

Java



자바 시작하기

자바란 무엇인가?

JDK의 다운로드 및 설치 방법

Hello, Java 프로그램의 작성

자바 프로그램의 작동 원리



❖ Java Introduction :

● Java의 탄생

- ➡ 1991년경 Sun Microsystems의 Green project라는 연구진으로부터 'Oak' 라는 language가 개발되면서 Java가 시작 되었다.

❖ Java의 특징

- 배우기 쉽다.
- 객체지향 프로그래밍
- Garbage Collection
- 플랫폼 독립성
- MultiThread

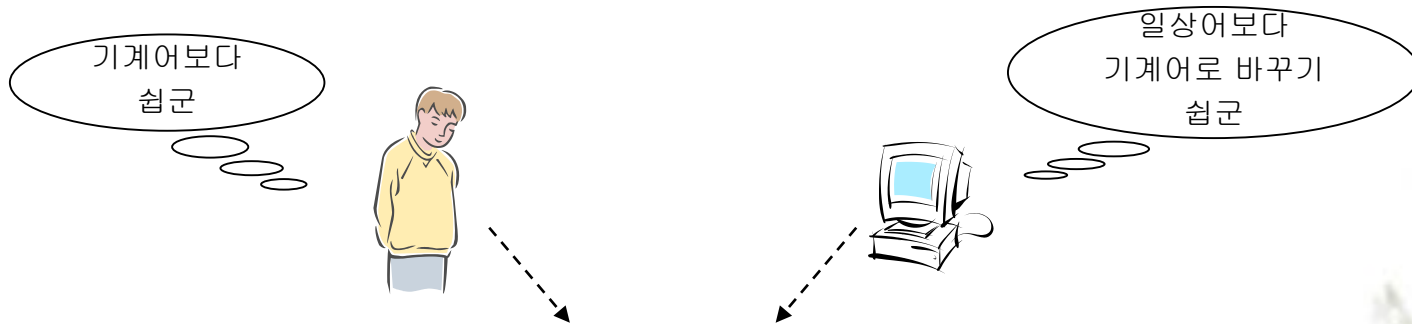


자바 시작하기

01. 자바란 무엇인가?

객체지향 프로그래밍 언어

- 객체지향 : ?
- 프로그래밍 언어란 : 사람과 컴퓨터의 의사소통 수단이 되는 언어



```
class Adder {  
    public static void main(String args[]) {  
        int total = 0;  
        for (int cnt = 1; cnt <= 100; cnt++)  
            total += cnt;  
        System.out.println(total);  
    }  
}
```

자바 시작하기

01. 자바란 무엇인가?

프로그램, 컴파일, 컴파일러

프로그래밍 언어로 작성된 프로그램

```
class Adder {  
    public static void main(String args[]) {  
        int total = 0;  
        for (int cnt = 1; cnt <= 100; cnt++)  
            total += cnt;  
        System.out.println(total);  
    }  
}
```



컴파일러



기계어 프로그램

```
0000010011010010100  
1111010100110101001  
0100010101001010101  
0011011111010100101  
0100010101001010101  
0100001010010101010  
010101111101010101
```

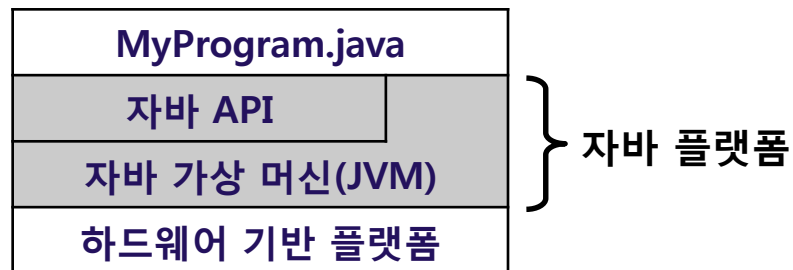


❖ Java Introduction :

● Java Platform

- ▶ Java Program이 실행되기 위해 제공되는 'Hardware적 Program'

❖ Java Platform의 구성



❖ Java Platform의 종류

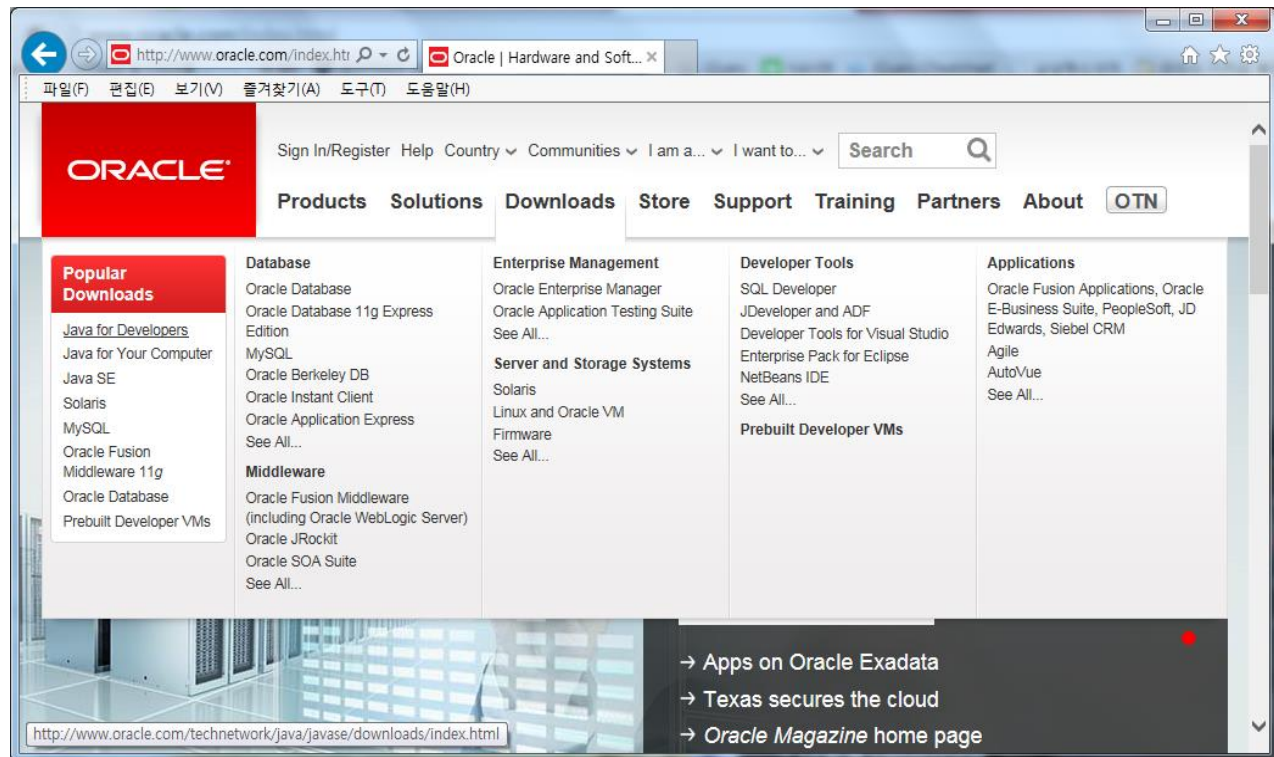
- Java 2 Platform, Standard Edition(J2SE)
- Java 2 Platform, Enterprise Edition(J2EE)
- Java 2 Platform, Micro Edition(J2ME)

자바 시작하기

02. JDK 설치하기

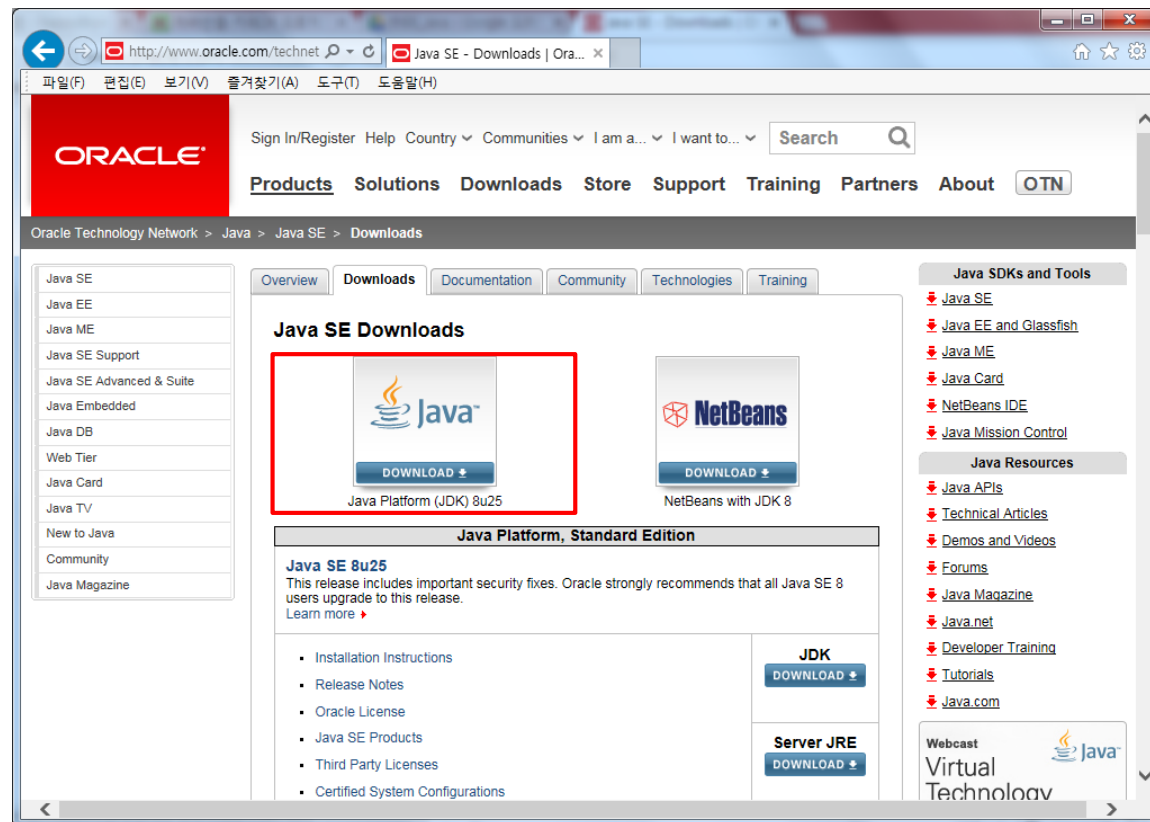
JDK의 다운로드

- ▶ • JDK : 자바 프로그램을 컴파일하고 실행하는데 필요한 소프트웨어
- ▶ • <http://www.oracle.com/>



자바 시작하기

- 02. JDK 설치하기
- JDK의 다운로드 (2)

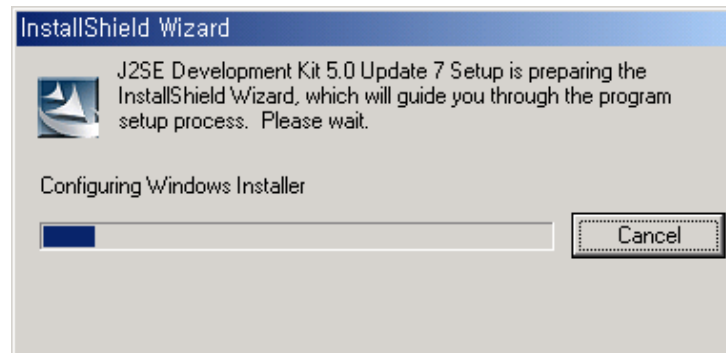


자바 시작하기

02. JDK 설치하기

설치 프로그램 (버전은 변경 될 수 있음)

- 1. 다운로드 받은 EXE 파일을 더블 클릭

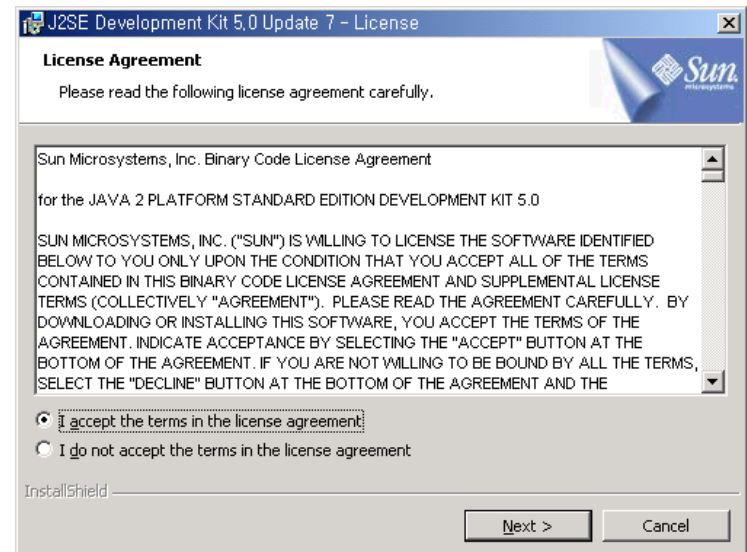
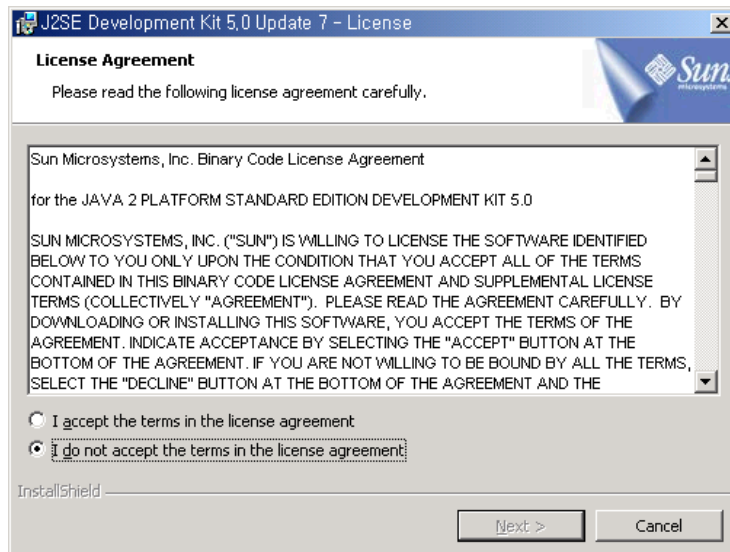


자바 시작하기

02. JDK 설치하기

설치 프로그램

➡ 2. License Agreement에 동의하기

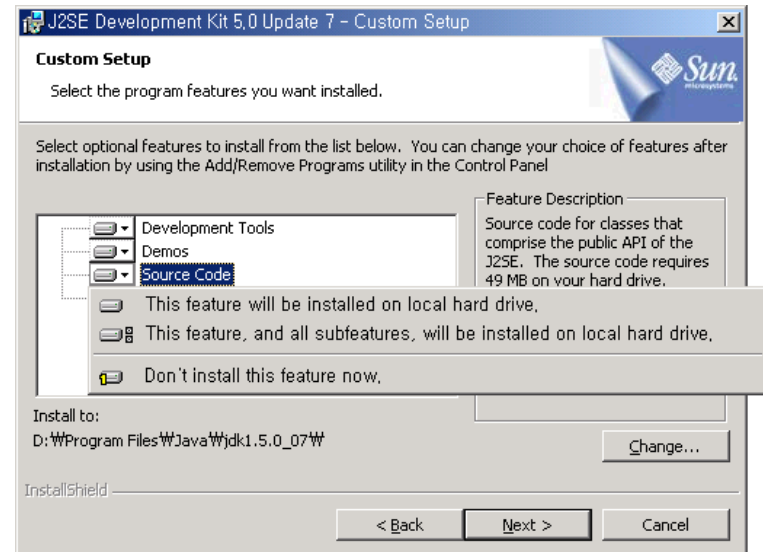
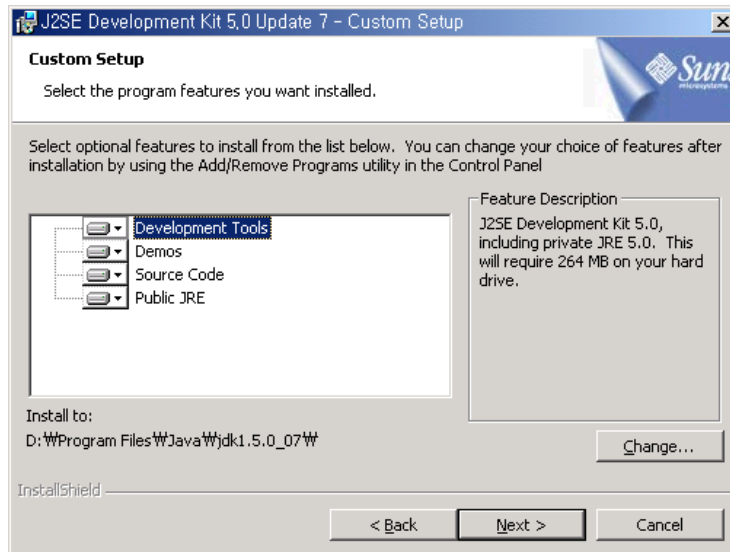


자바 시작하기

02. JDK 설치하기

설치 프로그램

3. 설치 항목 선택하기

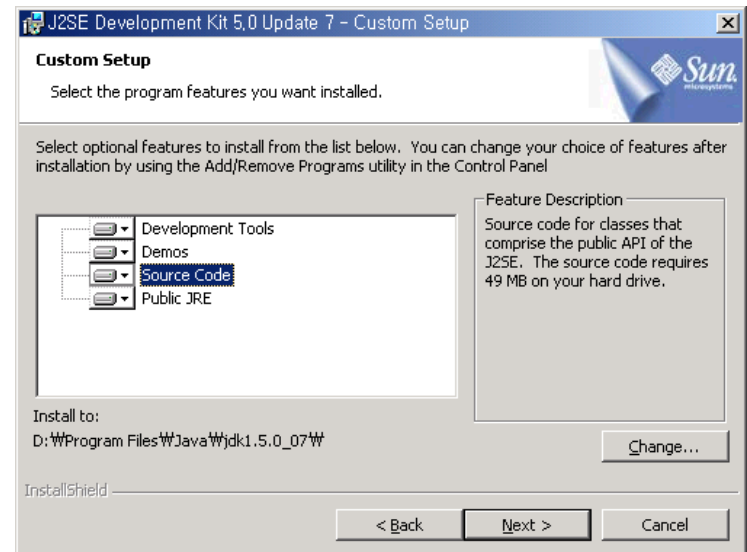
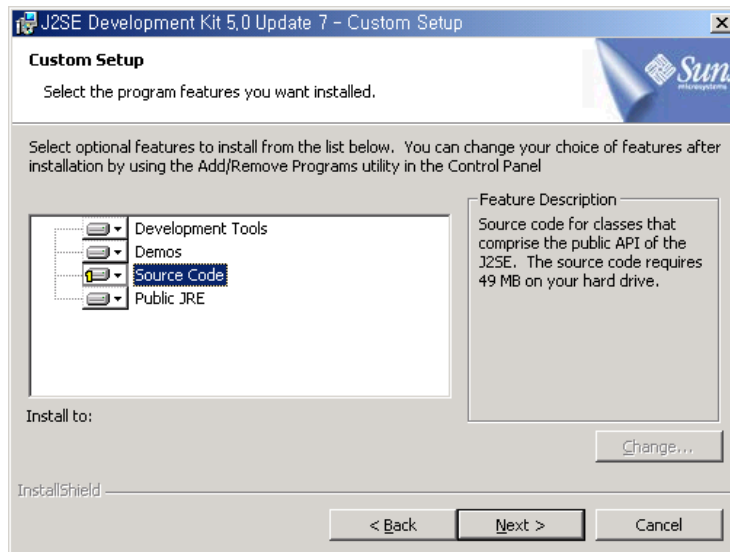


자바 시작하기

02. JDK 설치하기

설치 프로그램

➡ 3. 설치 항목 선택하기 (계속)

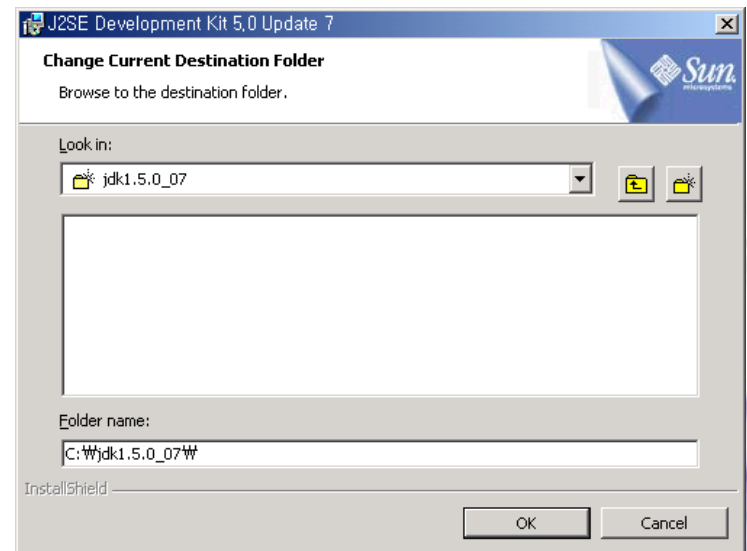
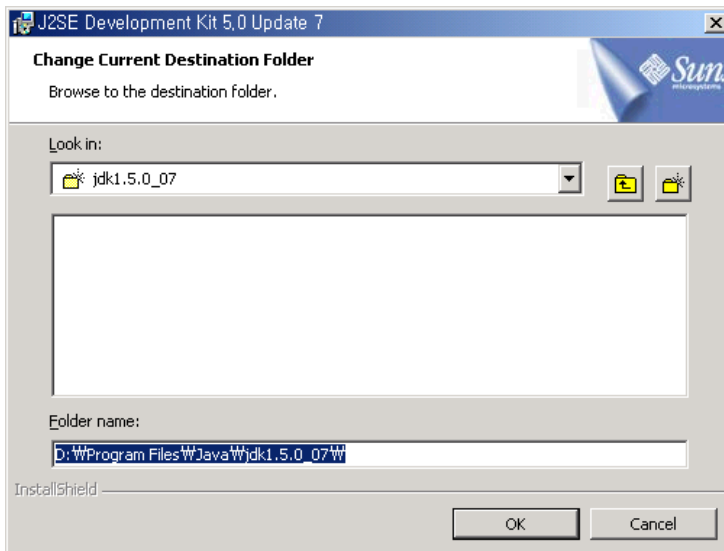


자바 시작하기

02. JDK 설치하기

설치 프로그램

➡ 4. 설치 디렉토리 선택하기

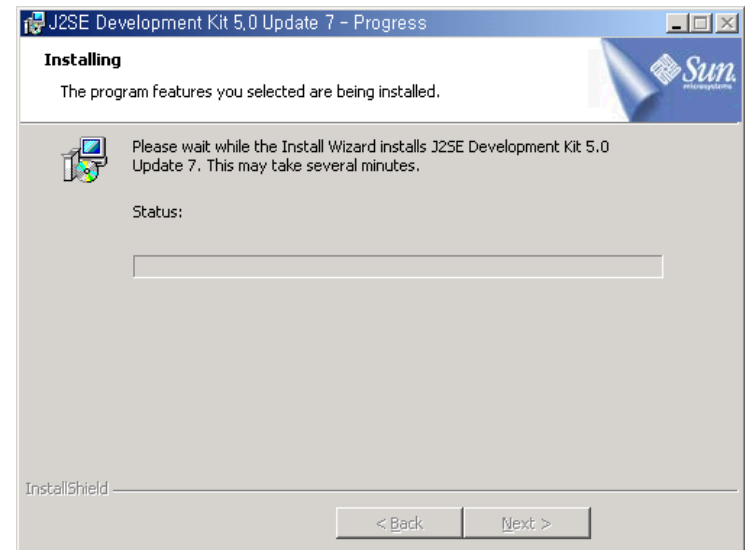
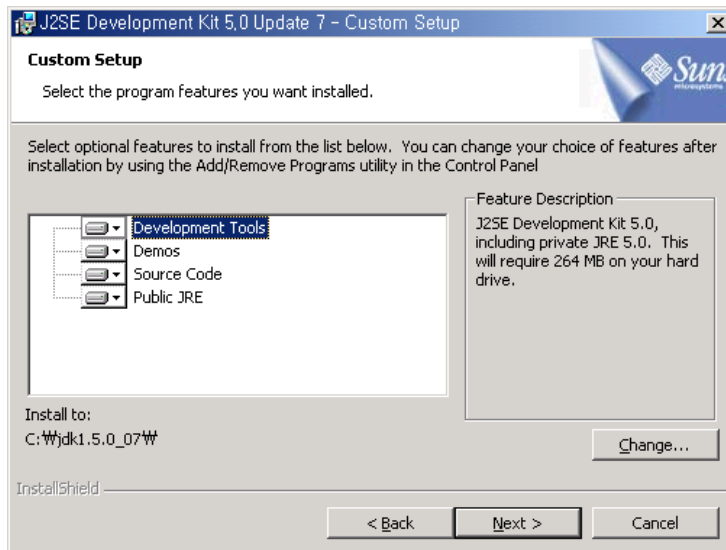


자바 시작하기

02. JDK 설치하기

설치 프로그램

➡ 5. 설치의 진행

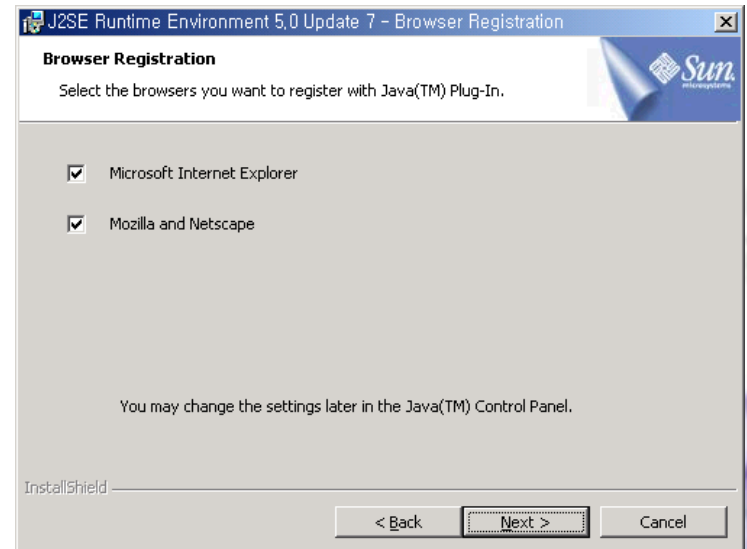
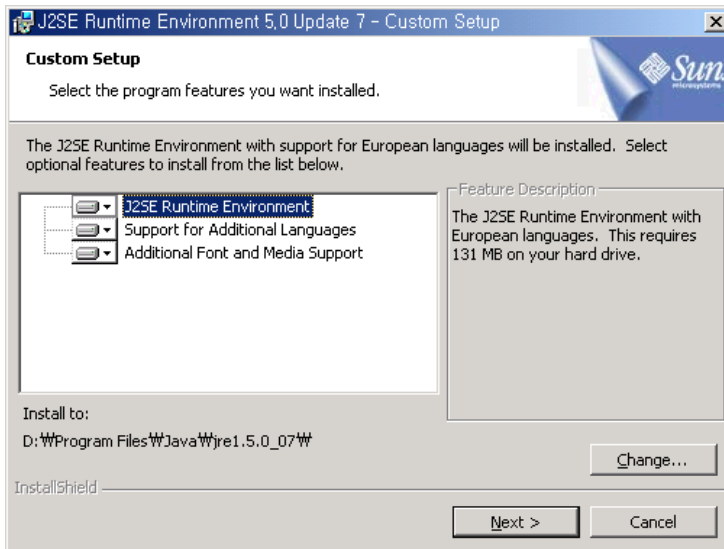


자바 시작하기

02. JDK 설치하기

설치 프로그램

6. Public JRE의 설치

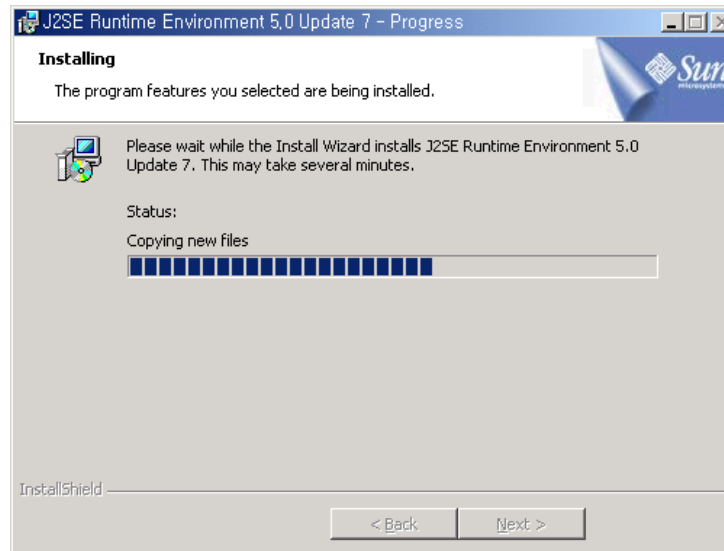


자바 시작하기

02. JDK 설치하기

설치 프로그램

➡ 6. Public JRE의 설치 (계속)

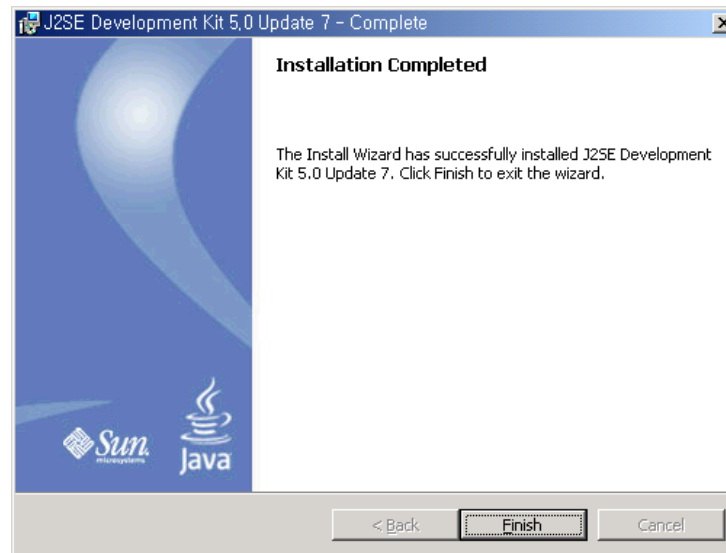


자바 시작하기

02. JDK 설치하기

설치 프로그램

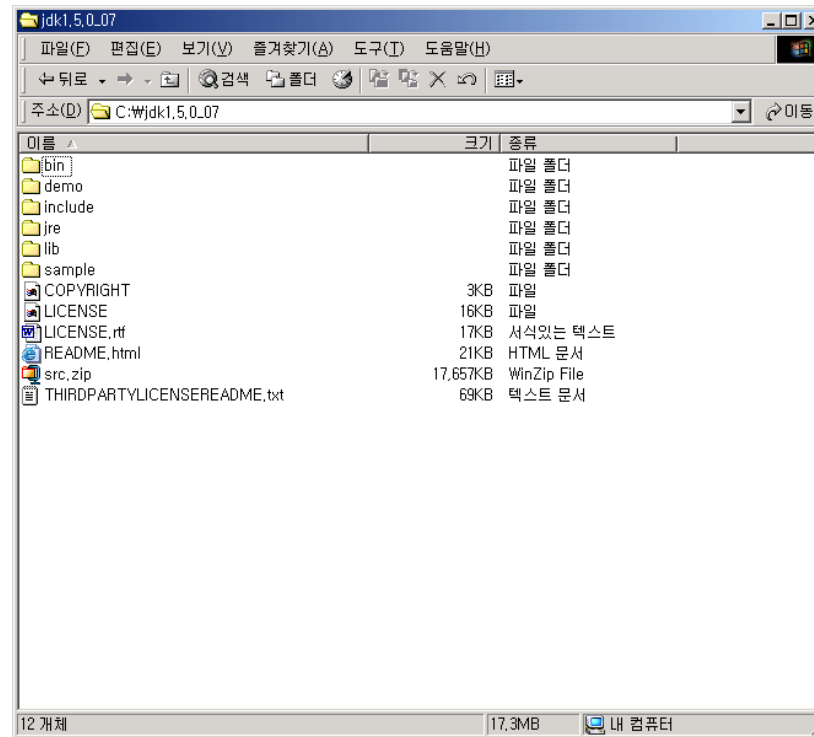
➡ 7. 설치 프로그램의 마지막 윈도우



자바 시작하기

02. JDK 설치하기

JDK 설치 디렉토리 구조

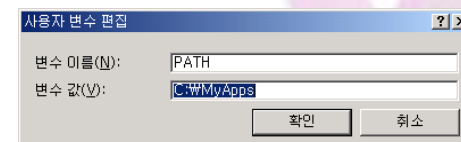
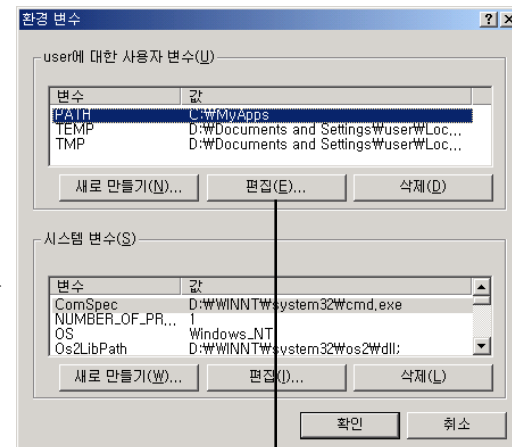
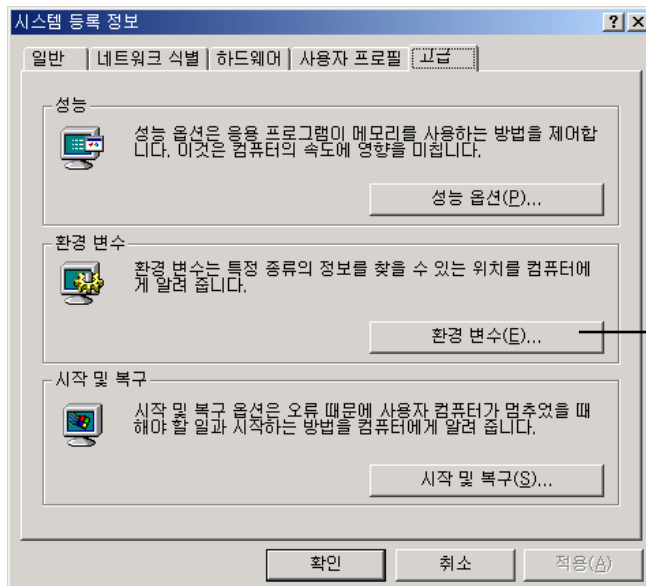


자바 시작하기

02. JDK 설치하기

환경 변수의 설정

- ➡ • Windows 2000 또는 Windows XP의 경우
- ➡ Window 7 환경 설정이 조금 다름

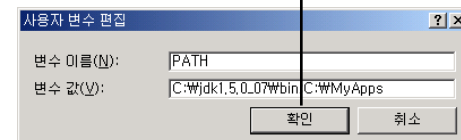
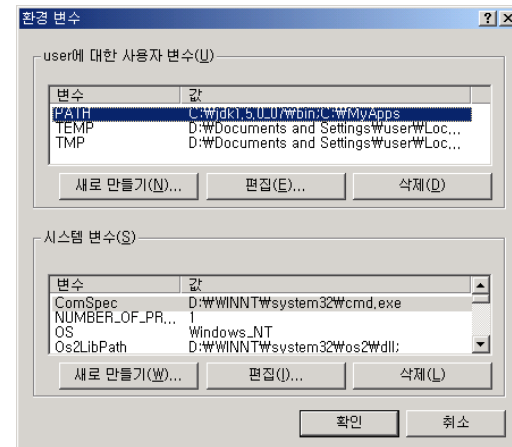
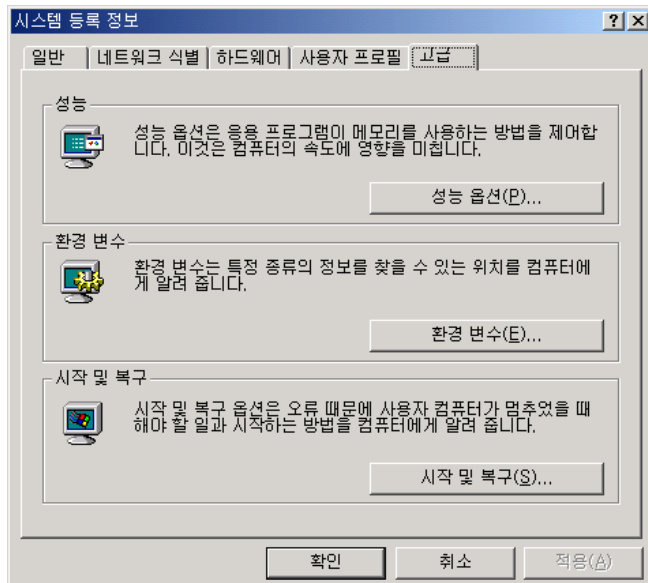


자바 시작하기

02. JDK 설치하기

환경 변수의 설정

- Windows 2000 또는 Windows XP의 경우 (계속)

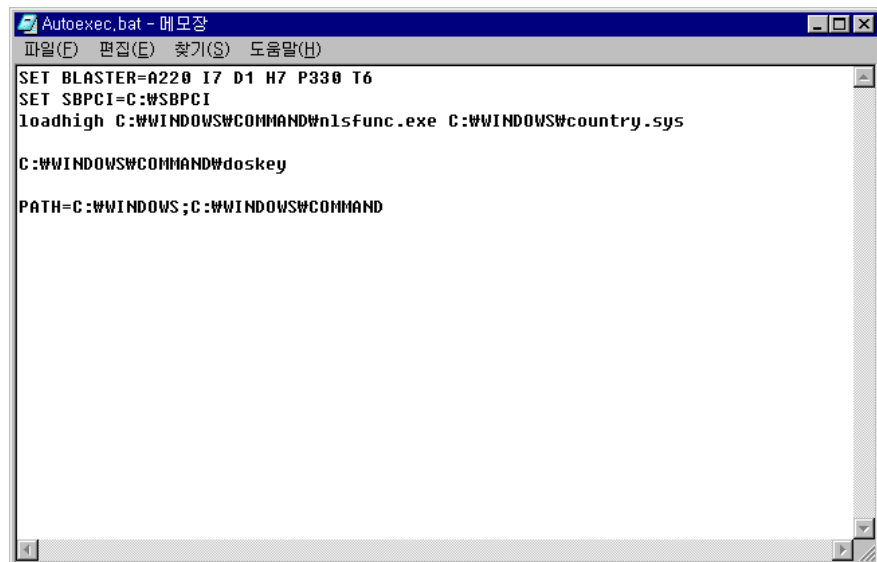
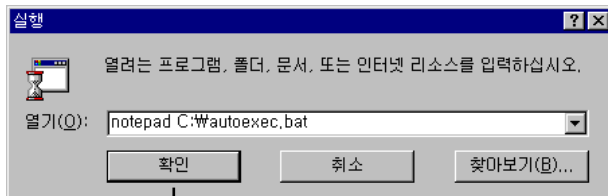


자바 시작하기

02. JDK 설치하기

환경 변수의 설정

- Windows 98의 경우

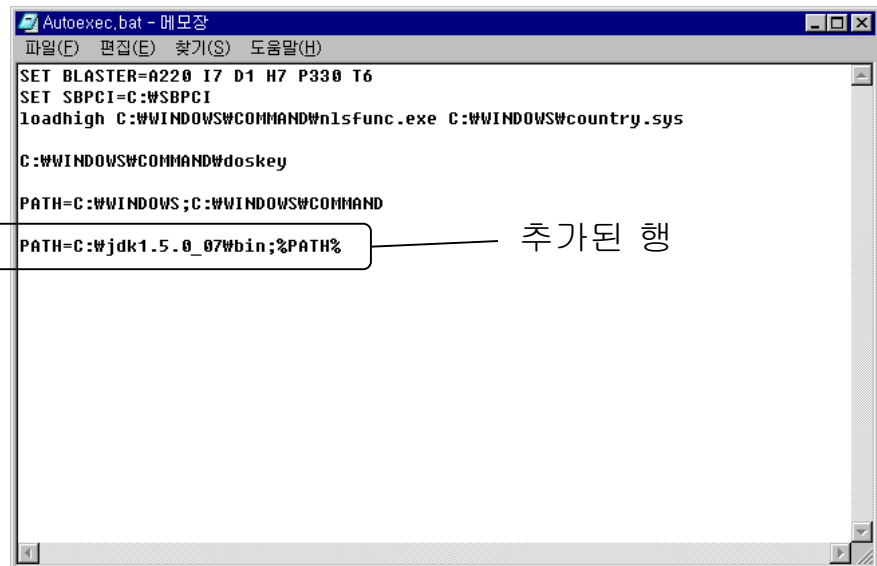
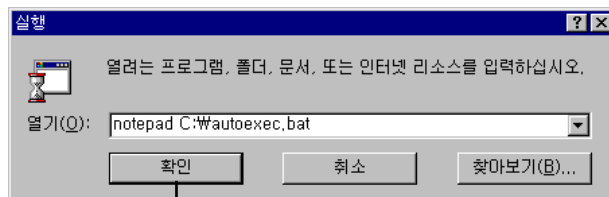


자바 시작하기

02. JDK 설치하기

환경 변수의 설정

➡ • Windows 98의 경우 (계속)

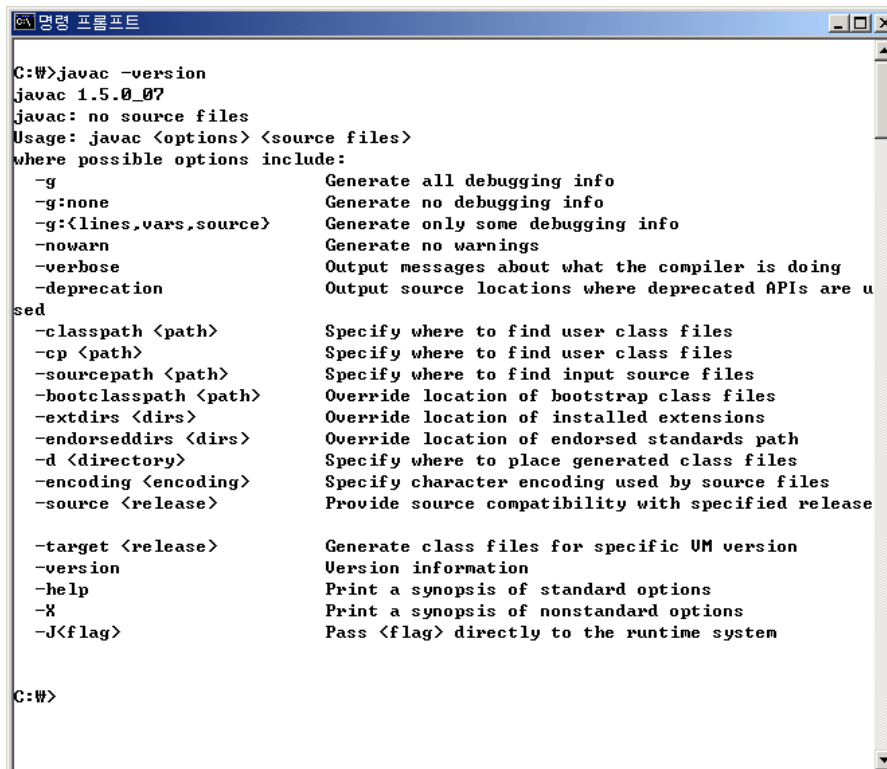


자바 시작하기

02. JDK 설치하기

JDK 버전 확인

- 자바 컴파일러의 버전 확인



```
C:\#>javac -version
javac 1.5.0_07
javac: no source files
Usage: javac <options> <source files>
where possible options include:
    -g                Generate all debugging info
    -g:none           Generate no debugging info
    -g:{lines,vars,source} Generate only some debugging info
    -nowarn           Generate no warnings
    -verbose          Output messages about what the compiler is doing
    -deprecation      Output source locations where deprecated APIs are used
    -classpath <path> Specify where to find user class files
    -cp <path>        Specify where to find user class files
    -sourcepath <path> Specify where to find input source files
    -bootclasspath <path> Override location of bootstrap class files
    -extdirs <dirs>    Override location of installed extensions
    -endorseddirs <dirs> Override location of endorsed standards path
    -d <directory>    Specify where to place generated class files
    -encoding <encoding> Specify character encoding used by source files
    -source <release> Provide source compatibility with specified release
    -target <release> Generate class files for specific VM version
    -version          Version information
    -help            Print a synopsis of standard options
    -X               Print a synopsis of nonstandard options
    -J<flag>         Pass <flag> directly to the runtime system

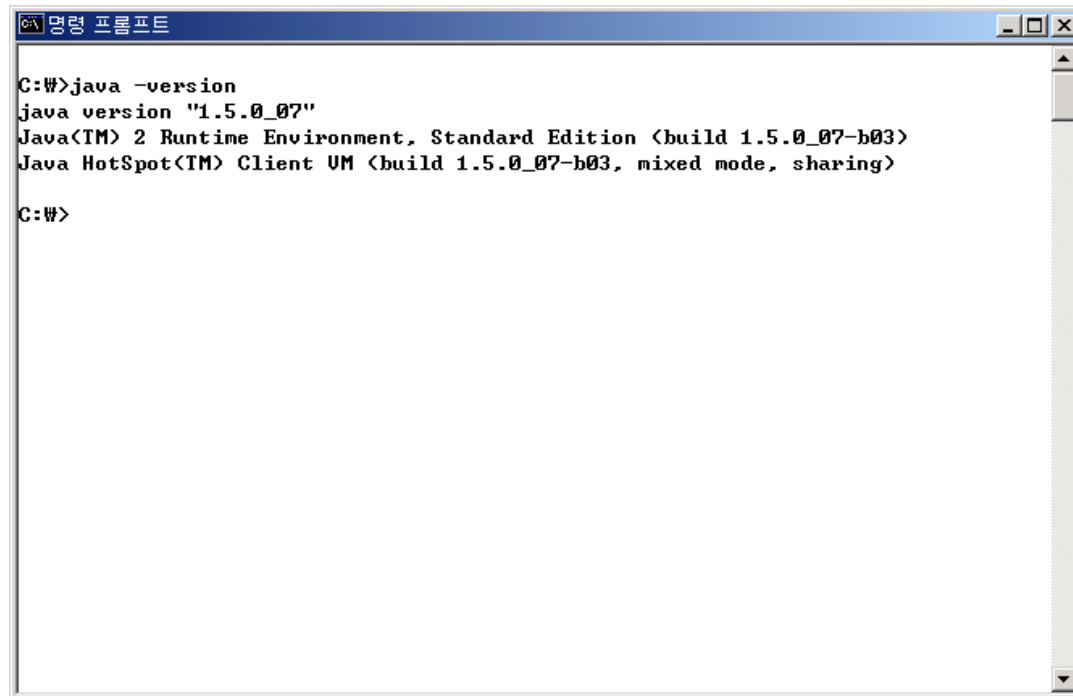
C:\#>
```

자바 시작하기

02. JDK 설치하기

JDK 버전 확인

- java.exe의 버전 확인



```
C:\>java -version
java version "1.5.0_07"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0_07-b03)
Java HotSpot(TM) Client VM (build 1.5.0_07-b03, mixed mode, sharing)
C:\>
```

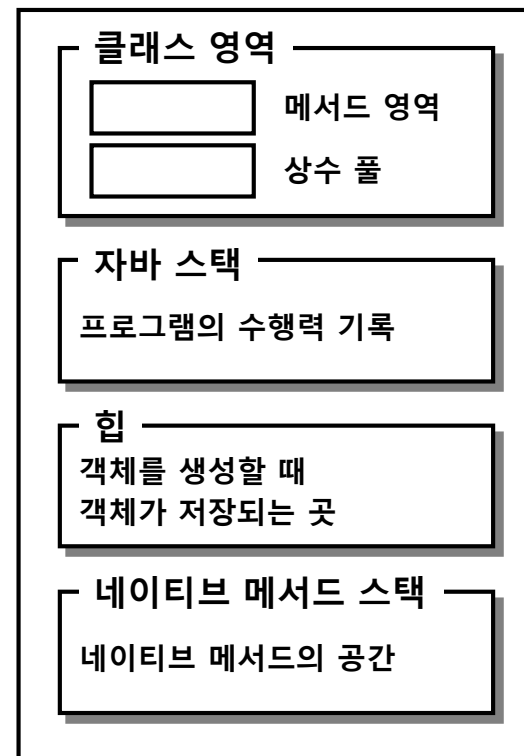

❖ Java Introduction :

❖ JVM(Java Virtual Machine)

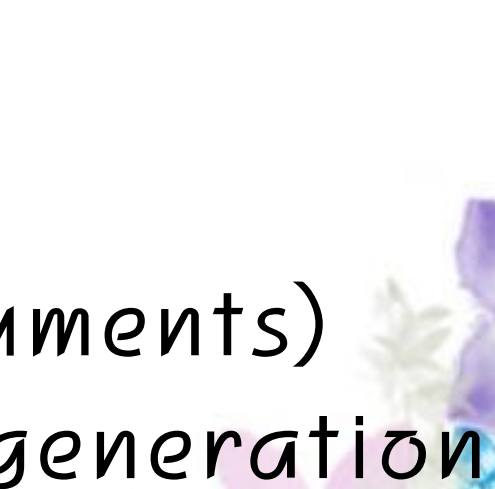
- ▶ Java Compiler에 의해 만들어진 byte code를 해석하고 실행하는 'Virtual Machine(CPU)'

❖ JVM의 개략적 구성

- Class Area
- Java Stack
- Heap
- Native Method Stacks



❖ JDK 5.0 Add-on.

- Generic Type
 - Meta Data
 - 개선된 루프
 - AutoBoxing/UnBoxing
 - static import
 - Formatter와 Scanner
 - Varargs(Variable Arguments)
 - Simpler RMI Interface generation
- 

03. Hello, Java 프로그램

소스 코드(source code)

```
1  class HelloJava {  
2      public static void main(String args[]) {  
3          System.out.println("Hello, Java");  
4          System.out.println( "Firust, Java");  
5      }  
}
```

➡ • 이 프로그램이 하는 일:

Hello, Java
Firust,Java

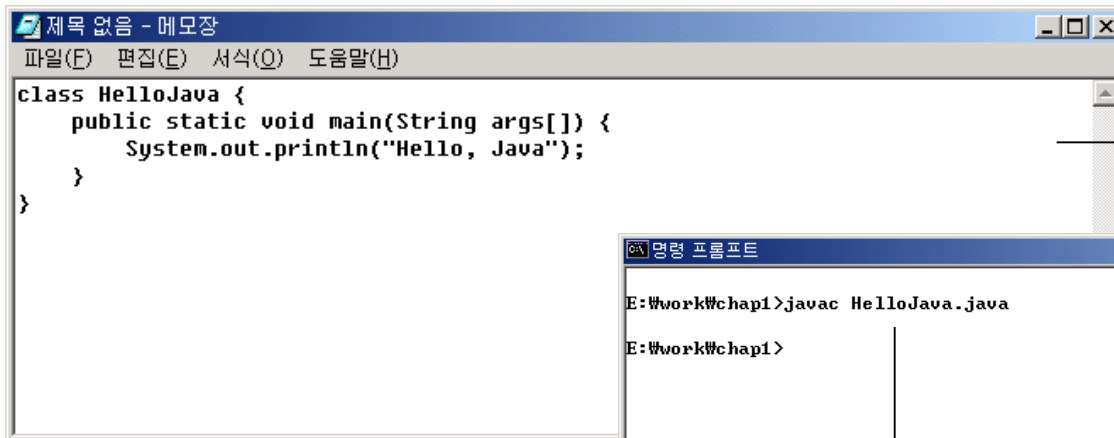
라고 모니터로 출력



자바 시작하기

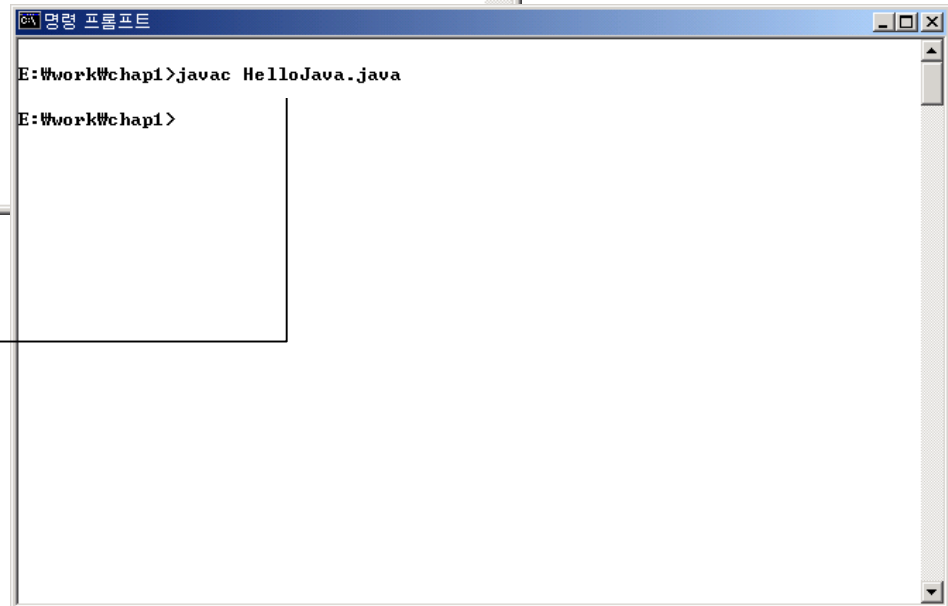
03. Hello, Java 프로그램

작성 및 컴파일



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

텍스트 에디터로 작성

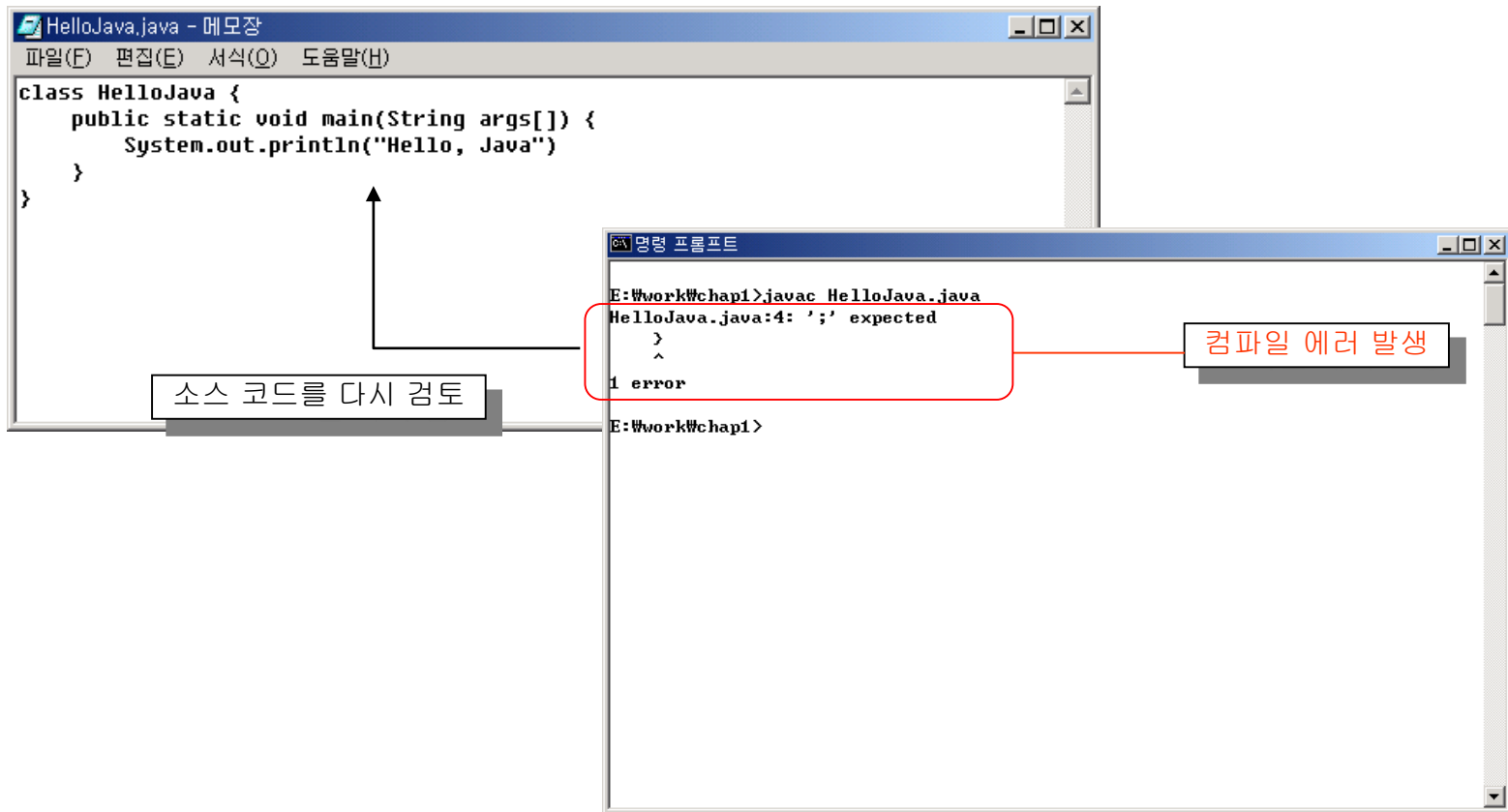


```
E:\work\chap1>javac HelloJava.java  
E:\work\chap1>
```

명령 프롬프트(도스) 창에서 컴파일

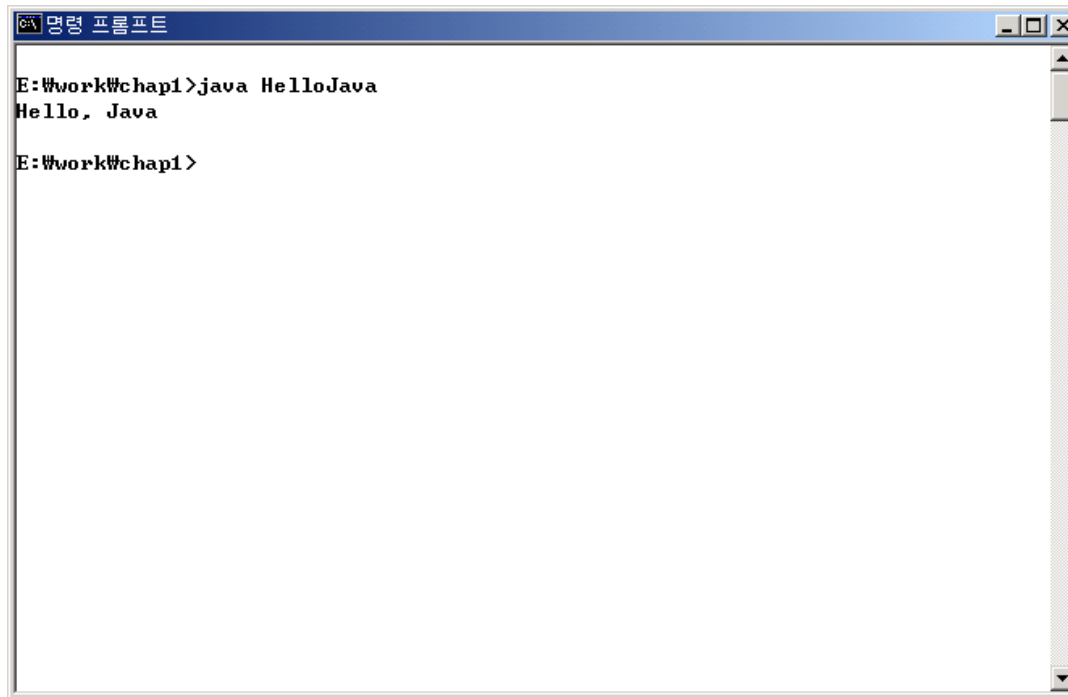
자바 시작하기

- 03. Hello, Java 프로그램
- 컴파일 에러가 있는 경우



자바 시작하기

- 03. Hello, Java 프로그램
- 프로그램의 실행



```
명령 프롬프트
E:\work\chap1>java HelloJava
Hello, Java
E:\work\chap1>
```

자바 시작하기

03. Hello, Java 프로그램

소스 코드 다시 살펴보기

```
1  class HelloJava {  
2      public static void main(String args[]) {  
3          System.out.println("Hello, Java");  
4      }  
5  }
```

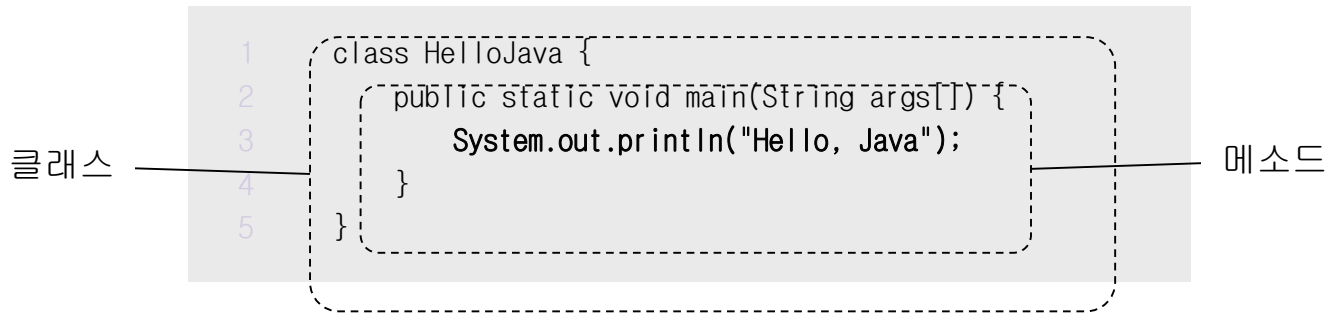
출력할 내용

명령문

자바 시작하기

03. Hello, Java 프로그램

소스 코드 다시 살펴보기



자바 시작하기

03. Hello, Java 프로그램

소스 코드 다시 살펴보기

클래스 이름

```
1 class HelloJava {  
2     public static void main(String args[]) {  
3         System.out.println("Hello, Java");  
4     }  
5 }
```

클래스 본체
(class body)



자바 시작하기

03. Hello, Java 프로그램

소스 코드 다시 살펴보기

메소드 이름

```
1  class HelloJava {  
2      public static void main(String args[]) {  
3          System.out.println("Hello, Java");  
4      }  
5  }
```

메소드 본체
(method body)

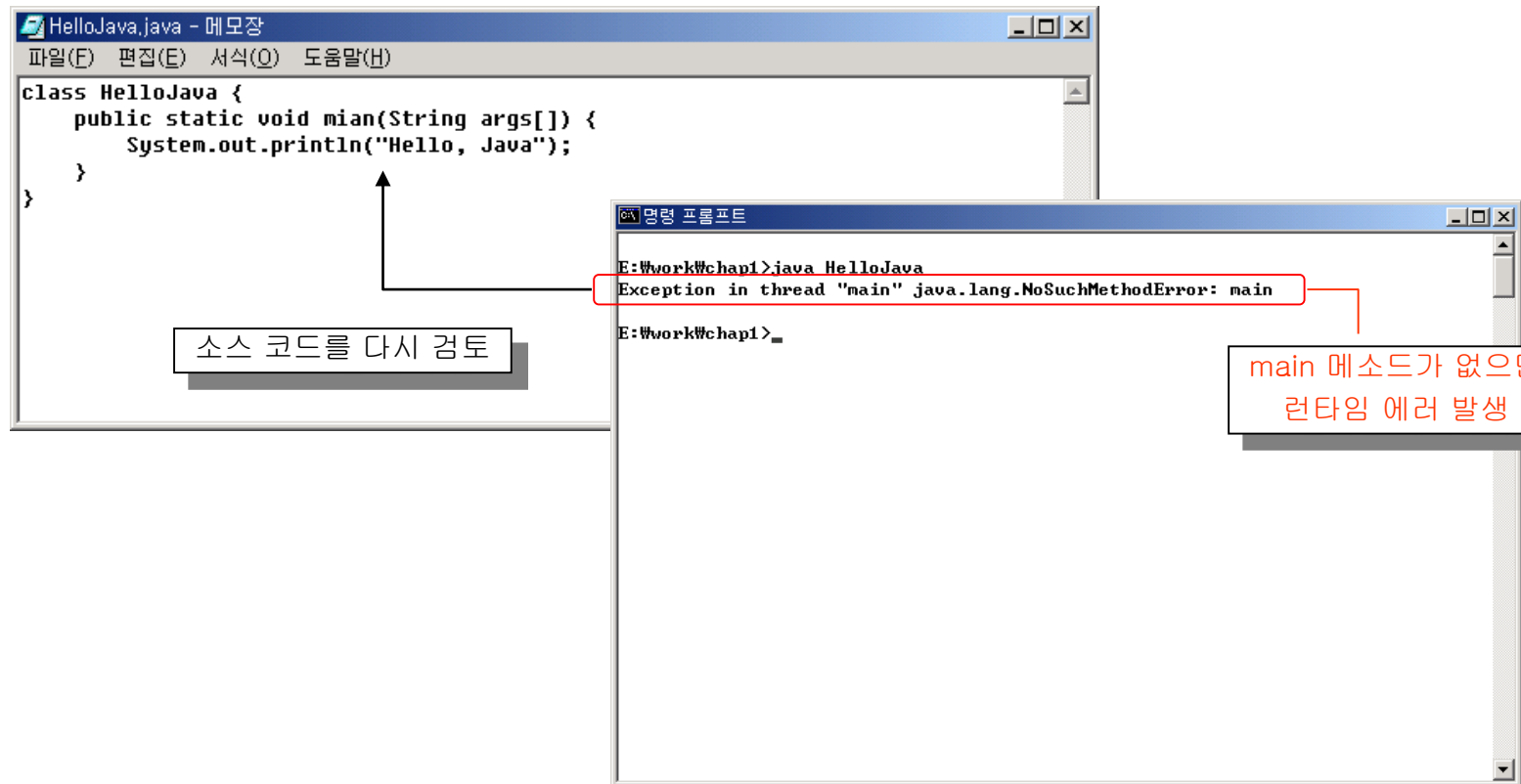


자바 시작하기

03. Hello, Java 프로그램

main이라는 이름의 메소드

- 프로그램의 실행 시작점이 되는 메소드

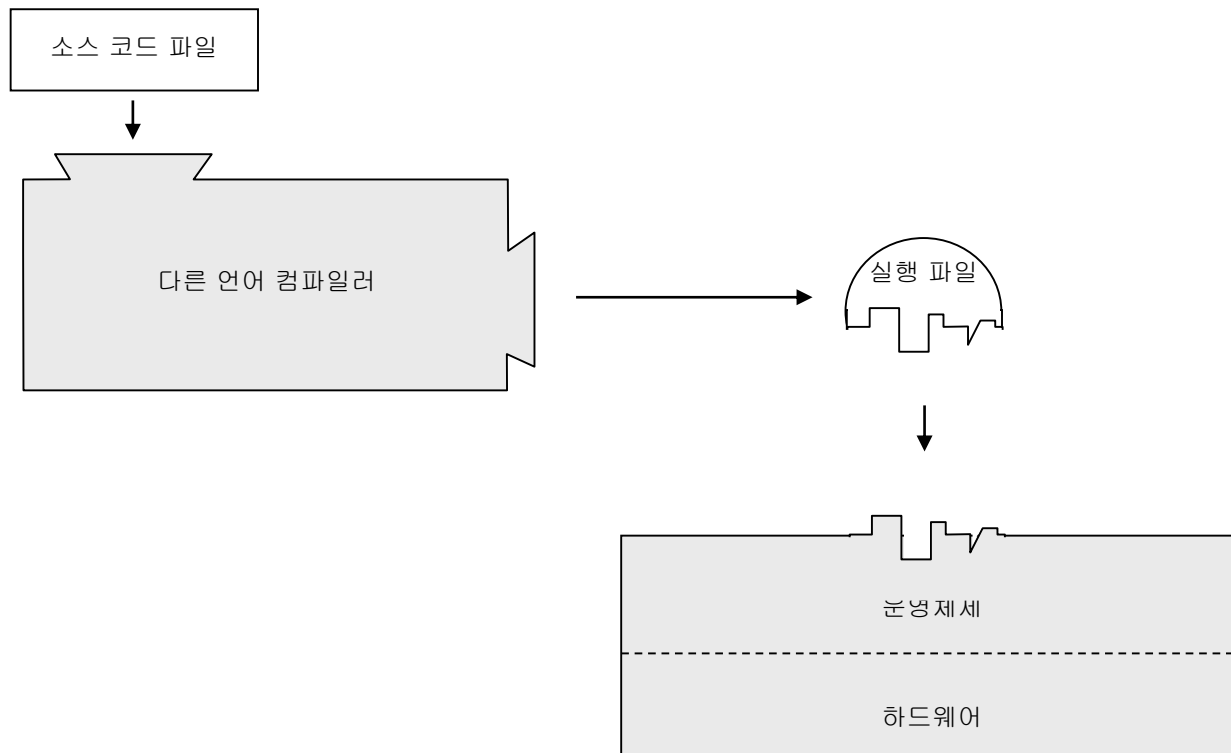


자바 시작하기

04. 자바 프로그램의 작동 원리

자바와 다른 언어의 차이점

- 다른 언어 프로그램의 컴파일과 실행

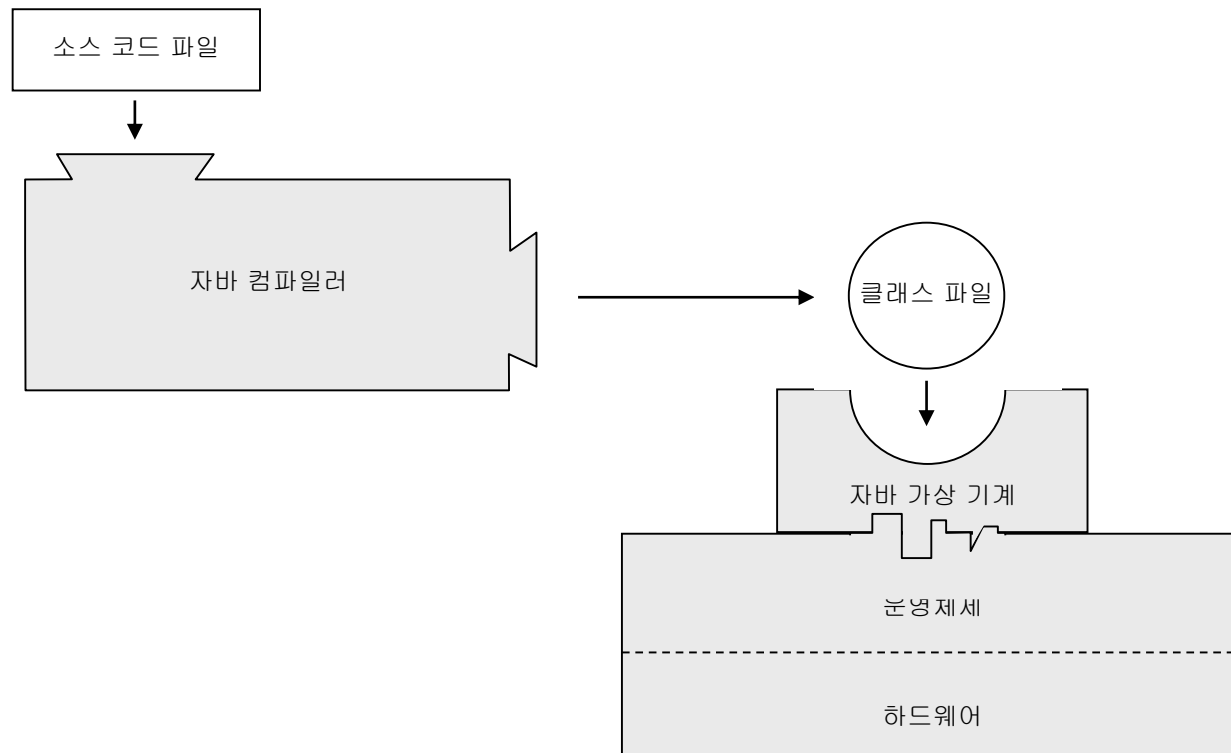


자바 시작하기

04. 자바 프로그램의 작동 원리

자바와 다른 언어의 차이점

- 자바 프로그램의 컴파일과 실행

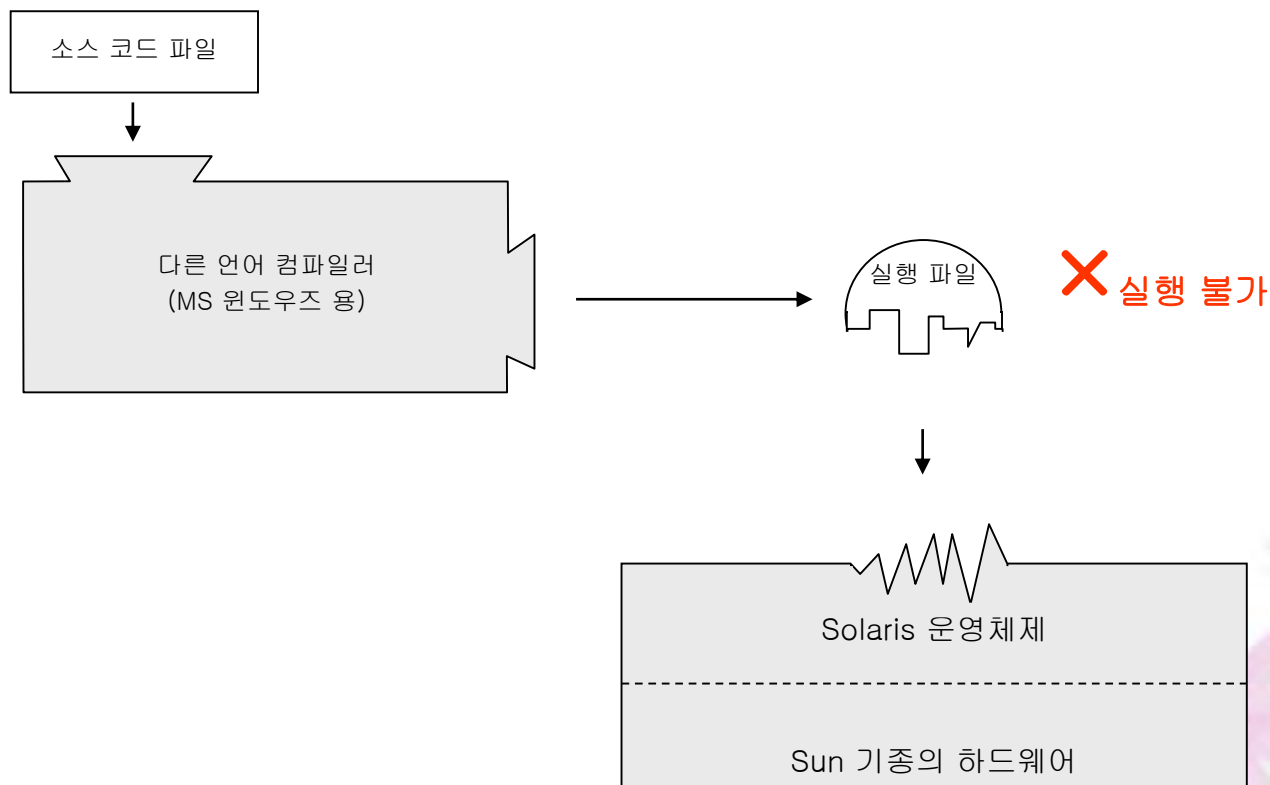


자바 시작하기

04. 자바 프로그램의 작동 원리

자바의 플랫폼 중립성

- 플랫폼 중립적이지 않은 다른 언어 프로그램

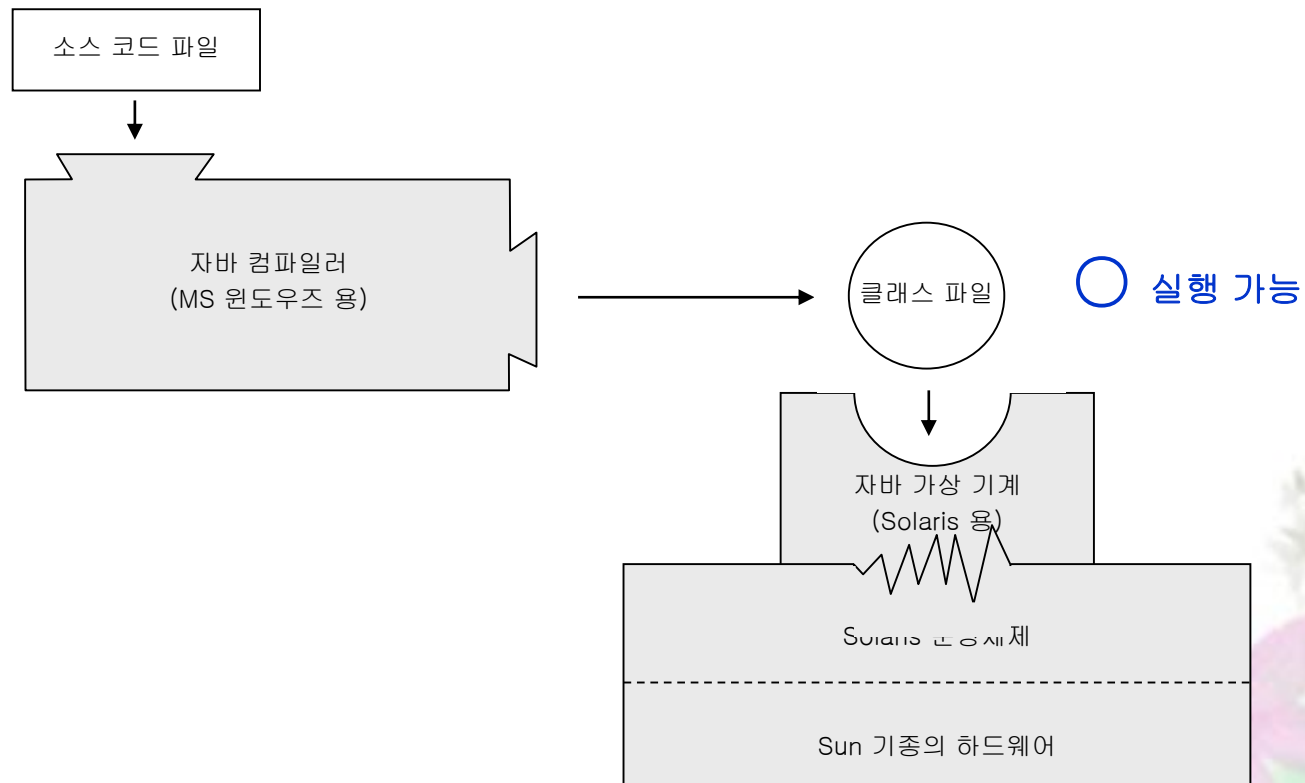


자바 시작하기

04. 자바 프로그램의 작동 원리

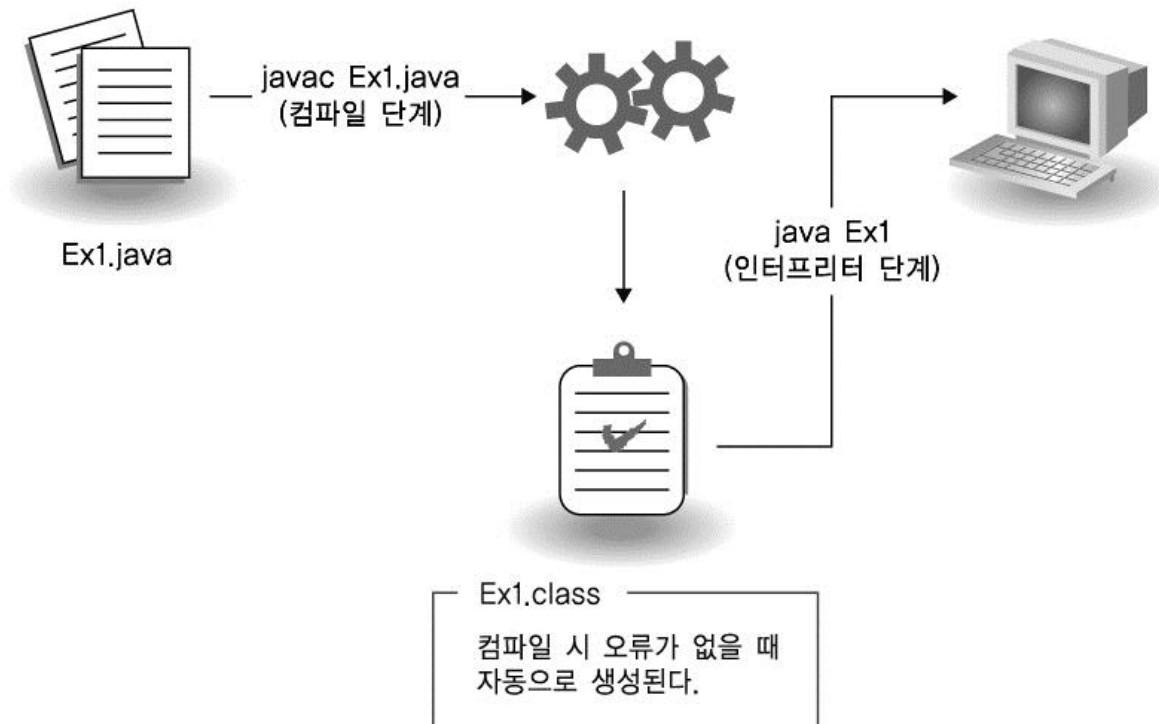
자바의 플랫폼 중립성

- 플랫폼 중립적인 자바 프로그램



❖ JDK5.0 Setting & Environment Setting

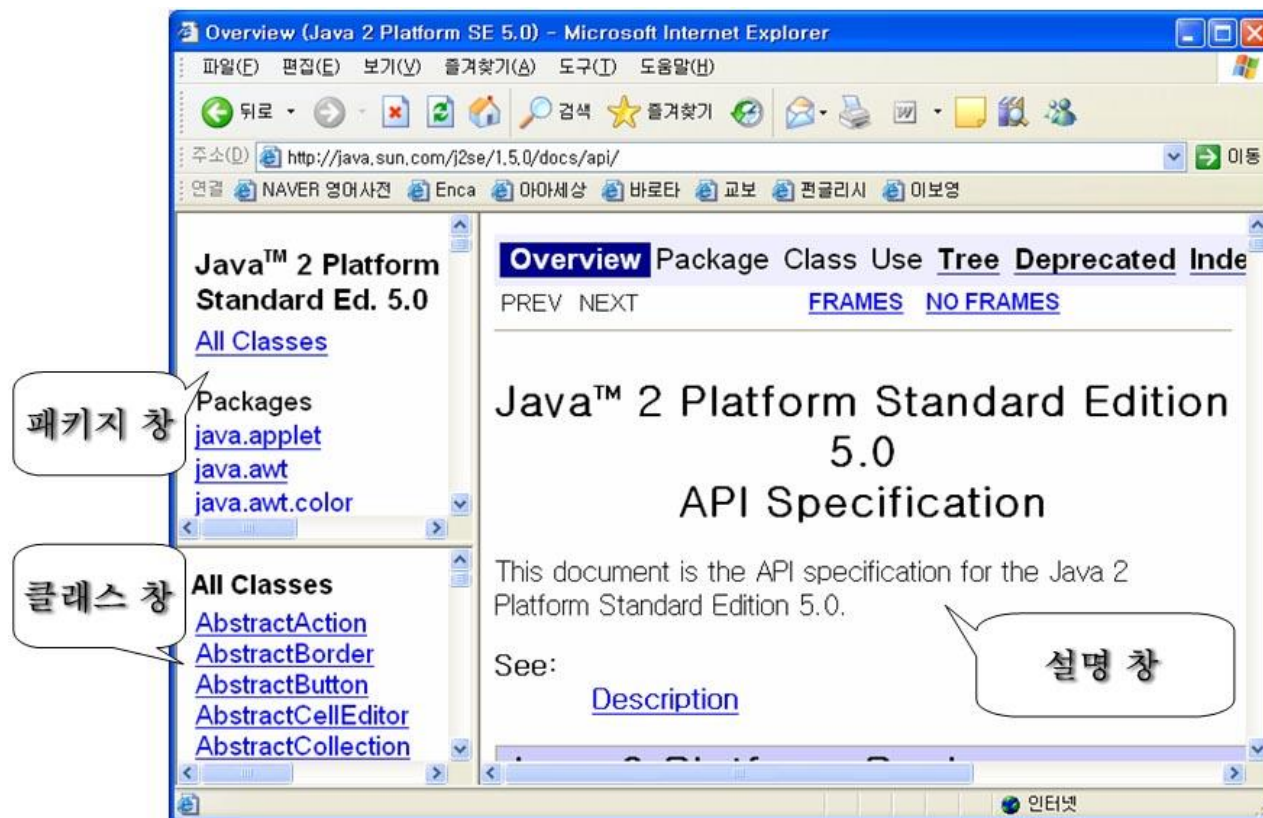
● Java Execution Structure



Java API :: J2SE API

Java API?

- ▶ JDK Setting시 제공되는 유용한 class들에 대한 사용 방법을 문서화 하여 제공해 주는 도움말.



정기

이

한

시

다

^^!