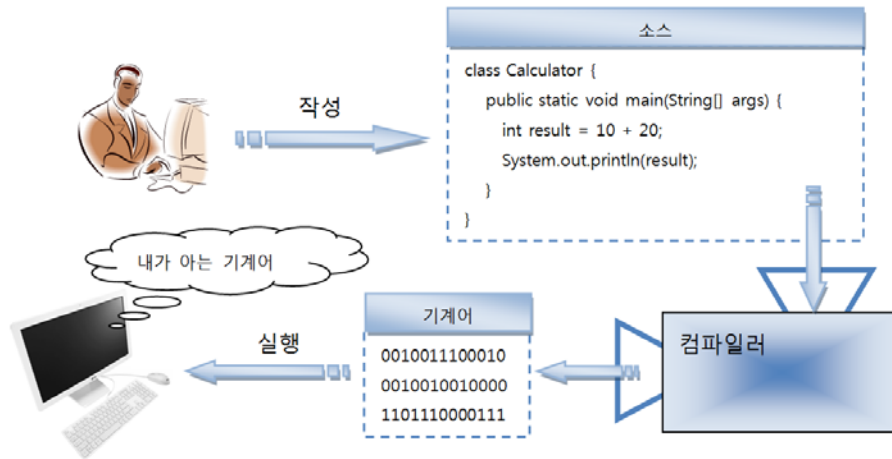


## 01장 자바 시작하기

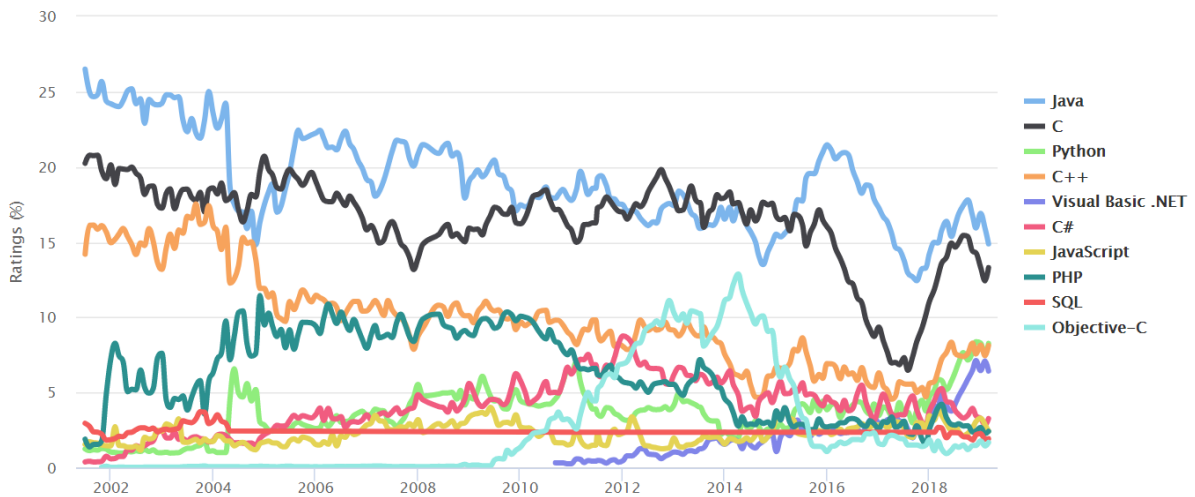
### 1.1 프로그래밍 언어란?

- 사람과 컴퓨터의 대화 도움
- 사람의 언어와 기계어 사이에서 다리와 같은 역할
- 고급 언어와 저급 언어로 구분



#### [꿀팁] 프로그래밍 언어의 인기도

- Source: <https://www.tiobe.com/tiobe-index/>



### 1.2 자바란?

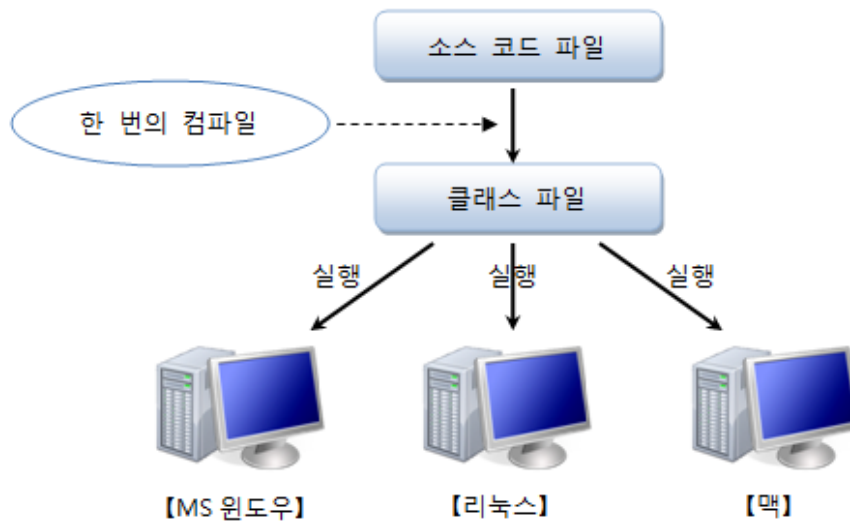
#### 1.2.1 자바 소개

- 1995년 썬마이크로시스템즈(Sun Microsystems)에서 최초 발표한 언어

- 1991년 가전 제품에서 사용할 목적인 오크(oak) 언어에서부터 시작
- 인터넷 프로그래밍 언어로 발전하면서 자바라는 이름으로 변경
- 2010년 오라클에서 썬을 인수하여 Java 개발, 관리, 배포 주관
- 버전 이력
  - JDK 1.0 (1996): C++의 단점(메모리 관리, 다중상속)을 극복하고자 만든 언어.
  - JDK 1.1 (1997): RMI, JDBC, Inner Class 개념이 포함
  - JDK 1.2 (1998): Java SE 1.2, ME 1.2, EE 1.2 발표
  - JDK 1.3 (2000): HotSpot, JNDI 포함
  - JDK 1.4 (2002): 오픈소스 정책으로 자바가 관리됨
  - JDK 1.5 (2004): Generics, annotation, auto boxing, enum, 향상된 for문 등
  - JDK 1.6 (2006): 보안, 성능강화 주력
  - JDK 1.7 (2011): NIO, JavaFX
  - JDK 1.8 (2014): Lambda expression, default method interface 등

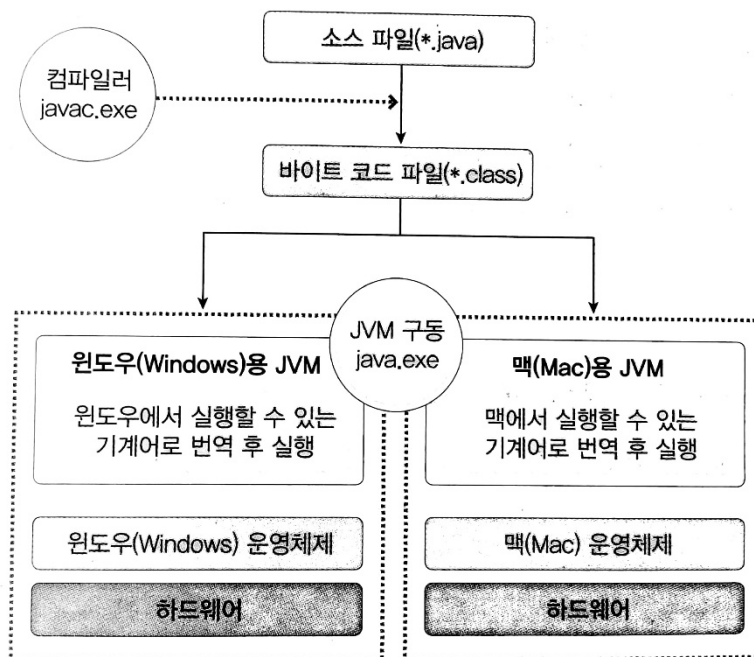
### 1.2.2 자바의 특징

- 이식성이 높은 언어이다.
- 객체 지향 언어이다.
- 함수적 스타일 코딩을 지원한다.
  - 대용량 데이터의 병렬 처리, 이벤트 지향 프로그램을 위해 적합
- 메모리를 자동으로 관리한다.
- 다양한 애플리케이션을 개발할 수 있다.
- 멀티 스레드(Multi-Thread)를 쉽게 구현할 수 있다.
- 동적로딩을 지원한다.
  - 애플리케이션이 실행될 때 모든 객체가 생성되지 않고, 객체가 필요한 시점에 클래스를 동적 로딩해서 객체를 생성한다. --> 유지보수가 용이함
- 막강한 오픈소스 라이브러리가 풍부하다.



### 1.2.3 자바 가상 기계(JVM)

- 자바 프로그램은 완전한 기계어가 아닌 중간 단계의 바이트 코드이기 때문에 이것을 해석하고 실행할 수 있는 가상의 운영체제가 필요하다. -> 자바 가상 기계(JVM, Java Virtual Machine)



### 1.3 자바 개발 환경 구축

### 1.3.1 OS 계정 생성 (선택사항)

- Windows 7 Enterprise K 버전에서 "제어판 > 관리 도구 > 컴퓨터 관리 > 시스템 도구 > 로컬 사용자 및 그룹 > 사용자" 선택하여 아래와 같이 OS 계정을 생성한다.
  - 사용자 이름: java
  - 암호: hi
  - 다음 로그인 시 사용자가 반드시 암호를 변경해야 함: uncheck
  - 사용자가 암호를 변경할 수 없음: check
  - 암호 사용 기간 제한 없음: check
  - 소속 그룹: Administrators 추가
- 새로 만든 java 계정으로 로그인하여 해당 계정과 관련된 폴더와 파일(예. 내 문서, 다운로드, 바탕화면등)을 생성시키도록 하고 로그오프한다.
- 다시 Administrator 계정으로 로그인하여 c:\Users\java 폴더를 "잘라내기"하여 d:\Users\ 밑으로 붙여 놓는다.
- 명령 프롬프트를 실행하여 아래와 같이 설정한다.

```
C:\Users\Administrator>cd ..
C:\Users>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 766A-BA2B

C:\Users 디렉터리

2021-02-18 오후 05:14 <DIR> .
2021-02-18 오후 05:14 <DIR> ..
2020-07-09 오전 10:15 <DIR> Administrator
2021-01-27 오후 01:02 <DIR> Administrator
2021-01-27 오전 09:43 <DIR> Public
                0개 파일                0 바이트
                5개 디렉터리 12,822,536,192 바이트 남음

C:\Users>mklink /d java d:\Users\java
java <<==>> d:\Users\java에 대한 기호화된 링크를 만들었습니다.

C:\Users>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 766A-BA2B

C:\Users 디렉터리

2021-02-18 오후 05:23 <DIR> .
2021-02-18 오후 05:23 <DIR> ..
2020-07-09 오전 10:15 <DIR> Administrator
2021-01-27 오후 01:02 <DIR> Administrator
2021-02-18 오후 05:23 <SYMLINKD> java [d:\Users\java]
2021-01-27 오전 09:43 <DIR> Public
                0개 파일                0 바이트
                6개 디렉터리 12,578,193,408 바이트 남음
```

### 1.3.2 자바 개발 도구(JDK) 설치

- JRE = JVM + 표준 클래스 라이브러리
- JDK = JRE + 개발에 필요한 도구(컴파일러)

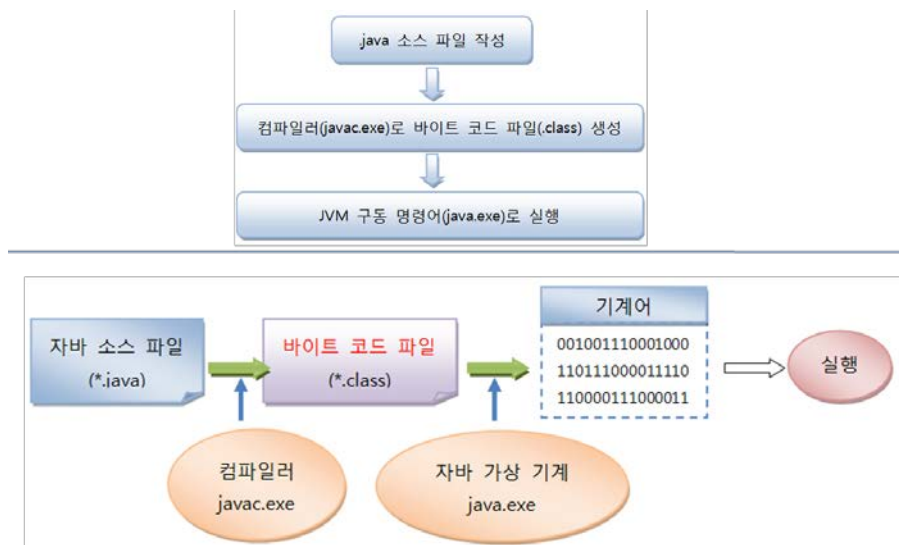
#### [꿀팁] 패스(path)설정

- 내컴퓨터 우클릭/속성/고급시스템설정/고급탭/환경 변수 /시스템 변수/새로만들기/  
변수이름 JAVA\_HOME  
변수값 C:\Program Files\Java\jdk버전
- path 편집/변수값 뒤에 (세미콜론);%JAVA\_HOME%\bin

### 1.3.3 API 도큐먼트

- JDK에서 제공하는 표준 클래스 라이브러리를 API(Application Programming Interface)라고도 한다. API들은 매우 방대하기 때문에 쉽게 찾을 수 있도록 API 도큐먼트를 제공한다.
- <http://docs.oracle.com/javase/8/docs/api/index.html>

## 1.4 자바 프로그램 개발 순서



#### [실습]

1. 메모장을 이용하여 아래와 같이 작성하여 c:\ 디렉토리에 "Hello.java"라고 저장한다.

[Hello.java] 소스 파일 생성과 컴파일

```
public class Hello {
    public static void main(String[] args) {
        System.out.println("Hello, welcome to the java world!");
    }
}
```

```
}
```

2. 명령 프롬프트를 실행하여 아래와 같이 수행한다.

**[Hello.java] 소스 파일 생성과 컴파일**

```
c:\>javac Hello.java
-- 한글처리
c:\>javac Hello.java -encoding utf-8
c:\>java Hello
Hello, welcome to the java world!
c:\>
```

## [과제] 자기소개 프로그램

- 다음과 같은 결과가 출력 되도록 프로그램을 작성하세요? (단, 1. 메모장으로 작성한 다음 console 에서 직접 컴파일하여 \*.class 파일을 제출할 것, 2. 컴파일된 class 파일명을 자신의 이름으로 한다. 예) Hongkildong.class )
  1. 이름 :
  2. SW경력:
  3. 희망사항:

## 1.5 주석과 실행문

### 1.5.1 주석과 실행문

- 프로그램 실행과는 상관없이 코드에 설명 붙인 것
- 컴파일 과정에서 주석은 무시되고 실행문만 바이트 코드로 번역
- 코드에서 사용하는 주석문의 종류

주석 기호	설명
//	//부터 라인 끝까지 주석으로 처리한다. (행 주석)
/* ~ */	/*와 */ 사이에 있는 모든 범위를 주석으로 처리한다. (범위 주석)

### 1.5.2 실행문과 세미콜론(;)

- 변수 선언, 값 저장, 메소드 호출에 해당하는 코드
- 실행문 끝에는 반드시 세미콜론(;)을 붙여 실행문의 끝 표시

```
int x = 1;           //변수 x를 선언하고 1을 저장
int y = 2;           //변수 y를 선언하고 2를 저장
int result = x + y;   //변수 result를 선언하고 변수 x와 y를 더한 값을 저장
System.out.println(result); //콘솔에 출력하는 메소드 호출
```

## 1.6 이클립스

### 1.6.1 이클립스 소개

- 2003년 IBM에서 개발
- 자바 통합 개발 환경(IDE: Integrated Development Environments) 제공
  - 프로젝트 생성 기능 제공
  - 자동 코드 완성 기능 제공
  - 디버깅 기능 제공
- 이클립스 연합(Eclipse Foundation) 설립 - 지속적 버전업과 배포
- 다양한 개발 환경을 구축할 수 있도록 플러그인(Plug-In) 설치 가능
  - 안드로이드 개발 환경
  - 스프링(Spring) 개발 환경
  - C, C++ 개발 환경

### 1.6.2 이클립스 다운로드

- 이클립스는 자바 언어로 개발된 툴 - JDK 필요
- 다운로드 사이트: <https://www.eclipse.org/downloads/>
  - Eclipse IDE for Java Developers 버전
    - 순수 자바 학습용
  - Eclipse IDE for Java EE Developers 버전 (수업에서 사용하는 버전)
    - 웹 애플리케이션 등의 Enterprise (Network) 환경에서 실행
  - CPU 사양에 맞게 다운로드

#### ■ Preferences 설정

- 에디터의 행 번호 표기하기: Eclipse > Windows > Preferences > General/Editors/Text Editors > "Show line numbers"를 체크
- 에디터의 인코딩 설정: Eclipse > Windows > Preferences > General/Workspace > Text file encoding > Default (MS949) 로 선택

### 1.6.3 워크스페이스

- 이클립스에서 생성한 프로젝트가 기본적으로 저장되는 디렉토리
- 최초 실행 시 워크스페이스 런처(Workspace Launcher)에서 설정
- .metadata 디렉토리
  - 자동 생성되며 이클립스 실행 시 필요한 메타데이터 저장
  - 이 디렉토리 삭제하고 이클립스 실행 - 초기 상태로 다시 실행

#### 1.6.4 퍼스펙티브와 뷰

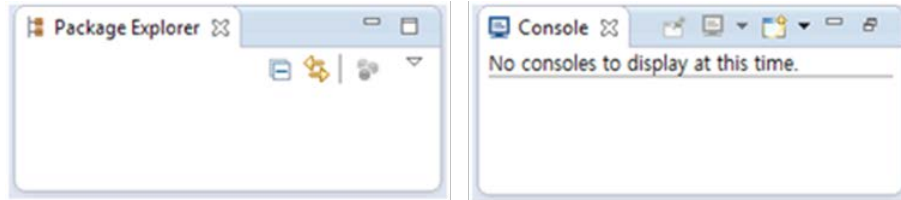
■ 퍼스펙티브(Perspective)

- 개발 프로젝트 종류별로 유용한 View들을 묶어놓은 것
- Eclipse IDE for Java EE Developers: 기본적으로 Java EE 퍼스펙티브



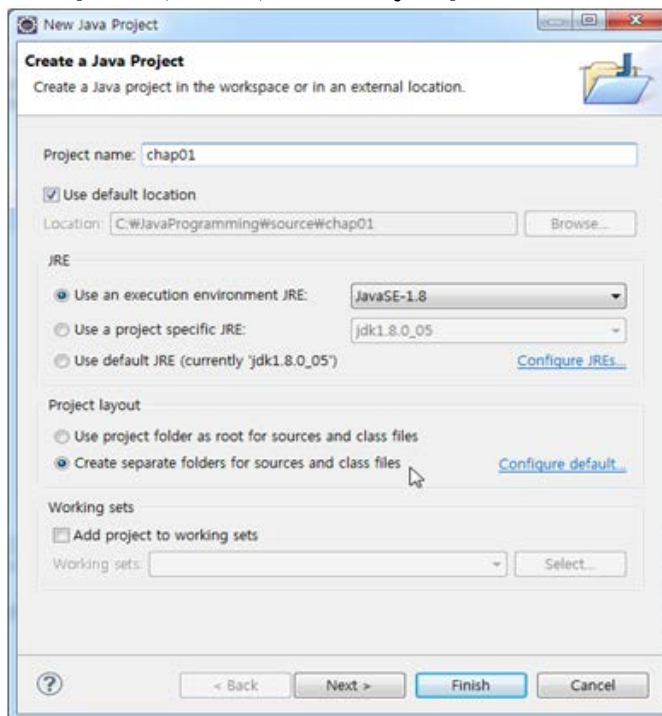
■ 뷰(View)

- 퍼스펙티브를 구성하는 작은 창으로 여러가지 목적에 맞게 내용 보여줌
- 자유롭게 제거 하거나 추가 가능
- 수업에서 유용한 뷰들: Package Explorer, Console



#### 1.6.5 프로젝트 생성

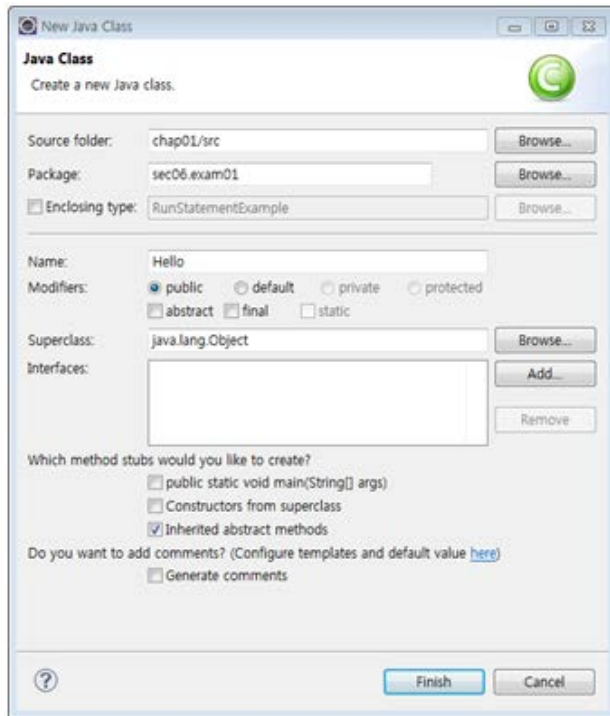
- 메뉴에서 [File -> New -> Java Project]를 차례대로 클릭한다.



#### 1.6.6 소스 파일 생성과 컴파일

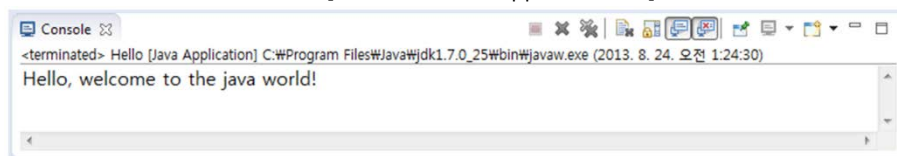


- chap01 프로젝트의 src 폴더를 선택하고 [마우스 오른쪽 버튼 -> New -> Class]를 선택한다.



### 1.6.7 바이트 코드 실행

- 실행 방법 - 툴바에서 클릭
- Package Explorer뷰에서 소스 파일 선택
  - 마우스 오른쪽 버튼 눌러 [Run As Java Application] 클릭



### 1.6.8 Eclipse 단축키

- 실행
  - **Ctrl + F11 : 바로 전에 실행했던 클래스 실행**
- 소스 네비게이션
  - Ctrl + O : 해당 소스의 메소드 리스트를 확인하려 할때
  - Ctrl + 마우스커서(혹은 F3) : 클래스나 메소드 혹은 멤버를 상세하게 검색하고자 할때
  - Alt + Left, Alt + Right : 이후, 이전
  - F4 : 클래스명을 선택하고 누르면 해당해당 클래스의 Hierarchy 를 볼 수 있다.
- 소스 편집
  - Ctrl + Space : 입력 보조장치(Content Assistance) 강제 호출 => 입력하는 도중엔 언제

라도 강제 호출 가능하다.

- Ctrl + Shift + O : 자동으로 클래스 import 시킨다.
- F2 : 컴파일 에러의 빨간줄에 커서를 갖저다가 이이 키를 누르면 에러의 원인에 대한 힌트를 제공한다.
- Ctrl + L : 원하는 소스 라인으로 이동, 로컬 히스토리 기능을 이용하면 이전에 편집했던 내용으로 변환이 가능하다.
- Ctrl + Shift + Space : 메소드의 메소드의 가로안에 커서를 놓고 이 키를 누르면 파라미터 타입 힌트를 볼 수 있다.
- Ctrl + D : 한줄 삭제
- Ctrl + W : 파일파일 닫기
- Ctrl + II : 들여쓰기 자동 수정
- Ctrl + / : 블록 주석()
- Ctrl + // : 여러줄이 한꺼번에 주석처리됨. 주석 해제하려면 반대로 하면 된다.
- Alt + Up(Down) : 위(아래)줄과줄과 바꾸기
- Alt + ShiftShift + 방향키 : 블록 선택하기
- Ctrl + Shift + Space : 메소드의 파라미터파라미터 목록 보기
- Ctrl + Shift + O : 자동으로 import 하기
- Ctrl + Shift + F4 : 열린 파일 모두 닫기
- Ctrl + M : 전체화면 토글
- Ctrl + Alt + Up(Down) : 한줄(블록) 복사
- Ctrl + , or . : 다음 annotation(에러, 워닝, 북마크 가능가능 )으로 점프
- Ctrl + 1 : 쿼 픽스
- F3 : 선언된 변수로 이동, 메소드 정의부로 이동이동
- Ctrl + T : 하이어라키 팝업 창 띄우기(인터페이스 구현 클래스간 이동시 편리편리 )
- Ctrl + O : 메소드나 필드 이동하기
- Ctrl + F6 : 창간 전환, UltraEdit 나 Editplus 의 Ctrl + Tab 과 같은 기능

## [과제] 확인문제

1. JDK와 JRE의 차이점을 설명한 것 중 틀린 것은 무엇입니까?

- (1) JDK는 JRE와 컴파일러 등의 개발 도구가 포함된다.
- (2) 자바 프로그램을 개발하려면 JDK가 반드시 필요하다.
- (3) 자바 프로그램을 실행만 하려면 JRE를 설치해도 상관없다.
- (4) JRE에는 컴파일러(javac.exe)가 포함되어 있다.

2. JVM에 대한 설명으로 틀린 것은 무엇입니까?

- (1) JVM은 java.exe 명령어에 의해 구동된다.
- (2) JVM은 바이트 코드를 기계어로 변환시키고 실행한다.
- (3) JVM은 운영체제에 독립적이다(운영체제별로 동일한 JVM이 사용된다).
- (4) 바이트 코드는 JVM에 독립적이지만, JVM은 운영체제에 종속적이다.

3. 자바 프로그램 개발 과정을 순서대로 나열하세요.

- (1) javac.exe로 바이트 코드 파일(~.class)을 생성한다.
- (2) java.exe로 JVM을 구동시킨다.

- (3) 자바 소스 파일(~.java)을 작성한다.
- (4) JVM은 main() 메소드를 찾아 메소드 블록을 실행시킨다.

4. 자바 소스 파일을 작성할 때 틀린 것은 무엇입니까?

- (1) 자바 소스 파일명과 클래스 이름은 대소문자가 동일해야 한다.
- (2) 클래스 블록과 메소드 블록은 반드시 중괄호{}로 감싸야 한다.
- (3) 실행문 뒤에는 반드시 세미콜론(;)을 붙여야 한다.
- (4) 주석은 문자열 안에도 작성할 수 있다.

5. 자바 주석문의 종류가 아닌 것은 무엇입니까?

- (1) //: 행 주석
- (2) /\* ... \*/: 범위 주석
- (3) /\*\* ... \*/: API 도큐먼트 주석
- (4) <!-- ... -->: 범위 주석

6. 이클립스에 대한 설명으로 틀린 것은 무엇입니까?

- (1) 이클립스는 JDK(JRE)를 설치해야만 실행할 수 있다.
- (2) 이클립스에서 소스 파일을 저장하면 바로 컴파일되어 바이트 코드 파일이 생성된다.
- (3) 워크스페이스는 프로젝트들이 생성되는 기본 디렉토리를 말한다.
- (4) .metadata는 프로젝트의 소스 파일들이 저장되는 디렉토리이다.