

[단원 01]

자바 개요

동양미래대학교
컴퓨터정보공학과 강 환수 교수



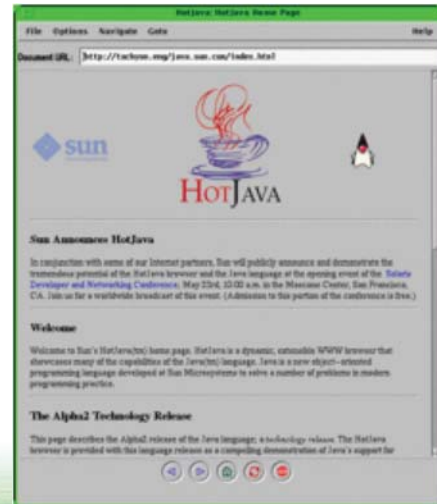
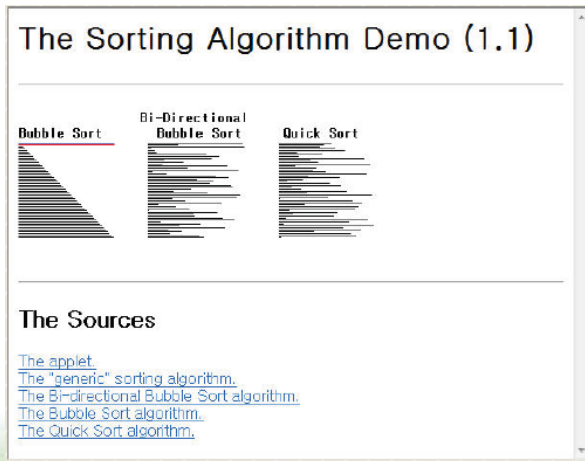
1. 자바 소개



자바 소개

❖ 1995년에 발표된 객체지향 프로그래밍 언어

- JDK (Java Development Kit)
 - 자바 개발 도구
- HotJava
 - 애플릿을 실행할 수 있는 전용 인터넷 브라우저



자바의 역사

❖ Green Project

- 선 마이크로시스템즈 사는 1990년 초 양방향 TV를 만드는 제어박스의 개발 프로젝트
- 초기에는 객체지향 언어로 C++ 언어를 이용
 - C++의 부족으로 오크(Oak, 떡갈나무)라는 언어를 직접 개발

❖ Java

- 제임스 고슬링 (James Gosling) 개발
- 자바는 인터넷 이용에 적합하도록 만들어져 현재까지 여러 분야에 사용되는 프로그래밍 언어로 발전
- 2009년 선 마이크로시스템즈 사는 오라클 (Oracle)에 합병





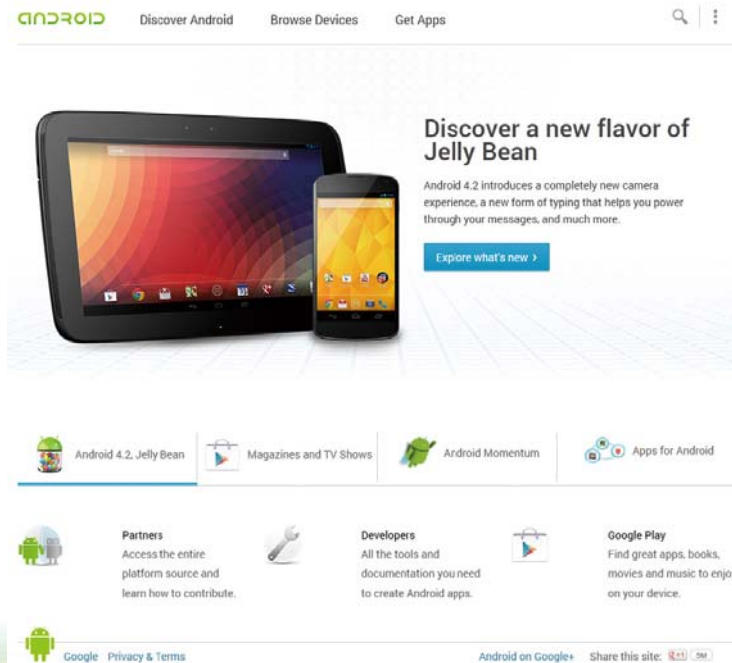
안드로이드 앱 개발 언어 자바

❖ 안드로이드

- 구글(google)에서 2007년 발표한 모바일 운영체제

❖ 안드로이드 앱

- 자바는 안드로이드 응용 프로그램의 대표적인 개발 언어



2. 자바 프로그램과 환경



자바 프로그래밍

❖ 자바 소스

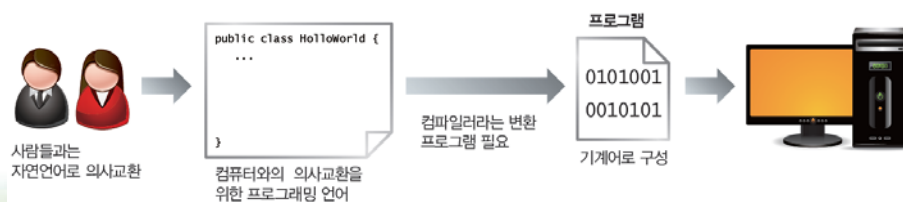
- 확장자는 `java`, `HelloWorld.java`와 같으며 대소문자를 구분
- 소스에서 `public` 인 클래스 이름과 동일
 - 다음 소스 파일은 반드시 `HelloWorld.java`

```

1 public class HelloWorld {
2
3
4     public static void main(String[] args) {
5
6         System.out.println("안녕하세요. 여러분!");
7
8     }
9
10 }
11
  
```

클래스 이름

❖ 컴파일러



바이트코드와 실행

❖ 바이트코드

- 자바 소스를 컴파일하면
 - 바이트코드(bytecode) 생성
- 바이트코드의 확장자는 `class`
 - 플랫폼에 독립적인 명령어로 구성된 이진 파일(binary file)
 - 자바 플랫폼에서 인터프리터(interpreter)에 의해 실행 되는 파일
 - 플랫폼에 독립적으로 자바 플랫폼이 설치된 여러 플랫폼에서 실행

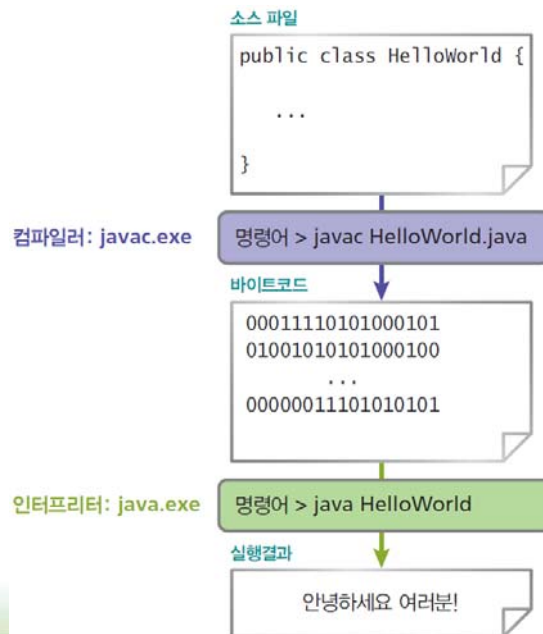




컴파일과 실행

❖ 컴파일과 실행

- 명령어 javac 자바소스파일
- 명령어 java 클래스이름



자바 플랫폼과 자바 가상 기계

❖ 플랫폼

- 각종 프로그램이 실행되는 하드웨어와 소프트웨어로 구성된 실행 환경

❖ 자바 플랫폼

- 자바 가상 기계(JVM: Java Virtual Machine)와 자바 응용 프로그래밍 인터페이스(Java API: Application Programming Interface)로 구성
- 자바 가상 기계
 - CPU와 같이 실행할 명령어 집합을 갖는 소프트웨어로 제작된 가상의 기계





자바 API

❖ 자바 API

- 자바 프로그램을 실행하기 위한 각종 클래스 라이브러리

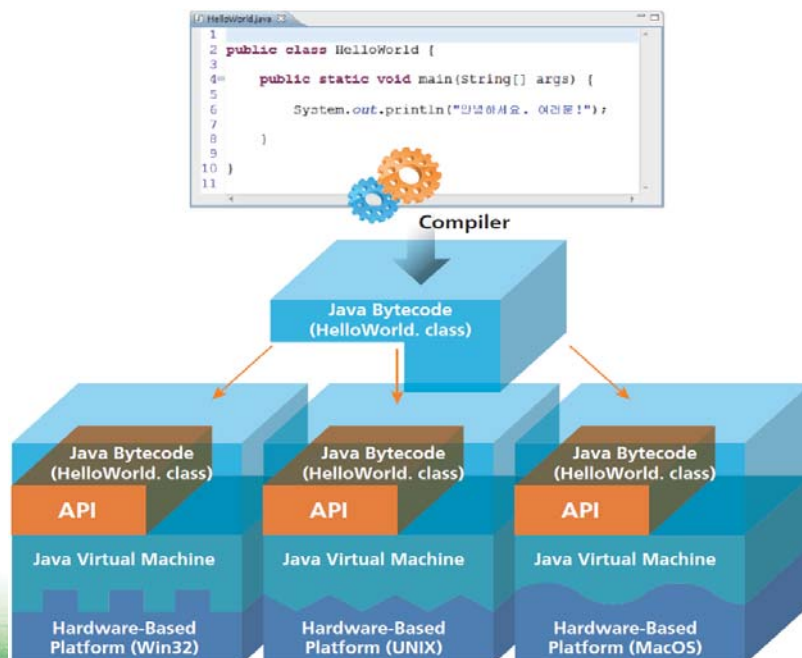
User Interface Toolkits	Swing		Java 2D		AWT	Accessibility		Java SE API
	Drag and Drop		Input Methods		Image I/O	Print Service	Sound	
Integration Libraries	IDL	JDBC	JNDI	RMI	RMI-IIOP		Scripting	
	Beans	Int'l Support		Input/Output		JMX		
Other Base Libraries	JNI	Math		Networking		Override Mechanism		
	Security	Serialization		Extension Mechanism		XML JAXP		
lang and util Base Libraries	lang and util	Collections		Concurrency Utilities		JAR		
	Logging	Management		Preferences API		Ref Objects		
	Reflection	Regular Expressions		Versioning		Zip	Instrumentation	
Java Virtual Machine	Java HotSpot Client and Server VM							



플랫폼에 독립적

❖ 자바 가상 기계에서 실행되는 자바 바이트코드는 플랫폼에 독립적

- “Write Once, Run Anywhere!”





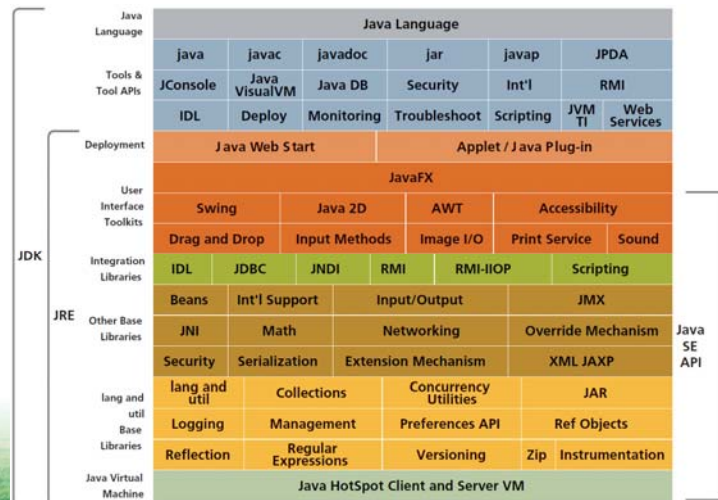
자바 개발 환경

❖ JRE(Java Runtime Environment)

- 자바 실행 환경
 - 자바 API와 자바 가상 기계
 - 그리고 자바 프로그램을 실행하기 위한 여러 컴포넌트(component)로 구성

❖ JDK(Java Development Kit)

- 자바 언어를 이용하여 프로그램을 개발하기 위한 최소한의 환경



단원 01 자바 개요

13

절대 JAVA



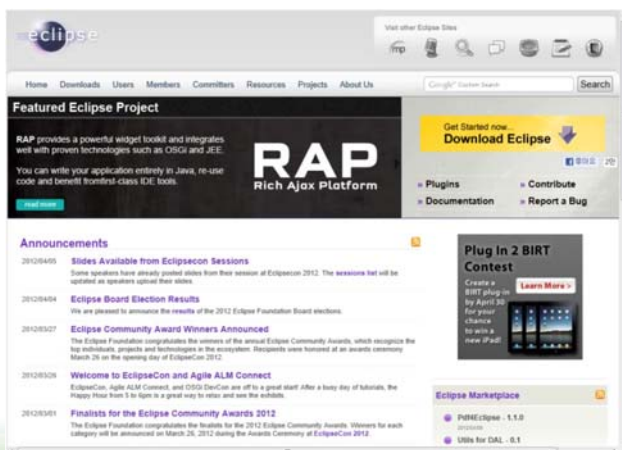
자바 통합 개발 환경

❖ IDE(Integrated Development Environments)

- 프로그램 개발에 필요한 컴파일러(compiler), 디버거(debugger), 링커(linker), 에디터(editor) 등을 통합적으로 제공하는 개발 환경

❖ 대표적인 자바 통합개발환경

- 이클립스와 넷빈



단원 01 자바 개요

14

절대 JAVA



3. 자바 개발 환경 설치



Java SE 설치

Section 3 자바 개발 환경 설치 p12

❖ JDK 설치

- 리눅스(Linux), 솔라리스(Solaris), 윈도우(Windows) 기반의 다양한 제품 중에서 설치하려는 플랫폼에 적합한 JDK 버전 설치
- 구체적인 설치는 책 참조

Java SE Development Kit 7u3

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

☐ Accept License Agreement ☒ Decline License Agreement

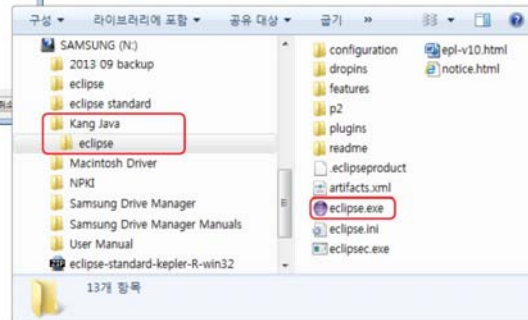
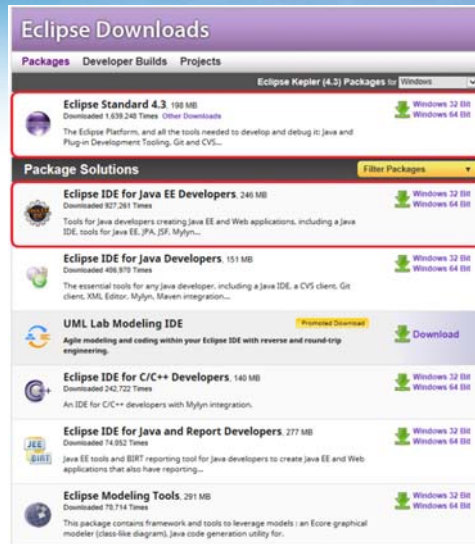
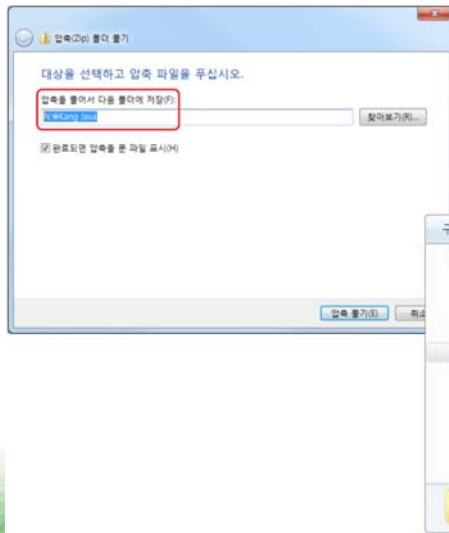
Product / File Description	File Size	Download
Linux x86 (32-bit)	63.65 MB	jdk-7u3-linux-i586.rpm
Linux x86 (32-bit)	78.66 MB	jdk-7u3-linux-i586.tar.gz
Linux x64 (64-bit)	64.53 MB	jdk-7u3-linux-x64.rpm
Linux x64 (64-bit)	77.3 MB	jdk-7u3-linux-x64.tar.gz
Solaris x86 (32-bit)	135.96 MB	jdk-7u3-solaris-i586.tar.Z
Solaris x86 (32-bit)	81.4 MB	jdk-7u3-solaris-i586.tar.gz
Solaris SPARC (32-bit)	138.92 MB	jdk-7u3-solaris-sparc.tar.Z
Solaris SPARC (32-bit)	86.07 MB	jdk-7u3-solaris-sparc.tar.gz
Solaris SPARC (64-bit)	16.14 MB	jdk-7u3-solaris-sparc64.tar.Z
Solaris SPARC (64-bit)	12.31 MB	jdk-7u3-solaris-sparc64.tar.gz
Solaris x64 (64-bit)	14.46 MB	jdk-7u3-solaris-x64.tar.Z
Solaris x64 (64-bit)	9.25 MB	jdk-7u3-solaris-x64.tar.gz
Windows x86 (32-bit)	84.12 MB	jdk-7u3-windows-i586.exe
Windows x64 (64-bit)	87.41 MB	jdk-7u3-windows-x64.exe





이클립스 설치

- ❖ 내려받기
 - Eclipse IDE for Java Developers
- ❖ 압축을 풀어 설치



절대
JAVA



4. 자바 개발 과정



절대
JAVA



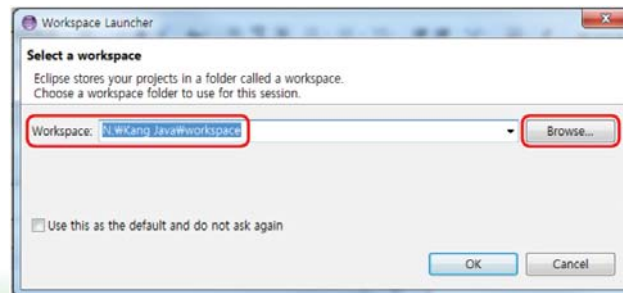
자바 개발과 작업공간

❖ 개발과정

- ① 작업공간(workspace) 지정
- ② 자바 프로젝트(Java Project) 생성
- ③ 자바 클래스(Java Class) 생성
- ④ 소스 작성
- ⑤ 자바 응용 프로그램 실행

❖ 작업공간(workspace)

- 프로그램 개발 과정에서 생성되는 여러 파일이 저장되는 장소



첫 자바 소스

❖ HelloJava.java

- 다음 주의
 - 대소문자
 - 큰따옴표 “ ”
 - .
 - ;

```
public class HelloJava {

    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub

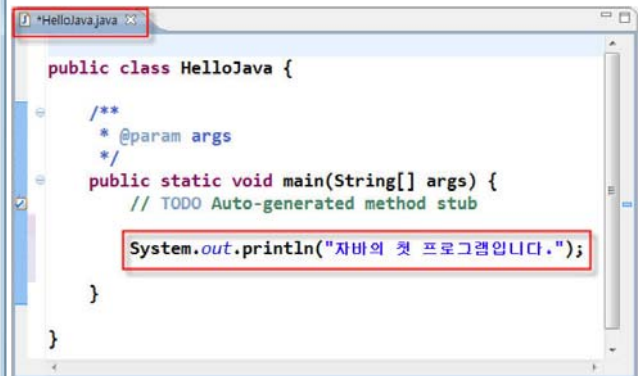
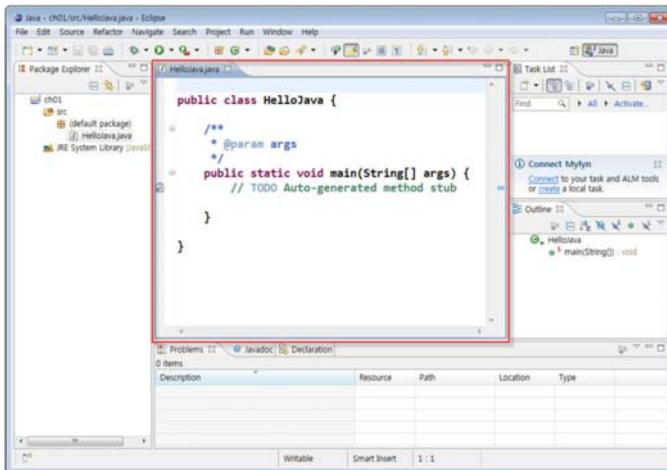
        System.out.println("자바의 첫 프로그램입니다.");
    }

}
```



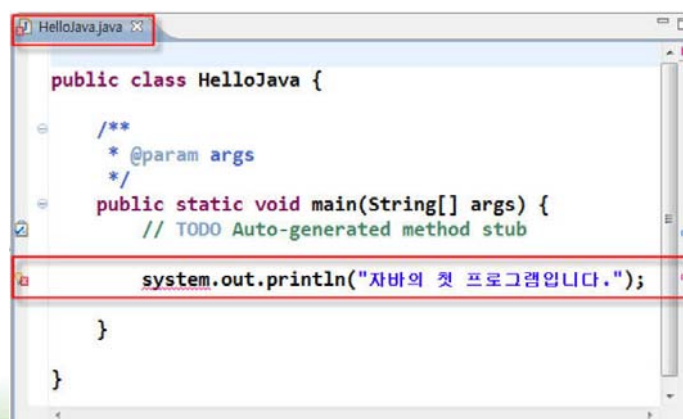
첫 자바 소스 화면

- ❖ 클래스 HelloJava
 - 자동 생성에서 코딩



프로그램 오류

- ❖ 다음은 흔히 발생하는 오류이니 주의
 - ① 괄호나 중괄호 또는 큰따옴표 등이 빠지면 오류가 발생
 - ② 키워드 static void 등이 잘못 입력되거나 빠지면 오류가 발생
 - ③ String 또는 System 등에서 s를 소문자로 잘못 입력한 경우 오류가 발생
 - ④ 문장 뒤에 세미콜론 ;이 빠진 경우 오류가 발생
 - ⑤ public class에서 순서가 바뀌어도 오류가 발생





자바 프로그램 실행

❖ 메뉴 [Run/Run As/Java Application]을 선택

- 소스에서 마우스의 오른쪽 버튼으로도 선택

- 화살표 모양을 아 이콘을 펼쳐 [Run As/Java Application]을 선택

- 단축 키 [ctrl]+[F11]으로도 가능

- [Console] 뷰에서 실행 결과를 확인

