Chap 01. Getting Started

학습 목표

1. 자바 기술의 특징
2. JVM 기능, GC
3. Sample Program 작성
4. Write, Compile, Run

Java 의 역사

- 1991년 Sun사의 James Gosling에 의해 가전제품에 이용하기 위해 개발이 시작(Green Project).

- 초기에 개발된 언어를 Oak라 하였으며 전자기기의 내장된 프로그램을 위해 사용. Oak는 별로 관심을 끌지 못하였다.

- 1994년 Gosling은 Oak를 JAVA(커피의 속어)로 다시 명명하고 당시 인터넷에서 급격히 성장한 WWW에 자바를 적용 결정

- Hot Java 검색기 개발. 넷스케이프, 익스플로러 자바 지원

Java 의 역사

- 1991년
  - Green 프로젝트가 생기면서 Java의 모태가 탄생하기 시작
  - James Gosling, Mike Sheridan, Patrick Naughton이 TV와 시청자가 서로 상호작용을 하는 것을 만들기 위해 Oak 라는 언어 탄생
- 1994년
  - World Wide Web 등장
  - Oak에서 Java로 명칭변경
  - Java, Hot Java project 시작
- 1995년
  - Hot Java, Java, Java context, source code가 Web에 공개
  - 플랫폼 : Sun SPARC Solaris, Windows NT, Windows95, Linux
  - Java beta1 발표(Sun Microsystems)
  - Netscape 지원결정
  - Java beta2 발표
  - JavaScript 발표(Sun & Netscape)

Java 의 역사

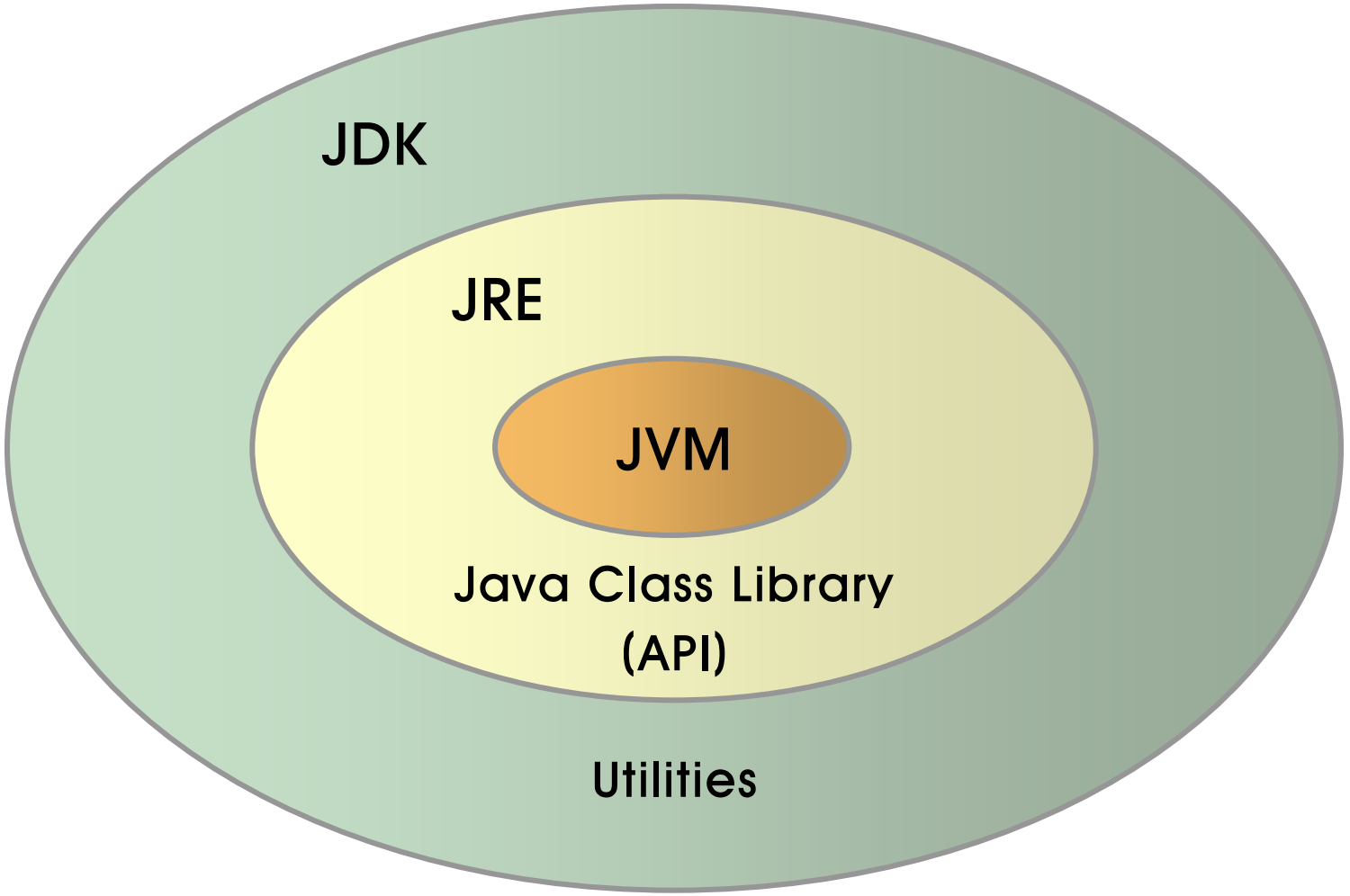
- 1996년
  - Java1.0 발표
  - Netscape2.0 Java 지원
- 2000년 J2SE 1.3 출시
- 2002년 J2SE 1.4 출시
- 2004년 J2SE 5.0 출시
  - Generic, foreach 루프, static import, Type Safe Enum
  - AutoBoxing/unBoxing, Concurrent API
- 2006년 J2SE 6 출시
  - JavaSE 6까지는 Sun Microsytems에서 Java에 대한 주요 스펙을 만듬
  - JAX-WS(Web Service Client), 모니터링 및 관리기능 강화, 스크립트언어지원
- 2011년 J2SE 7 출시
  - JavaSE 7부터는 Oracle이 Java에 대한 주요 스펙을 만듬
  - String을 이용한 switch 구문, NIO 2.0, Fork-Join에 의한 병렬처리
  - try-with-resources구문,
- 2014년 J2SE 8 출시
  - Lamda, 함수형 프로그래밍, Functional Interface, Stream, default method

- Programming Language
- Development Environment
- Application Environment
- Deployment Environment

JVM / JRE / JDK

Java Development Kit

사용자/개발자 입장에 따라 설치하는 범위가 달라진다.



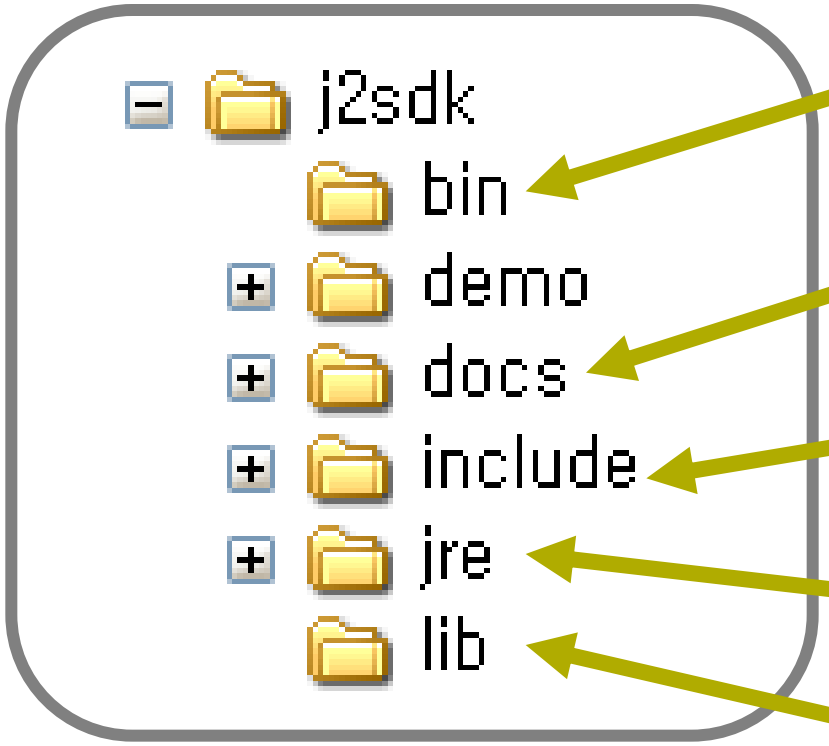
Java Platform 종류

J2SE : Java 2 Standard Edition

J2EE : Java 2 Enterprise Edition

J2ME : Java 2 Micro Edition

JDK 폴더 구성



컴파일, 실행, 디버그, 문서 제작에 관련된 개발 도구가 들어 있는 폴더

Java API Document 폴더

C 헤더 파일이 들어 있는 폴더

자바 실행 관련 API 가 들어 있는 폴더

보조 라이브러리가 들어 있는 폴더

Java Technology의 지향점

- 사용하기 쉬운 언어 제공
  - 다른 언어의 단점 보완 (포인터/메모리)
  - 객체 지향 언어
  - 능률적이고 명확한 코드를 작성하게 해 준다.
- Interpreted Environment
  - 개발 속도 향상
  - 코드 이식성 (bytecode)
- Thread 제공
- 클래스의 동적 메모리 Load
- 변경 된 class들을 원격지로부터 다운로드하여, Runtime시 반영 (Applet)
- 보다 나은 코드 안정성

Java 언어의 특징을 가능하게 해주는 요소

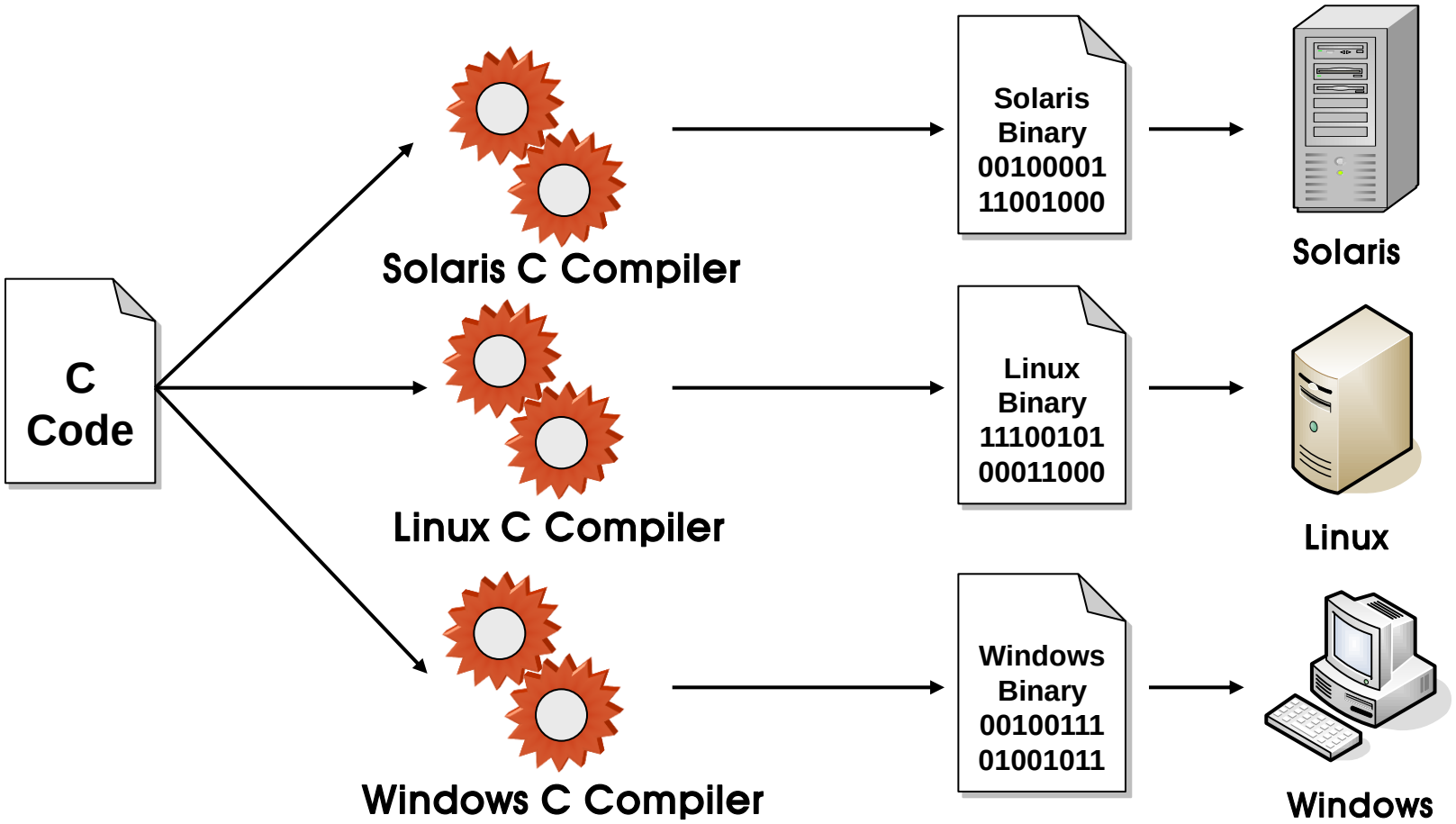
Java Virtual Machine

Garbage Collection

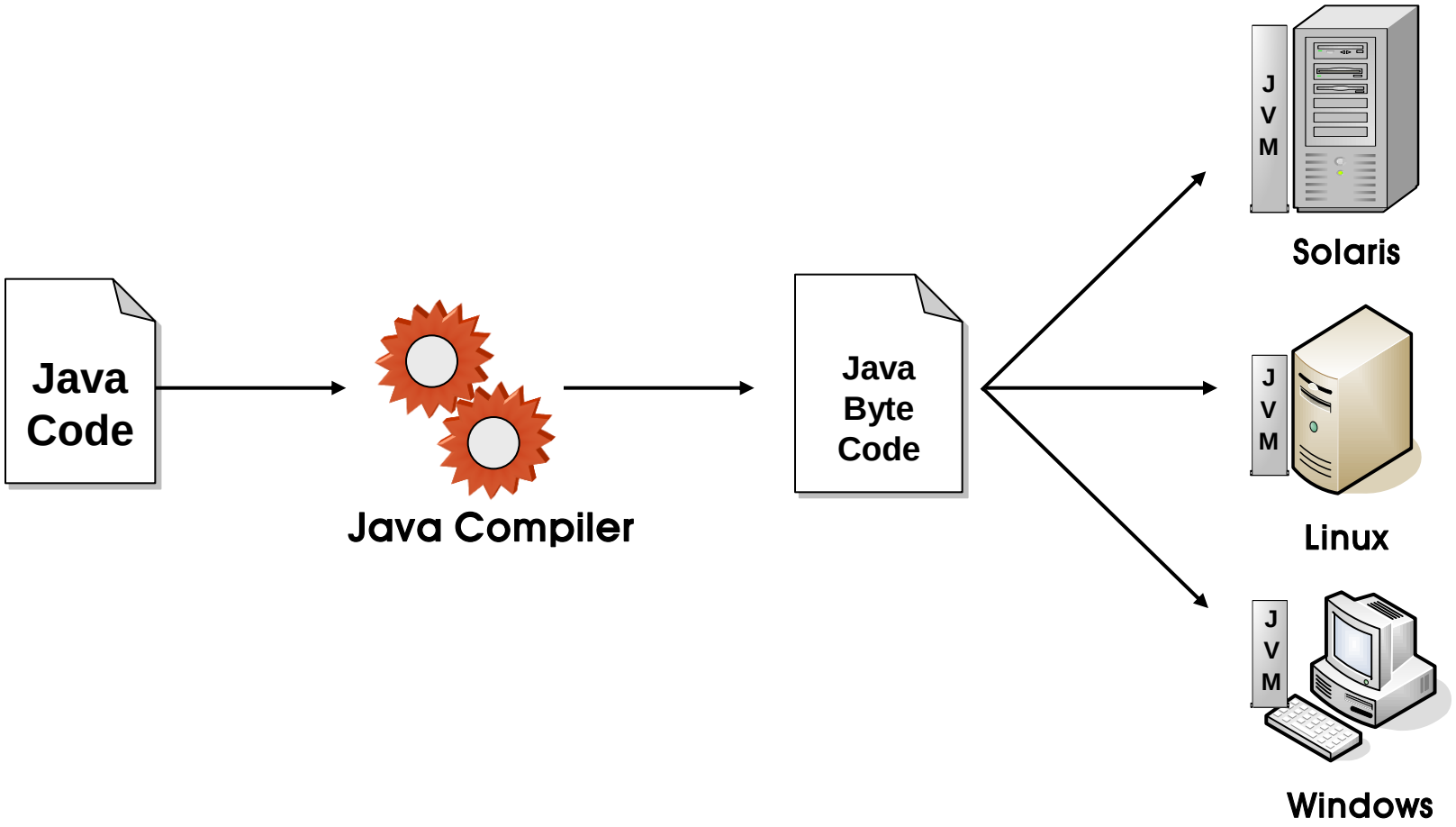
Code Security

- H/W Platform 규약을 제공한다.
- Platform 독립적인 bytecode를 해석하고, 실행한다.
- S/W 또는 H/W로 구현 된다.
- 실행환경은 일반 PC이거나, Web Brower일 수 있다
- 설치 된 Platform에 종속적이다.

이식성 문제


C
Code
Solaris C Compiler
Linux C Compiler
Windows C Compiler
Solaris
Binary
00100001
11001000
Linux
Binary
11100101
00011000
Windows
Binary
00100111
01001011
Solaris
Linux
Windows

이식성 문제


Java
Code
Java Compiler
Java
Byte
Code
J
V
M
Solaris
J
V
M
Linux
J
V
M
Windows

Garbage Collection

❑ 더 이상 사용되지 않는 메모리는 재사용 가능하게 해제시켜야 한다.

❑ 다른 언어에서는 메모리 헤제는 프로그래머가 직접 코드로서 구현해야 한다.

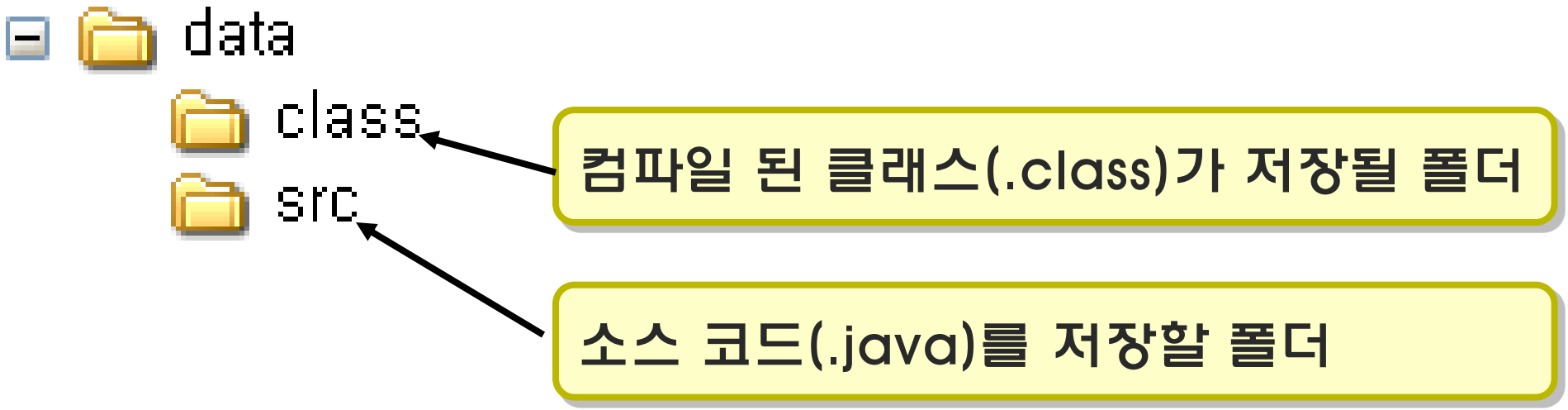
❑ 자바에서는 메모리를 지속적으로 감시하면서 더 이상 사용되지 않는 메모리를 해제 시켜준다.

❑ Garbage Collector

- 더 이상 사용 되지 않는 메모리를 검사한다.
- 자동적으로 일어난다. (개발자는 신경 쓸 필요 없음)
- 일어나는 시점은 구현 된 JVM마다 다를 수 있다. (Vendor Dependent)
- 개발자가 System.gc(); 로 호출할 수 있다.

실습 폴더 구성

❑ 다음의 폴더를 c:\ 에 작성합니다.



Sample Program 작성

TestGreeting.java

```
public class TestGreeting {
    public static void main( String[] args ) {
        Greeting hello = new Greeting();
        hello.greet();
    }
}

class Greeting {
    public void greet() {
        System.out.println( "hi" );
    }
}
```

Sample Program 작성

Greeting.java

```
public class Greeting {
    public void greet() {
        System.out.println( "hi" );
    }
}
```

TestGreeting.java

```
public class TestGreeting {
    public static void main( String[] args ) {
        Greeting hello = new Greeting();
        hello.greet();
    }
}
```

Consol 에서의 컴파일과 실행

❑ Compile

```
javac -d c:₩data₩class Greeting.java
```

```
javac -d c:₩data₩class -classpath c:₩data₩class TestGreeting.java
```

① 명령어

② 컴파일된 클래스파일이 저장될 위치 지정

③ 컴파일을 위해 필요한 클래스 파일들을 찾아올 위치 지정

④ 확장자를 포함한 소스파일명

❑ Run

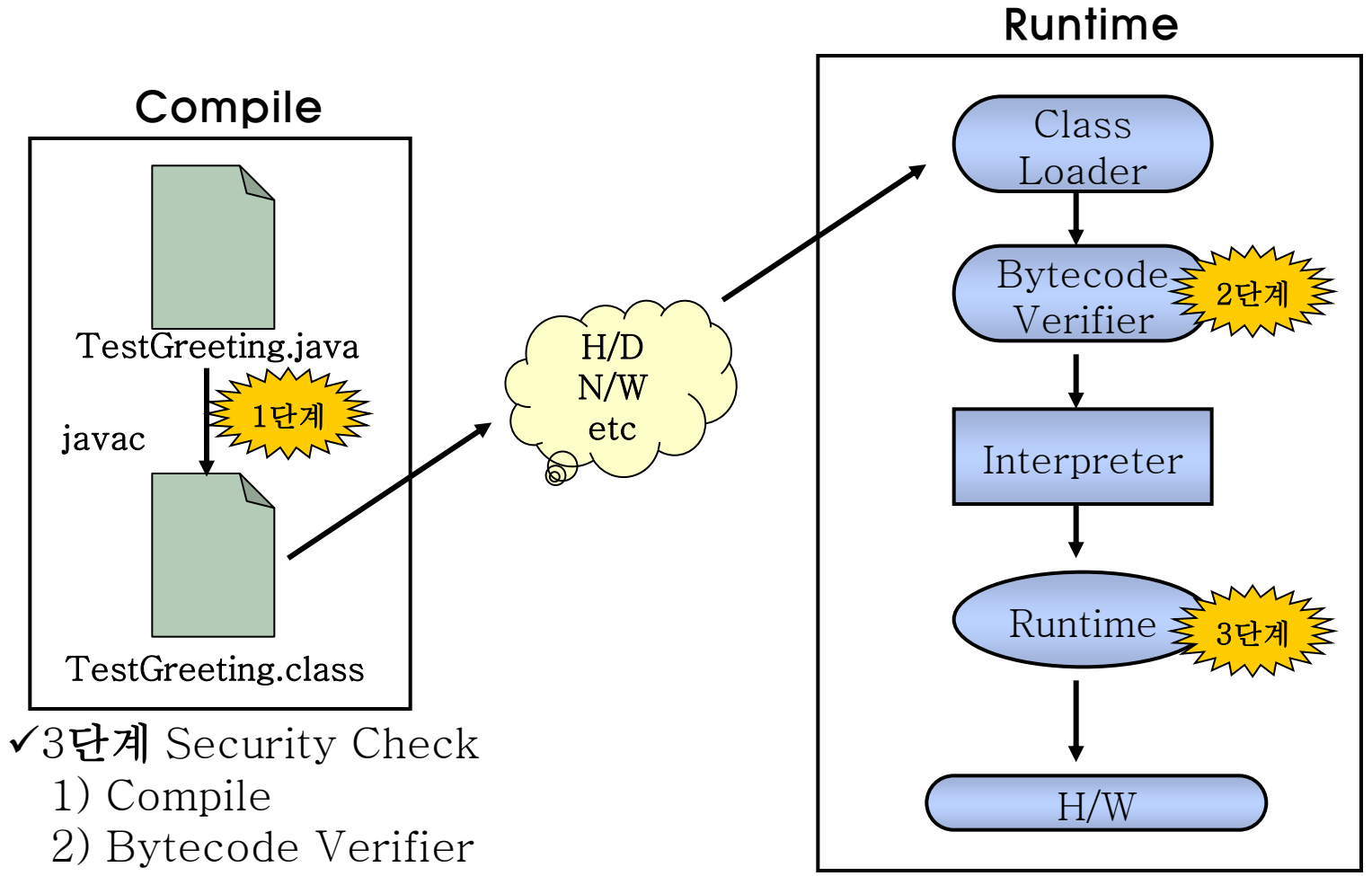
```
java -classpath c:₩data₩class TestGreeting
```

① 명령어

② 어느 위치에서부터 클래스를 찾을 것인가

③ 확장자를 제외한 클래스명

자바 소스 작성에서부터 실행 단계까지의 과정



✓3단계 Security Check
1) Compile
2) Bytecode Verifier
3) Runtime

Java 프로그램 수행 과정