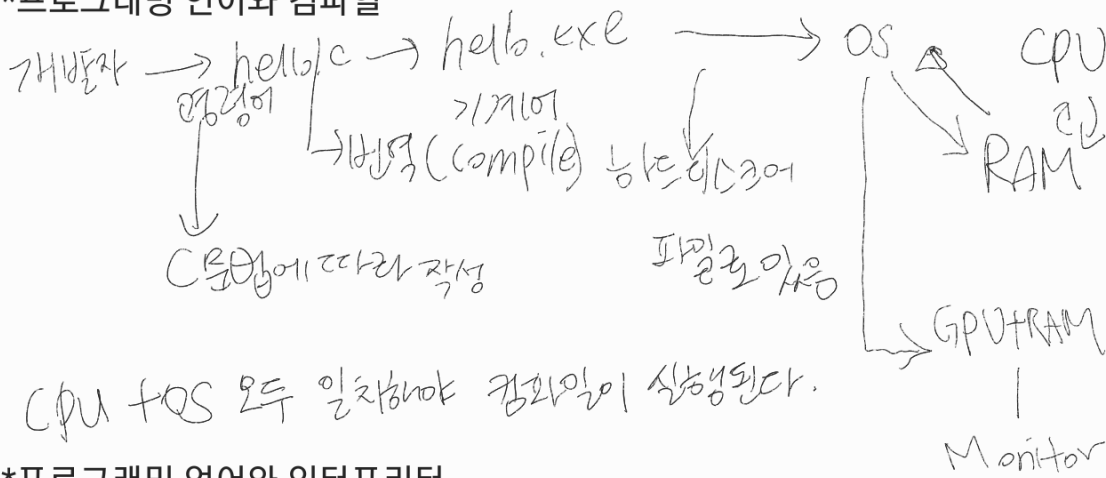
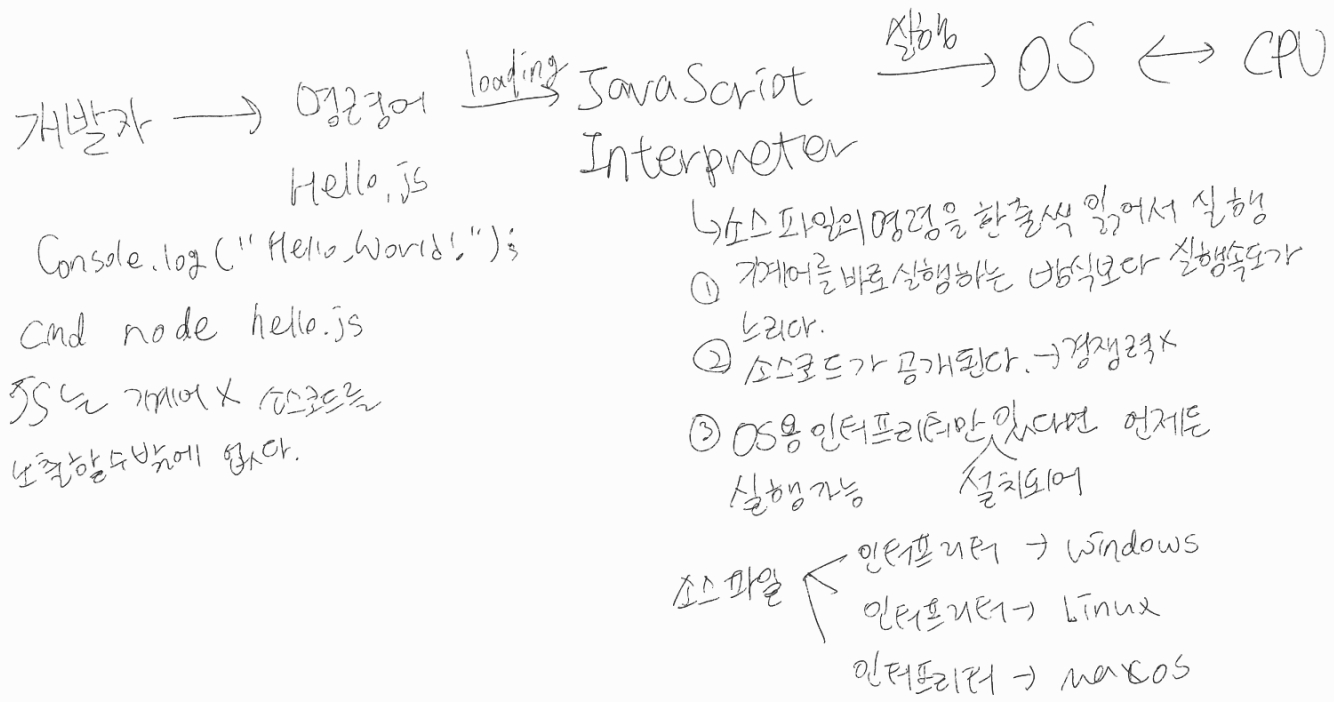


## \*프로그래밍 언어와 컴파일

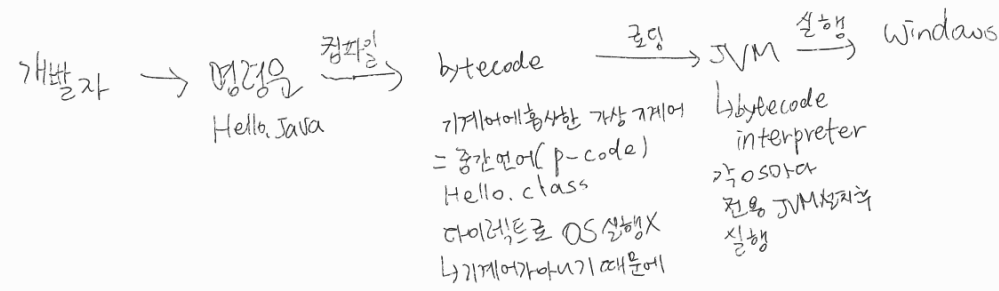


## \*프로그래밍 언어와 인터프리터



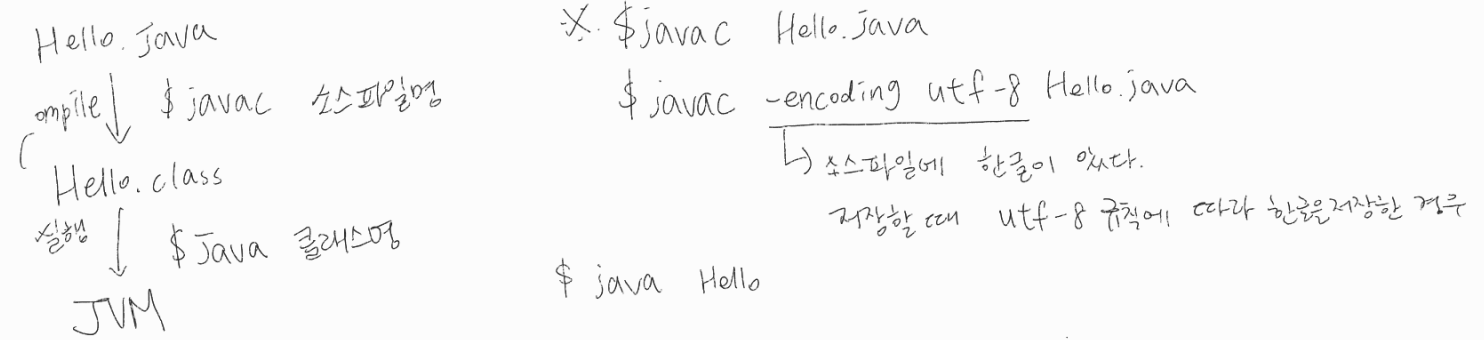
인터프리터 aka 엔진 (engine)  
aka player  
aka runtime  
aka VM (Virtual Machine)

# \* 자바 프로그래밍 ("Write Once, Run Anywhere") 하이브리드 (컴파일 + 인터프리터)

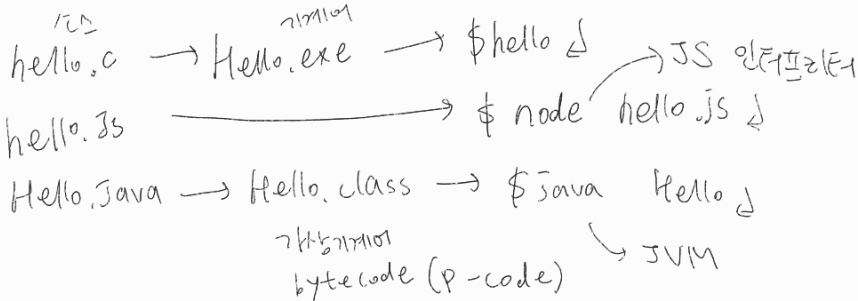


컴파일 과정에서 문법검사가 완료  
실행에 귀중한 단축 명령어인 bytecode로 변환  
인터프리터가 실행할 때 직접 읽는 것보다  
실행 속도가 빠르다.

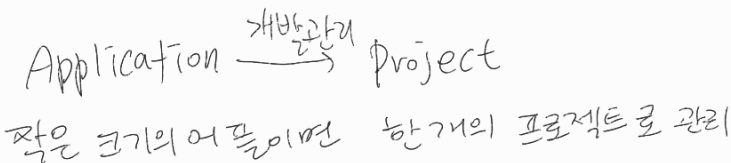
## \* 자바 컴파일과 실행



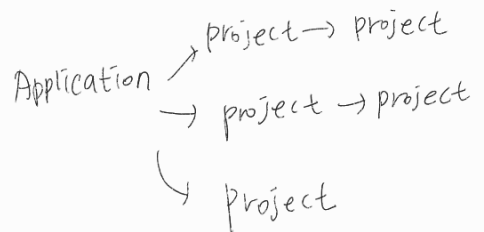
## \* compile vs 인터프리터 vs Hybrid



## \* 애플리케이션과 프로젝트 (기본 관리 단위)



큰 크기의 어플리케이션 개발시에 여러개의 프로젝트로 관리  
프로젝트 사이의 의존관계가 가능



프로젝트는 개발의 최소 관리 수단이다.  
효과적인 코드 관리를 위해 (비용 절감)

## \* 자바 어플리케이션을 프로젝트로 관리

### 1. 1단계

git repo. = project 관리폴더

## 2. 2단계

소스폴더와 class파일을 두는 폴더 분리

src/자바 소스파일

bin/.class파일

bitcamp-study \$javac -d bin src/Hello.java

↳ Compile 후 생성되는 .class 파일은 bin 폴더의 위치

이클립스는 컴파일러가 아니고 에디터일 뿐이다

bitcamp-study\$ java -classpath bin Hello

↳ bin폴더 안 들어간 상태에서 실행하고자 하는 경우 bin이 있는 경로

를 적어주는 것

## 3. 3단계

애플리케이션의 기능이 복잡해지고 규모가 커지면서 프로그래밍 소스 외에 다른 파일들도 필요하게 되었다.

bitcamp-study/

|

src/

└ main/

└└ java/

└└ resources

└ test/

└└ java/

└└ resources

|

bin

└ main/

Maven project  
해결 방향 찾기

\$javac -d bin/main src/main/java/Hello.java

\$java -cp bin/main Hello

## 4. 4단계

애플리케이션 규모가 커지면 단일 프로젝트로 관리하기 힘들다.

한 저장소에 여러 프로젝트를 둘 수 있도록 구조를 변경한다.

bitcamp-study/

|

java-basic

|

src/

└ main/

└└ java/

└└ resources

└ test/

- └ java/
- └ resources

|

bin

- └ main/

실무의 프로젝트는 3단계, 4단계 모두 있다.

1)repo. = project

2)repo.

- └ project 1

- └ project 2

