



2

COMPUTER PROGRAMMING



**Execution Environment
for JAVA Development**



CONTENTS

- **General Architecture of JAVA**
- **Configuration**
- **Execution Environment of JAVA using EditPlus tool**
- **JAVA Development Environment**
- **JAVA Integration Environment**



Main Structure

01: `package javaapplication1; /* 패키지 정의: */`

02: `public class Main {`

03: `public Main() {`

`//`

04: `}`

05: `public static void main(String[] args) {`

06: `System.out.println("Hello World");`

07: `}`

08: `}`

Process Flow of JAVA

□ JAVA Execution Structure

[1] 소스 코드
(Main.java)

```
package javaapplication1;  
public class Main{  
    public Main( ){  
    }  
    public static void main(String[] args){  
        System.out.println("Hello World");  
    }  
}
```

소스 코드를 입력한다.

컴파일 과정

javac Main.java

javac를 실행시켜서
Main.java를 컴파일
하여 오류가 없으면
Main.class라는 바
이너리 파일이 생성

[2] 바이너리 코드
(Main.class)

컴파일된 결과물
Main.class

실행과정

java Main

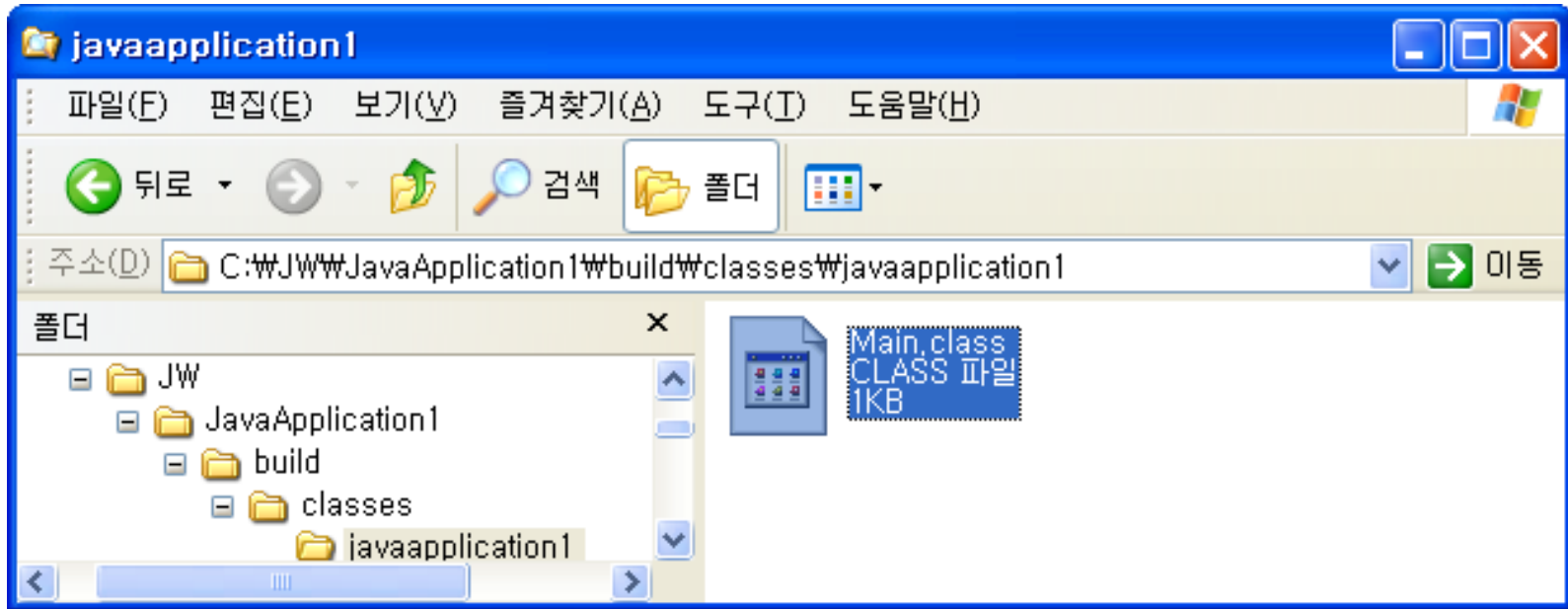
자바 가상 머신을 통해
Main.class 파일을 실행
JVM은 바이트코드를 해당
플랫폼에서 이해할 수 있는
형태로 해석하여 그 프로그
램을 실행시켜줌

[3] 가상 머신

Hello World

Process Flow of JAVA

□ Binary Code



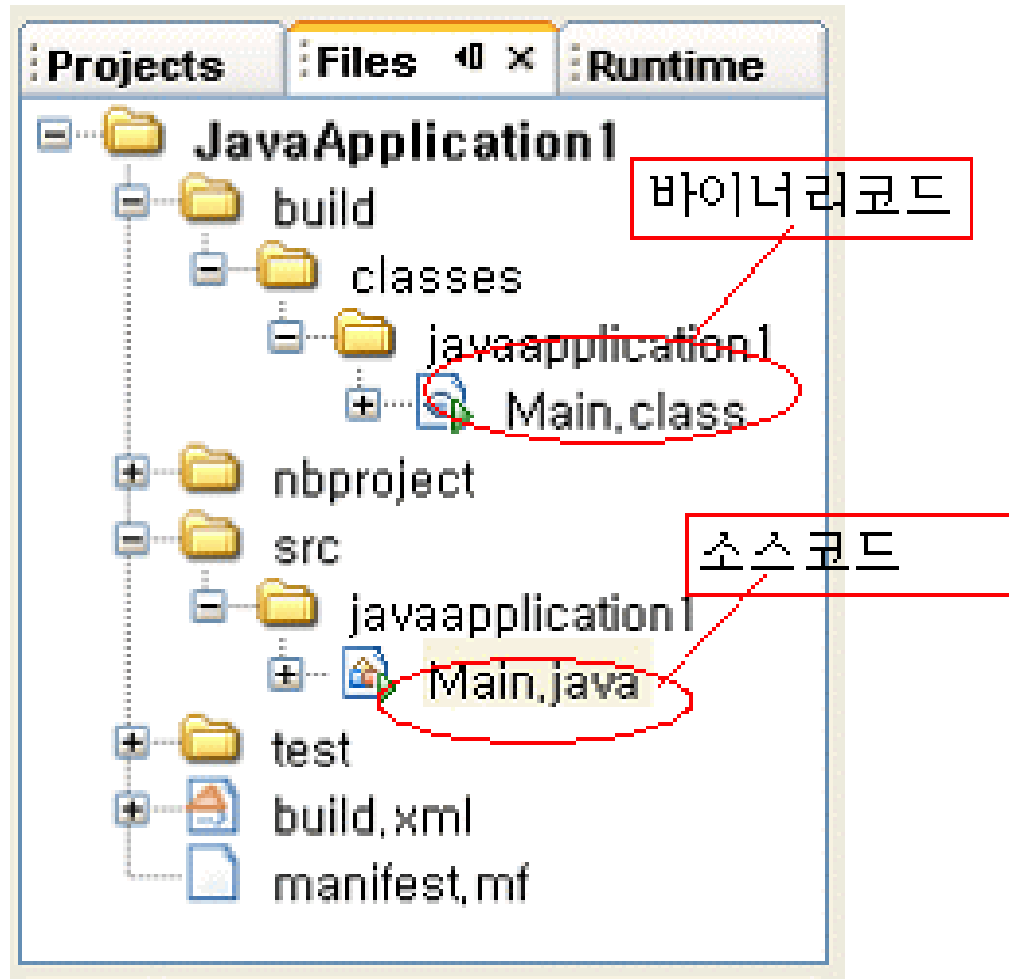
Process Flow of JAVA

- **JAVA Compiler Command : javac**

- **JAVA Interpreter Command : java**
 - That is a tool that can run Java byte code (filename.class) on a JVM(java virtual machine).
 - Java Interpreter command is “java” in command line to execute.

- **SDK (software development kit) tools**
 - C header files and Stub file generator : javah
 - Java Document Creator : javadoc
 - Java Compression tool : jar

Process Flow of JAVA



Java Configuration to Execute

□ JDK installation

- <http://java.sun.com/javase/downloads/index.jsp>

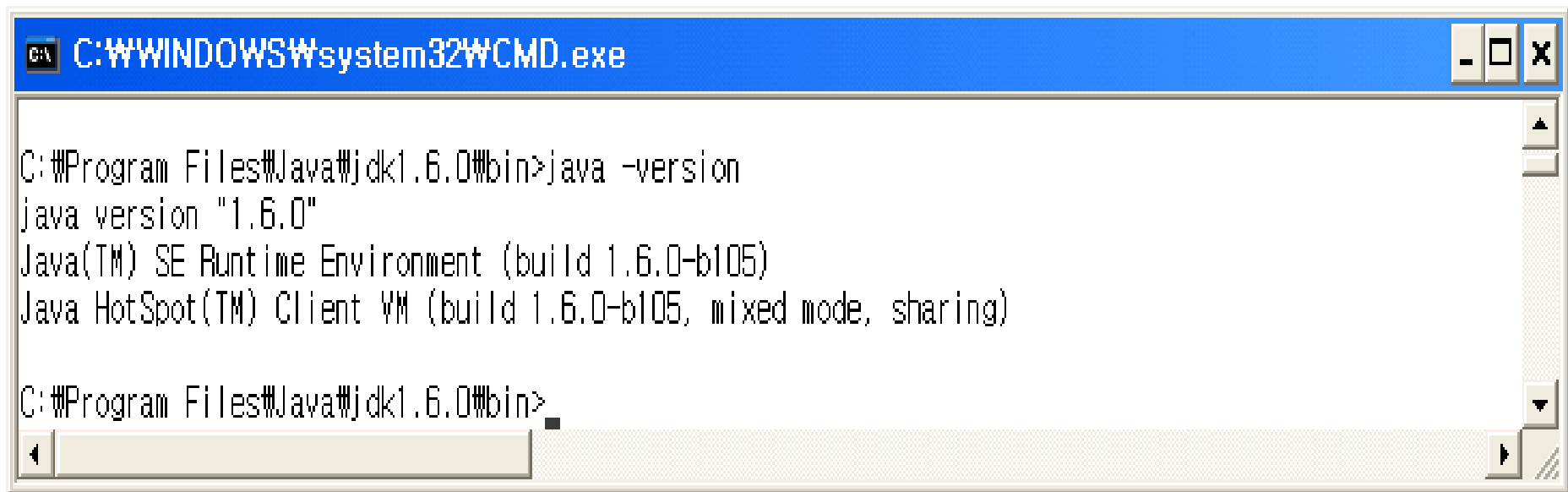
□ Check

- Java compiler file and execution file in folder C:\Program Files\Java\jdk1.6.0\bin whether installed.

Java Configuration to Execute(not Netbean)

□ Check Configuration

- Open a command window to check whether Java execution environment operates properly after installing JDK
 - Start – command line – write "cmd"
- Write "java -version" in **JDK installed directory**(~/bin)
 - move: c~>cd \ or c~>cd directory name



The screenshot shows a Windows command prompt window titled "C:\WINDOWS\system32\CMD.exe". The command prompt is open at the directory "C:\Program Files\Java\jdk1.6.0\bin". The user has entered the command "java -version", and the output is displayed as follows:

```
C:\Program Files\Java\jdk1.6.0\bin>java -version
java version "1.6.0"
Java(TM) SE Runtime Environment (build 1.6.0-b105)
Java HotSpot(TM) Client VM (build 1.6.0-b105, mixed mode, sharing)

C:\Program Files\Java\jdk1.6.0\bin>
```

Java Configuration to Execute(not Netbean)

□ JAVA in Notepad (Memo)

```
class Hello{  
    public static void main(String [] args){  
        System.out.println("Hello World");  
    }  
}
```

Java Configuration to Execute(not Netbean)

□ Java Compile before configuration

- C:\javawork> **javac** Hello.java
>> error occurred, therefore need configuration (path)



Java Configuration to Execute(not Netbean)

- System Variable and Environment Variable (**PATH, CLASSPATH**)
 - OS has the both values.



Java Configuration to Execute(not Netbean)

□ Environment variable : PATH

- If Javac.exe file runs in everywhere, configure PATH

□ Environment variable : CLASSPATH

■ JRE ?

- Is Java execution environment and provides various classes for java program as a compress format.

Java Configuration to Execute(not Netbean)

- JRE(java runtime environment)?



Java Configuration to Execute(not Netbean)

□ To use classes provided by JAVA, configure CLASSPATH.

■ CLASSPATH = .; C:\Program Files\Java\jre 1.6.0\lib

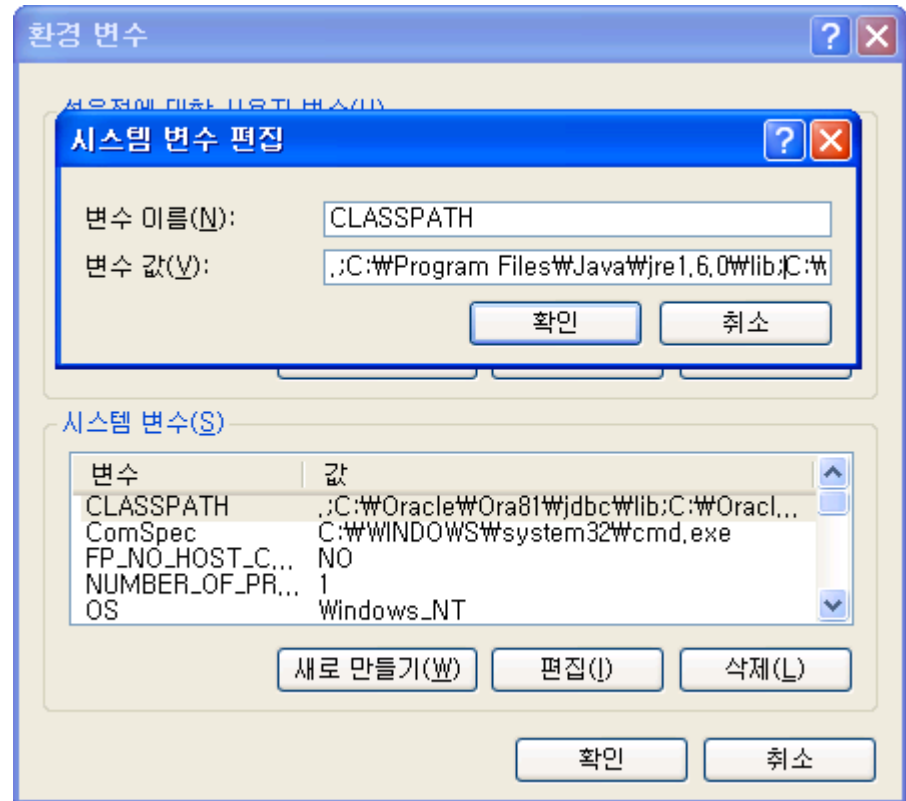
➤ “.” means current directory.

➤ “;” means symbol to distinguish more than two paths as configuring path.

Java Configuration to Execute(path,classpath)

□ 내컴퓨터속성-[시스템 등록 정보] 창의 [고급] 탭에서 [환경 변수] 설정

- Copy string C:\Program Files\Java\jdk1.6.0\bin
javac.exe의 마우스 우측버튼-속성에서 위치를 복사
내컴퓨터의 속성-시스템등록정보-고급탭-환경변수
 - Path: "C:\Program Files\Java\jdk1.6.0_11\bin;"
 - Classpath: "C:\Program Files\Java\jre6\lib"



Java Configuration to Execute

□ Steps of Java compiling after configuration

자바 소스
파일이 있
는지 확인

자바 클래스
파일이 있는
지 확인

```
C:\WINDOWS\system32\CMD.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\성운정>cd javawork
C:\javawork>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: E090-C5DD

C:\javawork 디렉터리

2007-01-07 오후 08:39 <DIR> .
2007-01-07 오후 08:39 <DIR> ..
2006-04-10 오후 08:04          109 Hello.java
                   1개 파일             109 바이트
                   2개 디렉터리 26,462,363,648 바이트 남음

C:\javawork>javac Hello.java
C:\javawork>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: E090-C5DD

C:\javawork 디렉터리

2007-01-07 오후 10:05 <DIR> .
2007-01-07 오후 10:05 <DIR> ..
2007-01-07 오후 10:05          416 Hello.class
2006-04-10 오후 08:04          109 Hello.java
                   2개 파일             525 바이트
                   2개 디렉터리 26,462,318,592 바이트 남음

C:\javawork>
```

자바소스파일이 저장된
javawork 폴더로 이동

자바소스파일을 컴파일하기

Java Configuration to Execute

□ Running java source program

■ Caution : Uppercase, lowercase (class name)

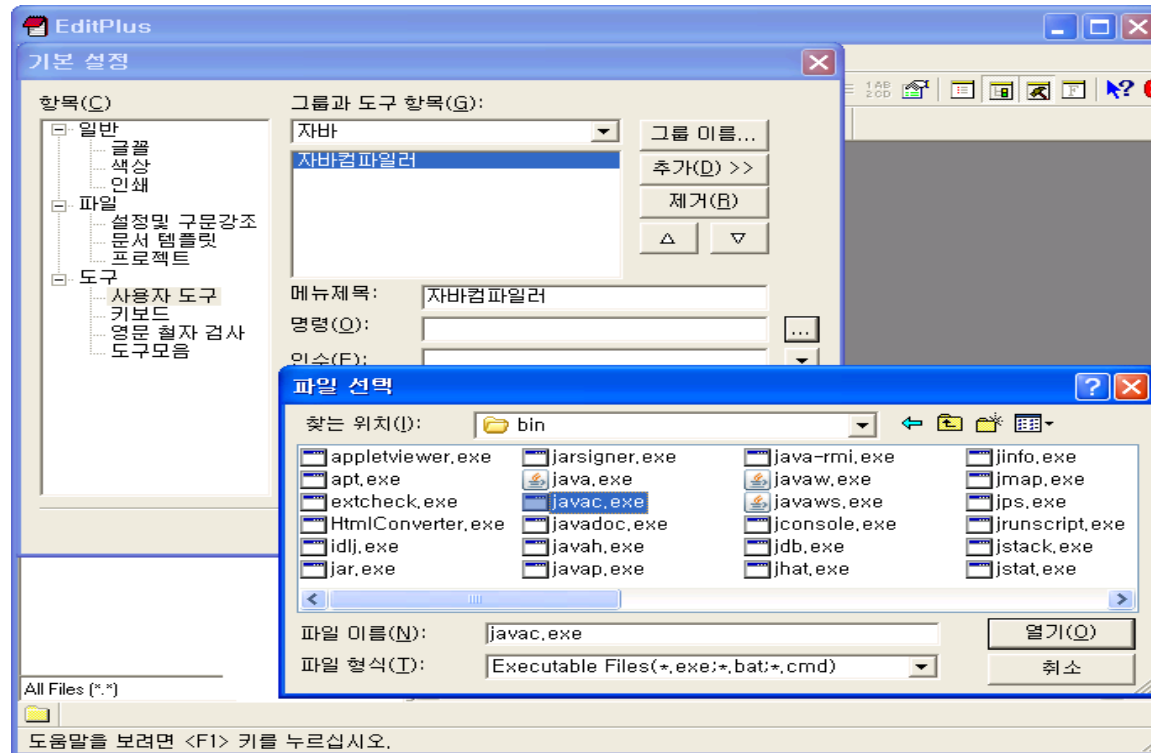
➤ Ex) > java Hello vs >java hello

□ <follow>- Java compiler configuration in EditPlus

- 1. Select menu [도구]-[기본설정]
- 2. [도구]-[사용자 도구]에서 그룹에 가서 [그룹 이름..] 버튼을 눌러서 그룹의 이름을 “자바”로 바꾼다.
- 3. [추가] 버튼을 누른 후 [프로그램]을 선택한다.

JAVA in EditPlus Tool

- 4. [메뉴제목:] 항목의 내용을 “자바컴파일러”로 바꾼다.
 - [명령:]의 [...] 버튼을 클릭 javac.exe 파일을 선택.
 - [인수:]에서 “파일 이름”을 선택.
 - [디렉토리:]에서 “파일 디렉토리” 선택
 - 출력내용 캡처 선택 및 적용 클릭
- 5. 상기에서처럼 “자바실행”을 추가
 - 단, [인수:]에서 “확장자를 뺀 파일이름”을 선택



□ EditPlus에서 자바 실행

- [파일] – [새 파일] – [Java] 선택
 - 새로운 자바예제(“Hello World” 출력) 파일(Class이름은 제외된상태)
- Class 이름에 “firstApp”입력 – [파일] – [저장]
 - (클래스이름과 동일한이름. 단, 확장자는 제외)
- [보기] – [도구모음/창] - [사용자도구모음]
 - 작성한 “자바” 그룹 표시
 - ✓ 좌측: 자바
 - ✓ 1:자바컴파일
 - ✓ 2:자바실행
 - 1번, 2번을 순차적으로 수행

Download NetBean and Installation

□ 선(sun) 사이트 접속

- <http://java.sun.com/javase/downloads/index.jsp>
 - JDK & with NetBean 5.5 선택
 - JDK 6 Update 11 with NetBeans 6.5 선택(web 개발도구 제외)
- 추천 : <http://www.netbeans.org/downloads/index.html>
(All 선택 download)
 - Download
 - Installation

□ Net Bean 6.5 실행

Download NetBean and Installation

- A execution snapshot of Net Bean 6.5



Print out “Hello World”

□ Menu – File – New Project

- Select Categories(JAVA) – Projects (Java Application)
- Choice default
- Click Finish

□ Main 메소드에 다음 문장(“Hello World”)출력해보기

- Public static void main(string[] args){

```
System.out.println(“Hello World”); // code insight, intellisense function
```

```
}
```

- “run”의 “compile _ main.java” 클릭하여 실행
- 결과 확인(“Hello World”) 출력

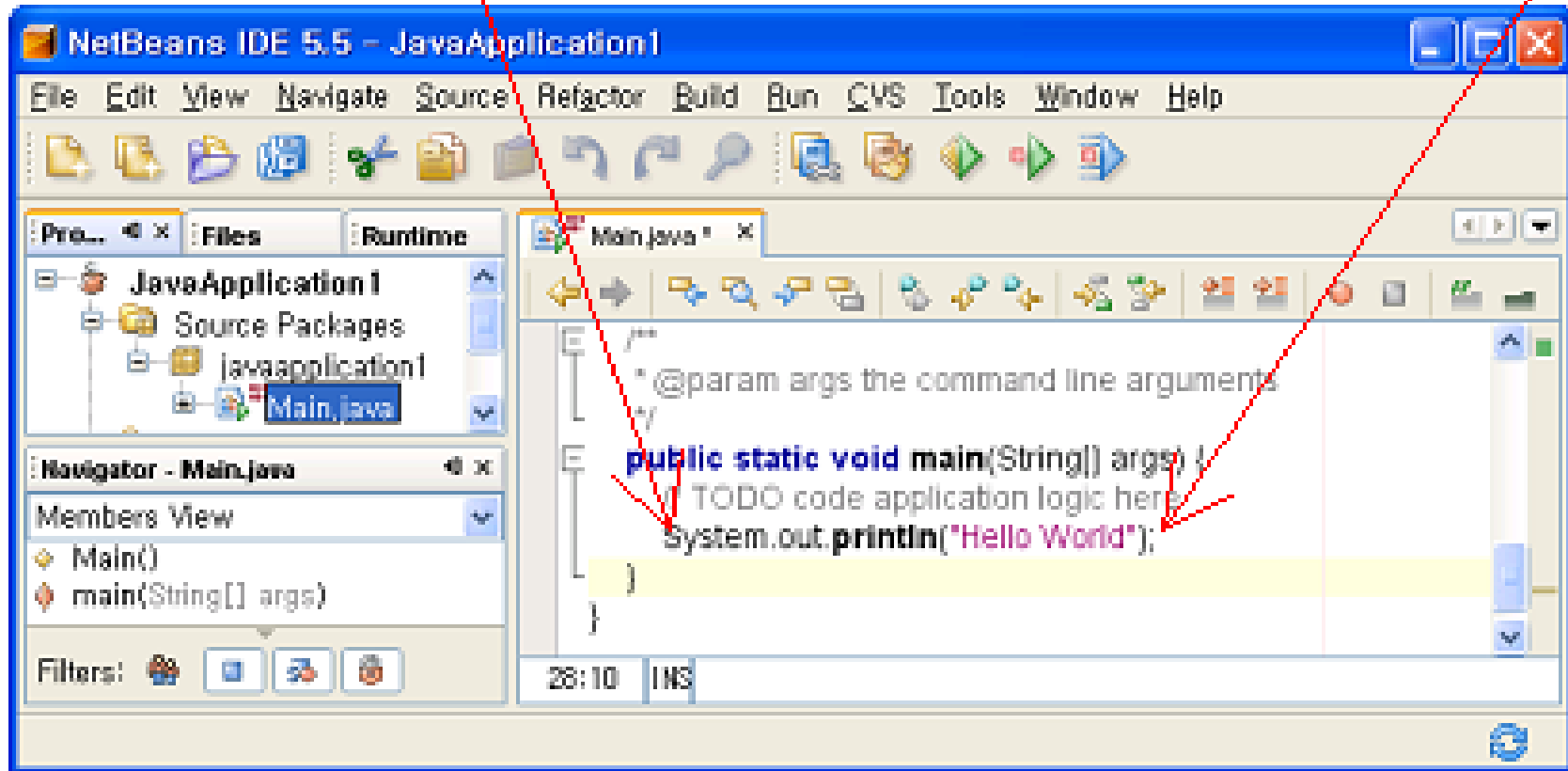
JAVA in Net Bean

외부에서 Main사용시:
`import javaapplication1.Main;`



JAVA in Net Bean

```
System.out.println("Hello World");
```



Code Insight function

The screenshot displays the NetBeans IDE 5.5 interface. The main editor window shows the source code of `Main.java` with a cursor positioned over the `System.out.p` call. A Code Insight popup window is open, listing various `print` methods from the `java.io.PrintStream` class. The selected method is `print(Object obj)`, which has a return type of `void`. Below the list, the IDE provides detailed documentation for the selected method, including its signature, a description of its behavior, parameters, and a 'See Also' section.

java.io.PrintStream.p*

- `print(Object obj)` void
- `print(String s)` void
- `print(boolean b)` void
- `print(char c)` void
- `print(char[] s)` void
- `print(double d)` void
- `print(float f)` void
- `print(int i)` void
- `print(long l)` void
- `printf(String format, Object... args)` PrintStream
- `printf(Locale l, String format, Object... args)` PrintStream
- `println()` void
- `println(Object x)` void
- `println(String x)` void
- `println(boolean x)` void
- `println(char x)` void
- `println(char[] x)` void

java.io.PrintStream

```
public void print(Object obj)
```

Prints an object. The string produced by the [java.lang.String.valueOf\(Object\)](#) method is translated into bytes according to the platform's default character encoding, and these bytes are written in exactly the manner of the [write\(int\)](#) method.

Parameters:

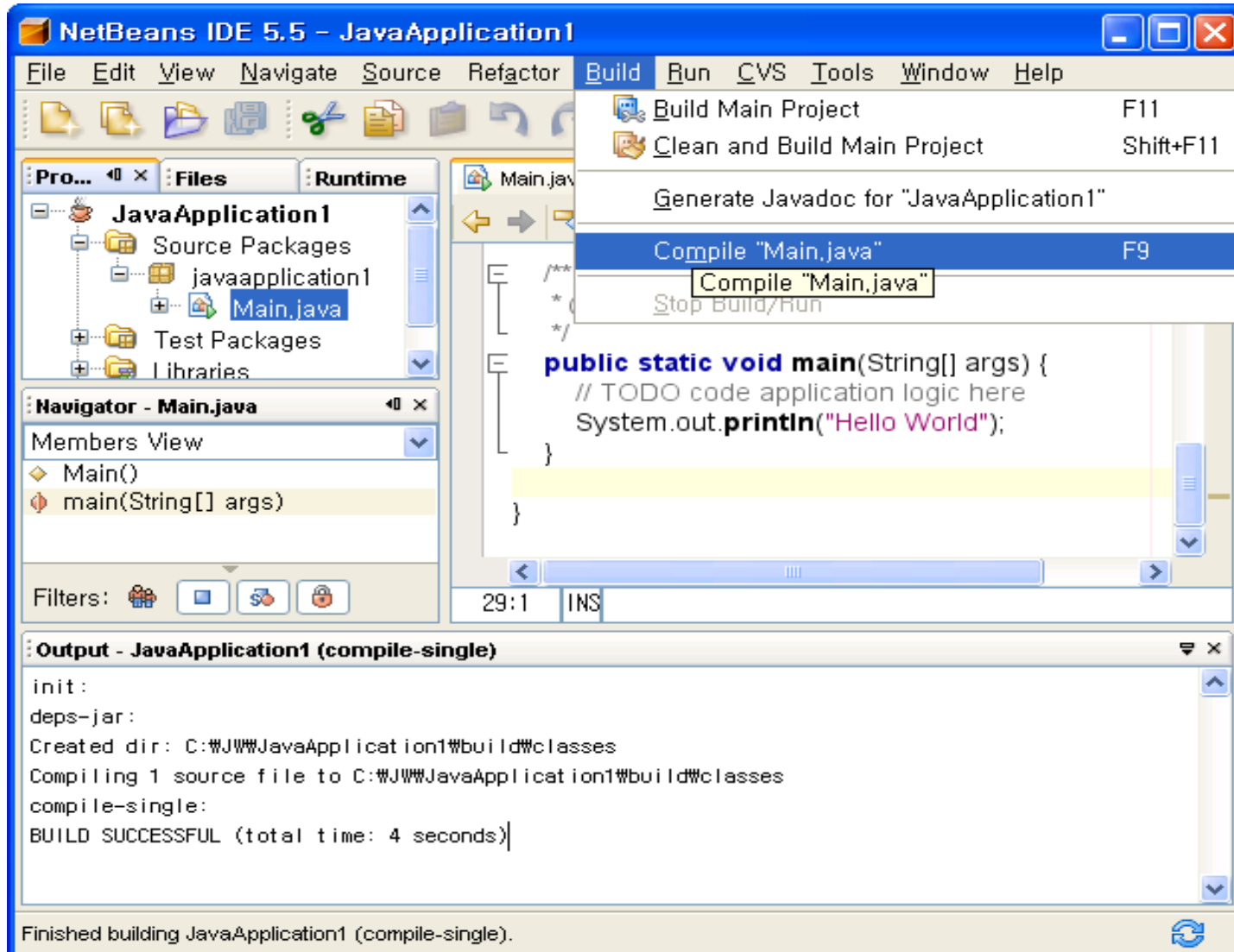
- `obj` - The Object to be printed

See Also:

- [java.lang.Object.toString\(\)](#)

JAVA Compiling

□ Menu [Build]-[Compile XXX.java]



The screenshot displays the NetBeans IDE 5.5 interface for a Java project named 'JavaApplication1'. The 'Build' menu is open, showing the following options:

- Build Main Project (F11)
- Clean and Build Main Project (Shift+F11)
- Generate Javadoc for "JavaApplication1"
- Compile "Main.java" (F9)** (highlighted)
- Compile "Main.java" (highlighted)
- Stop Build/Run

The 'Main.java' file is open in the editor, showing the following code:

```
/**  
 *  
 */  
  
public static void main(String[] args) {  
    // TODO code application logic here  
    System.out.println("Hello World");  
}
```

The 'Output' window at the bottom shows the compilation process:

```
init:  
deps-jar:  
Created dir: C:\J\JavaApplication1\build\classes  
Compiling 1 source file to C:\J\JavaApplication1\build\classes  
compile-single:  
BUILD SUCCESSFUL (total time: 4 seconds)
```

At the bottom of the IDE, a status bar message reads: 'Finished building JavaApplication1 (compile-single)'.

JAVA Run

□ Menu [Run]-[Run Main Project]

The screenshot displays the NetBeans IDE 5.5 interface for a Java project named 'JavaApplication1'. The 'Run' menu is open, showing the following options:

- Run Main Project (F6)
- Debug Main Project (F5)
- Test "JavaApplication1" (Alt+F6)
- Run File
- Attach Debugger...
- Finish Debugger Session (Shift+F5)
- Pause
- Continue (Ctrl+F5)
- Step Over (F8)
- Step Into (F7)
- Step Out (Ctrl+F7)
- Run to Cursor (F4)
- Run Into Method (Shift+F7)
- Apply Code Changes
- Stack
- Toggle Breakpoint (Ctrl+F8)
- New Breakpoint...
- New Watch... (Ctrl+Shift+F7)
- Evaluate Expression... (Ctrl+F9)

The 'Output' window at the bottom shows the following text:

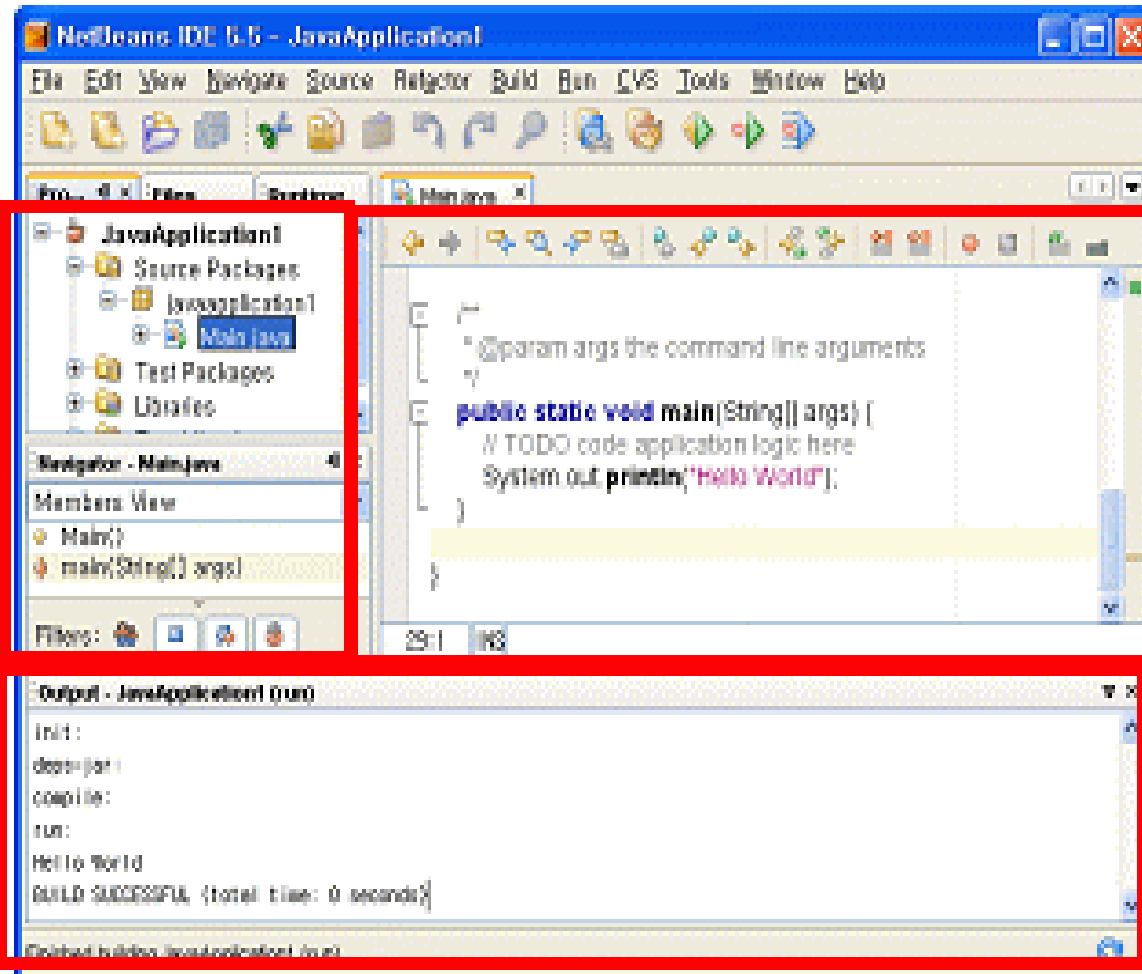
```
init:  
deps-jar:  
compile:  
run:  
Hello World  
BUILD SUCCESSFUL (total time: 0 seconds)
```

The status bar at the bottom indicates: 'Finished building JavaApplication1 (run).'

넷빈의 화면 구성

작업 공간

코드 창



출력 결과 창

CONCLUDE

- General Architecture of JAVA
- Configuration
- Execution Environment of JAVA using EditPlus tool
- JAVA Development Environment
- JAVA Integration Environment
- Application Program and Applet Program
- JAVA Application
- Download Netbean and installation