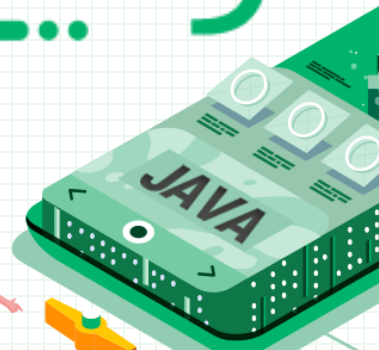




# 안드로이드 프로그래밍을 위한 자바기초 \_...



## ① JAVA 언어 특성 이해하기



## 학습목표

- JAVA 언어의 개요 및 특성을 이해할 수 있다.
- JAVA 언어를 이용하여 컴퓨터 프로그래밍을 하기 위한 개발환경을 구축할 수 있다.
- JAVA 언어를 이용하여 컴퓨터 프로그래밍을 작성하고 디버깅 및 실행할 수 있다.



## 학습내용

- JAVA 언어의 개요 파악하기
- JAVA 언어의 개발환경 구축하기
- JAVA 언어 프로그램 작성 및 디버깅하기

## ① JAVA 언어 특성 이해하기

### JAVA 언어의 개요 파악하기

#### ▶ 자바(JAVA) 언어의 개요

##### ▶ 개요

- (1) 1995년 썬마이크로시스템(Sun Microsystem) 사에 의해 개발됨
  - ✓ 현재 오라클(Oracle)사가 저작권을 가지고 있음
- (2) 객체지향 프로그래밍 언어(OOP : Object Oriented Programming)
- (3) 크로스 플랫폼(Cross Platform) 언어
  - ✓ 운영체제에 종속되지 않은 언어
  - ✓ 한번 컴파일된 코드는 재컴파일 없이 다른 운영체제에서 실행 가능
- (4) C/C++언어와 매우 흡사하여 프로그래머들에게 친숙함

#### ▶ 자바(JAVA) 언어의 기술

##### 1) 자바 개발 키트(JDK : Java Development Kit)

- (1) 자바 애플리케이션 개발에 필요한 자바 라이브러리
- (2) 자바 애플리케이션 개발에 필요한 명령들 (컴파일러 등)
- (3) 자바 실행환경( JRE : Java Runtime environment )
  - ✓ 자바 애플리케이션 실행에 필요한 모듈들
  - ✓ 자바 가상 머신(JVM : Java Virtual Machine) : 자바 애플리케이션 실행

##### 2) 자바 통합개발환경(IDE)

EclipseIntelliJ IDEANetBeansJdeveloper

## ① JAVA 언어 특성 이해하기

## JAVA 언어의 개요 파악하기

## ▶ 자바 프로그래밍 작업 순서

## 1) 코딩(Coding) 작업

## (1) 컴퓨터언어로 소스파일(Source File)을 만드는 작업

- ✓ 소스파일을 소스코드(Source Code) 또는 원시프로그램(Source Program)이라고도 함
- ✓ 문자로 구성된 텍스트 파일 형태임

## (2) 컴퓨터 언어에 따라 파일의 확장자가 다름

- ✓ Java 언어 : \*.java
- ✓ C언어 : \*.c

## 2) 컴파일(Compile) 작업

## (1) 소스파일을 바이트코드로 변환하는 작업

- ✓ 문법적 오류를 점검하여 오류가 없는 경우에 실행 가능한 파일로 변환
- ✓ 중간 코드 형태인 바이트코드 파일이 생성됨

## (2) 어떠한 운영체제에서 컴파일 작업을 하더라도 동일한 바이트 코드가 생성됨

- ✓ 확장자 : \*.class
- ✓ 다른 운영체제에서도 사용이 가능 (크로스 플랫폼 언어)

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

Java 언어 소스파일 (\*.java)

컴파일  
(Compile)

```
101011111011
010111110110
010111110110
~
010111110110
101111101100
011111011010
```

바이트코드 (\*.class)

## ① JAVA 언어 특성 이해하기

## JAVA 언어의 개요 파악하기

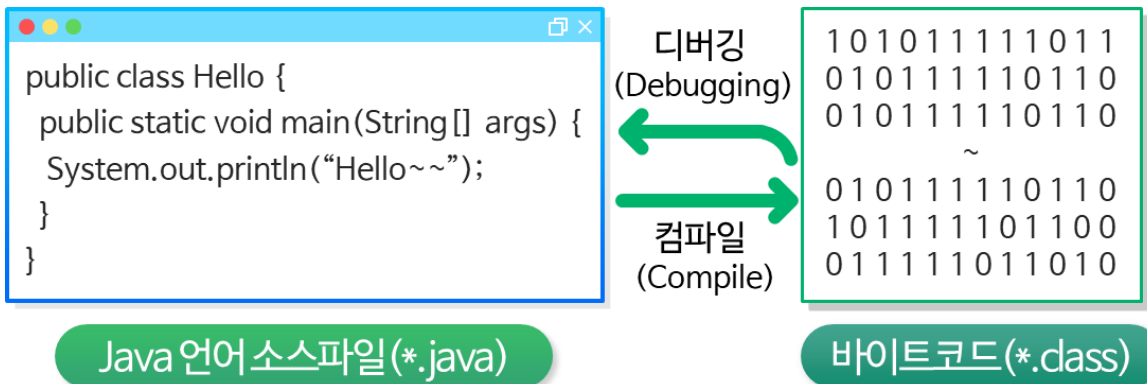
## 3) 디버깅(Debugging) 작업

## (1) 문법적 오류를 수정하는 작업

- ✓ 소스파일에 문법적 오류가 발생한 부분을 수정하는 작업

## (2) 소스파일을 수정하였으면 컴파일 작업을 다시 해야함

- ✓ 디버깅과 컴파일을 반복하여 문법적 오류 해결



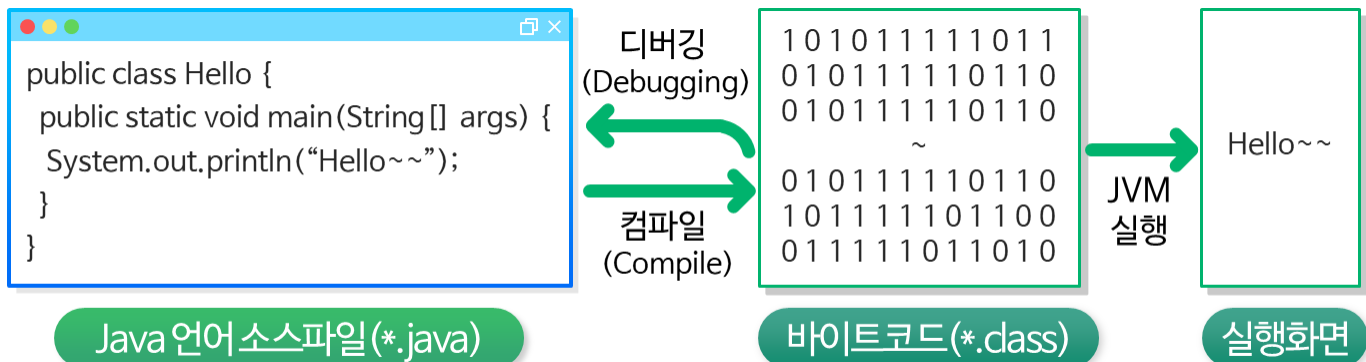
## 3) 실행 작업

## (1) 문법적 오류가 없는 바이트코드 파일을 실행

- ✓ 자바 가상 머신(JVM)이 바이트코드를 실행

## (2) 실행 시 오류가 발생되면

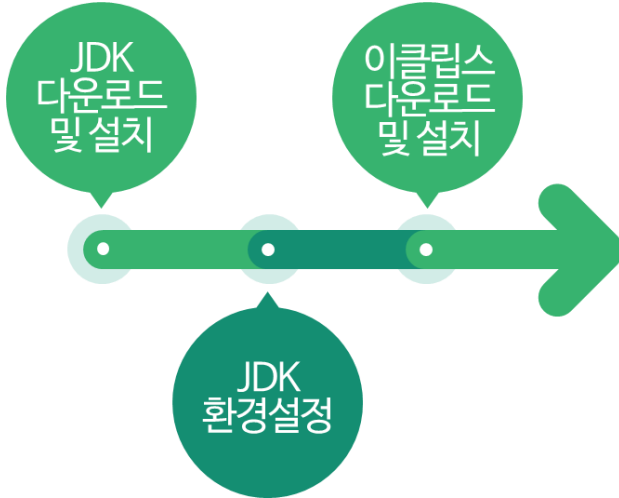
- ✓ 소스파일의 알고리즘을 점검하여 수정
- ✓ 컴파일과 디버깅 및 실행을 반복



## ① JAVA 언어 특성 이해하기

## JAVA 언어의 개발환경 구축하기

### 자바 개발환경 구축



### JDK 다운로드 및 설치

#### ▶ JDK SE(Standard Edition)

- 오라클 홈페이지 : <https://www.oracle.com/java>
- 64비트 윈도우 운영체제용 설치 파일을 다운로드 및 실행

### JDK 환경설정

#### 1) PATH 환경 설정

##### (1) JDK 설치 폴더

- ✓ C:\Program Files\Java\jdk-15\

##### (2) PATH 환경변수에 bin 폴더 추가

- ✓ C:\Program Files\Java\jdk-15\bin

## ① JAVA 언어 특성 이해하기

### JAVA 언어의 개발환경 구축하기

#### 2) PATH 환경 설정 확인

(1) "명령 프롬프트" 실행

(2) 자바 실행환경 및 컴파일러 동작 확인

- ✓ 자바 실행환경 확인 명령 : `java -version`
- ✓ 자바 컴파일러 확인 명령 : `javac`

#### ▶ 이클립스 설치

##### ▶ 이클립스

- 자바 언어 통합개발환경
- 다운로드 및 설치 : <https://www.eclipse.org>

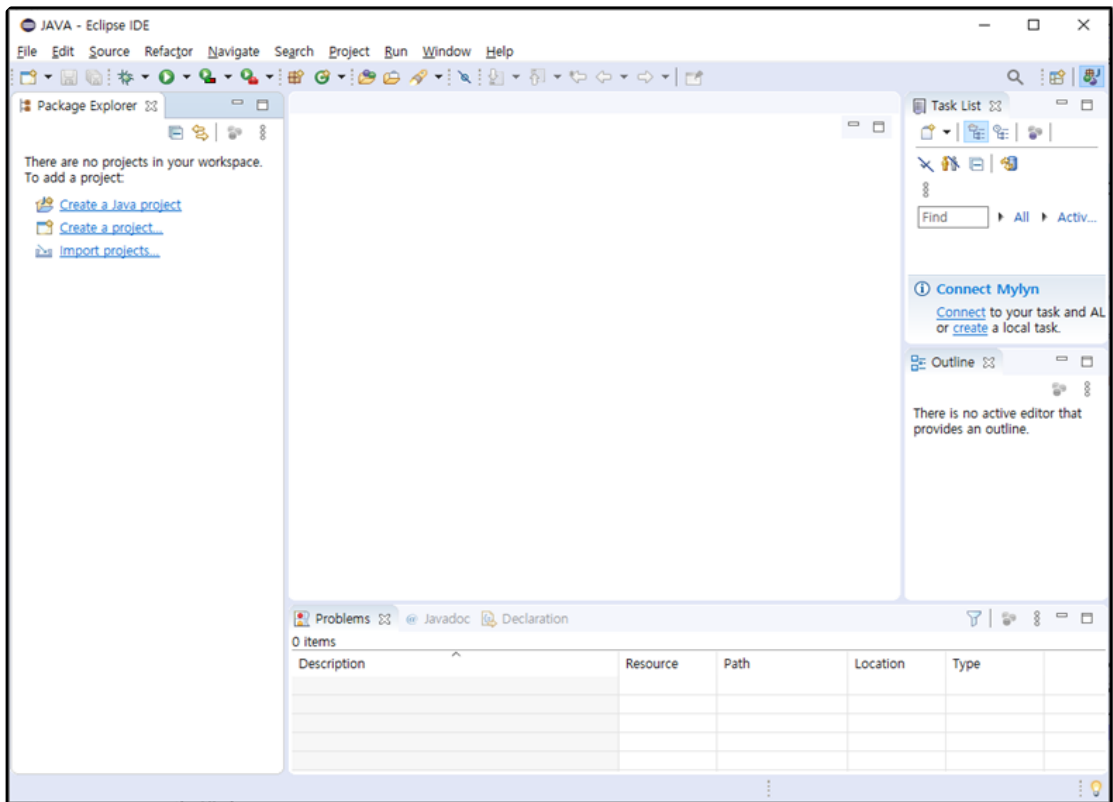
## ① JAVA 언어 특성 이해하기

# 실습

## 자바 환경 설정 실습



### 실행 화면



- 자세한 내용은 실습 영상을 확인해보세요.



## ① JAVA 언어 특성 이해하기

## JAVA 언어 프로그램 작성 및 디버깅하기

## ▶ JAVA 언어 프로그램 기본 구조

```

package test;
import java.util.*;

class Hello
{
    int seek;
    .....

    void sort( )
    {
        .....
    }

    public static void main(String args[])
    {
        .....
        Hello h = new Hello();
    }
}

```

패키지(package) 지정 및 선택

클래스 안에 선언하는 멤버 변수(필드) 선언

클래스 안에 선언하는  
멤버 메소드(method) 선언

- 프로그램의 시작부분이 main 메소드
- 멤버 메소드는 아님

클래스 선언

## ▶ 주석문 이해하기

## 1) 주석문이란?

- 주석문
  - ✓ 프로그램 소스에 대한 설명을 표기하는 문장
  - ✓ 프로그램 실행결과에 영향을 주지 않음

## 2) 사용 목적

- 프로그램 소스에 설명을 표기하여 문서화 작업
- 프로그램 디버깅 시 테스트 목적으로 사용

## ① JAVA 언어 특성 이해하기

## JAVA 언어 프로그램 작성 및 디버깅하기

## 3) 주석문의 종류

한 줄 주석문

//

여러 줄 주석문

/\* ~~ \*/

```
// 합계를 계산하는 변수
```

한줄 주석문

```
/*
  이 프로그램은 자바프로그램
  작성자 : 000
  일자 : 년 월 일
*/
```

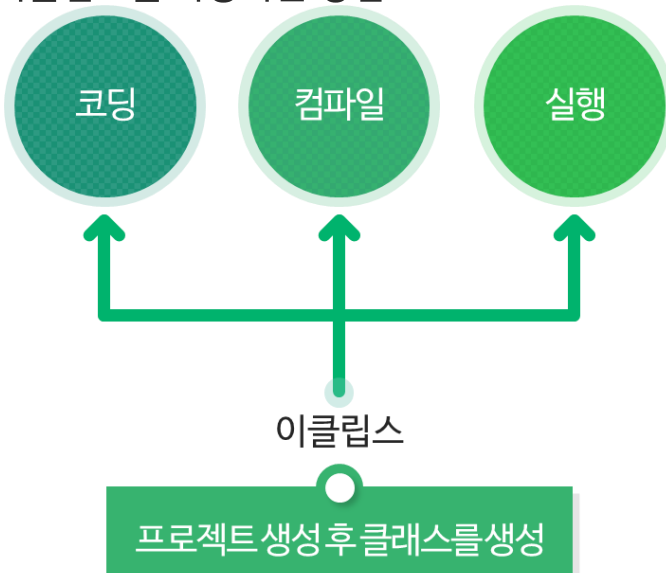
여러줄 주석문

## ➤ JAVA 언어 프로그램 작성방법

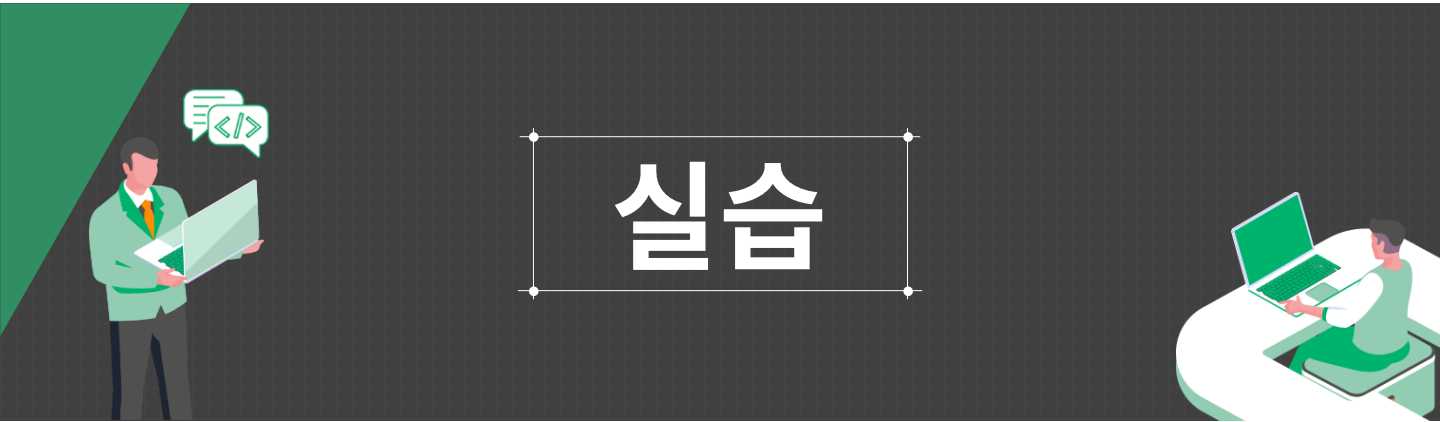
## 1) 이클립스를 사용하지 않고 개발하는 방법

- 코딩 - 메모장
- 컴파일 - javac
- 실행 - java

## 2) 이클립스를 사용하는 방법



## ① JAVA 언어 특성 이해하기



## JAVA 언어 프로그램 작성 및 실행



### 실행 화면

Hello~~

- 소스 파일명 : [hello.java]
- 자세한 내용은 실습 영상을 확인해보세요.



## 정리하기

### ■ JAVA 언어의 개요 파악하기

- 자바(JAVA) 언어의 개요
  - 객체지향 프로그래밍 언어(OOP : Object Oriented Programming)
  - 크로스 플랫폼(Cross Platform) 언어
- 자바 개발 키트(JDK : Java Development Kit)
  - 자바 애플리케이션 개발에 필요한 자바 라이브러리
  - 자바 애플리케이션 개발에 필요한 명령들(컴파일러 등)
  - 자바 실행환경( JRE : Java Runtime environment )
    - 자바 가상 머신(JVM : Java Virtual Machine)
- 자바 통합개발환경(IDE) : Eclipse
- 자바 프로그래밍 작업 순서
  - 코딩(Coding) 작업 : \*.java
    - 컴퓨터언어로 소스파일(Source File)을 만드는 작업
  - 컴파일(Compile) 작업
    - 소스파일을 바이트코드로 변환하는 작업
    - 어떠한 운영체제에서 컴파일을 하더라도 동일한 바이트 코드가 생성됨
    - 확장자 : \*.class
  - 디버깅(Debugging) 작업
    - 문법적 오류를 수정하는 작업
    - 소스파일을 수정하였으면 컴파일 작업을 다시 해야 함
  - 실행 작업
    - 문법적 오류가 없는 바이트코드 파일을 실행



## 정리하기

### ■ JAVA 언어의 개발환경 구축하기

- JDK 다운로드 및 설치
  - JDK SE(Standard Edition) : <https://www.oracle.com/java>
- JDK 환경 설정
  - PATH 환경변수에 bin 폴더 추가
  - PATH 환경 설정 확인
    - 자바 실행환경 확인 명령 : `java -version`
    - 자바 컴파일러 확인 명령 : `javac`
- 이클립스 다운로드 및 설치
  - 자바 언어 통합개발환경
  - 다운로드 및 설치 : <https://www.eclipse.org>



## 정리하기

### ■ JAVA 언어 프로그램 작성 및 디버깅하기

- 자바 프로그램 기본구조
  - 클래스 단위로 프로그래밍
  - 프로그램 시작 : main 메소드
  - 패키지 선언 및 선택 지정
- JAVA 언어 프로그램 작성 방법
  - 이클립스를 사용하지 않게 개발하는 방법
    - 코딩 : 메모장
    - 컴파일 : javac
    - 실행 : java
  - 이클립스를 사용하는 방법
    - 코딩, 컴파일, 실행 : 이클립스
    - 프로젝트 생성 후 클래스를 생성