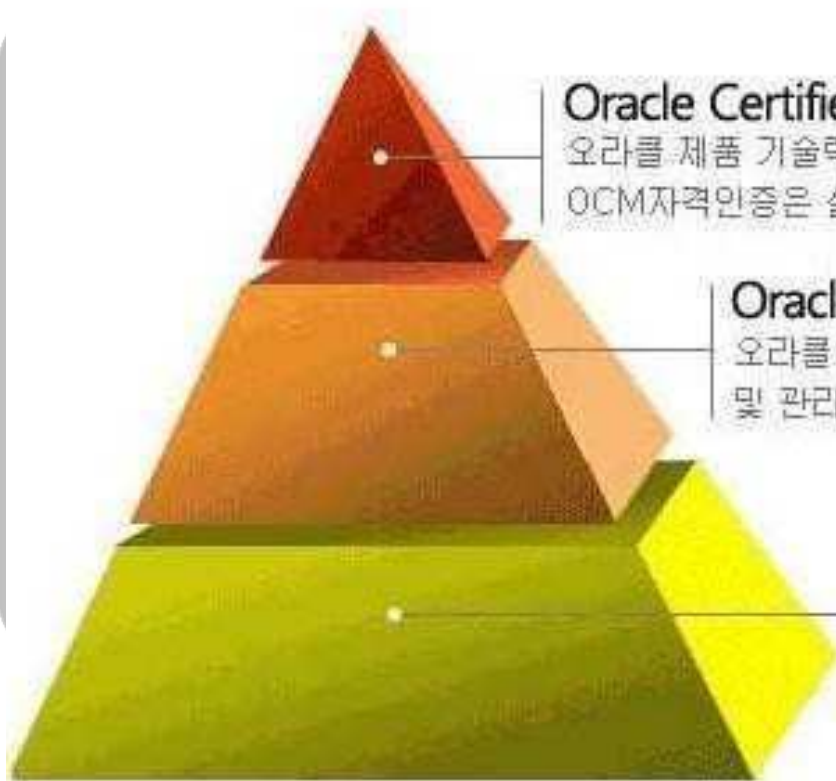
5. 국제공인자격증(JAVA)

1. OCJP(Oracle Certified Java Programmer)=SCJP
 - JAVA기반의 모든 툴 개발에 필요한 기초 지식 인증
2. OCWCD(Oracle Certified Web Component Developer)
 - JAVA, JSP 정도의 모든 툴 개발에 필요한 지식을 인증
3. OCBCD(Oracle Certified Business Component Developer)
 - EJB 기술과스펙, 인터페이스와 웹 서비스를 이용하여 컴포넌트 설계 / 개발능 력 인증

	OCJP	OCWCD	OCBCD
시 험 시 간	180분	180분	145분
문 할 수	60문항	69문항	61문항
문 제 유 형	객관식, 단답형	객관식, D&D	객관식, D&D
합 격 점 수	58%	70%	59%
응 시 자 격	없음	OCJP 취득자	OCJP 취득자



5. 국제공인자격증(DB)



Oracle Certified Master (OCM)

오라클 제품 기술력에 대한 최고의 지식과 능력을 보유하였음을 인증합니다.
OCM자격인증은 실무형 Practice를 기반으로 복잡한 문제해결 능력을 평가합니다.

Oracle Certified Professional (OCP)

오라클 제품기술에 대해 필요한 개발능력(기업형 데이터베이스/미들웨어 개발)
및 관리능력에 대한 전문기술을 인증합니다.

Oracle Certified Associate (OCA)

OCA자격인증은 개인이 오라클 제품에 대한 기본적인 기술을
갖추었음을 증명합니다. 또한 OCA 자격인증은 IT기술 직무 부분에
반드시 필요한 능력을 평가하게 됩니다.