

# 학습문제

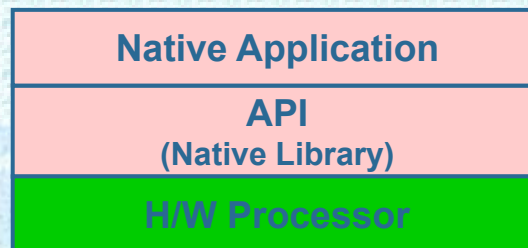
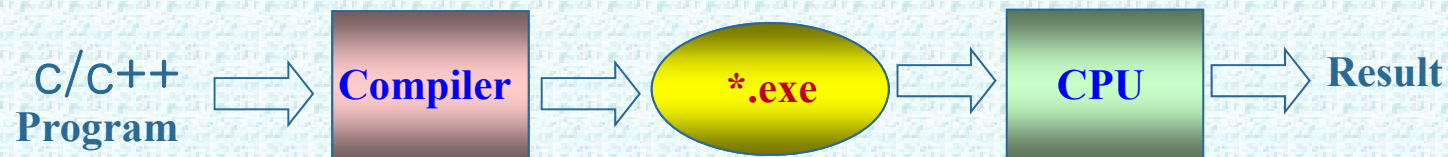
□ 일반적으로 컴퓨터에서 사용하는 C, C++, C#, Java 과 같은 프로그래밍 언어는 프로그램 실행하는 방식에 따라 Native Code 방식과 VM(Virtual Machine) Code 방식으로 구분된다.

□ 두 방식의 특징을 살펴보고 어떤 방식으로 프로그램이 실행되는지, 어떤 장단점이 있는지 다음 페이지 자료와 인터넷에서 해당 자료를 찾아 공부하시오

## □ Native 방식

### ▣ Native 방식

- ▣ 프로그래머가 C나 C++ 언어로 프로그램을 작성
- ▣ 다음에 C 나 C++ 컴파일러를 통해 실행할 컴퓨터의 기계어 코드 생성
- ▣ 컴퓨터의 CPU로 프로그램을 실행
- ▣ 장점 : 프로그램을 직접 CPU로 실행하므로 실행속도가 빠름
- ▣ 단점 : 작성한 프로그램을 환경이 다른 컴퓨터에서 실행하려 할 경우 플랫폼이 달라 처음부터 다시 개발해야 함. 생산성이 떨어짐.

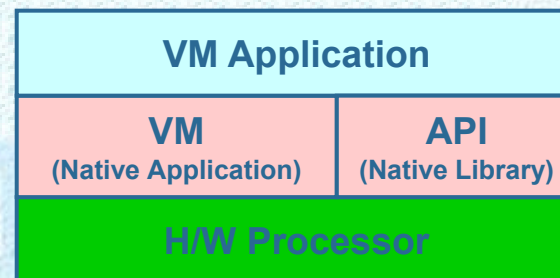


< Native 방식의 구조 >

## □ VM(Virtual Machine) 방식

### ▣ VM 방식

- ▣ 프로그래머가 Java 언어로 프로그램을 작성
- ▣ 다음에 Java 컴파일러를 통해 플랫폼 독립적인 VM 코드(예:바이트코드) 생성
- ▣ 생성된 VM 코드를 VM(예:JVM)으로 실행하여 결과를 출력
- ▣ 장점 : 플랫폼 독립적이므로 마이그레이션이 용이하고 생산성이 높음.
- ▣ 단점 : 프로그램을 CPU가 아닌 VM으로 실행하므로 실행속도가 다소 느림



< VM 방식의 구조 >



## □ VM 방식과 Native 방식의 비교

	VM 방식	Native 방식
H/W 독립성	<b>H/W 독립적</b> (H/W 변경시 콘텐츠수정 불필요)	<b>H/W 의존적</b> (H/W 변경시 콘텐츠 수정 필요)
시스템보안성	<b>보안성 확보됨</b> (컨텐츠가 시스템 메모리에 접근할 수 없음)	<b>보안성 확보 안됨</b> (컨텐츠가 시스템 메모리에 접근할 수 있음)
확장성/이식성	H/W 변경에 대한 <b>확장성</b> 우수함(VM S/W 변경으로 확장 및 이식 용이)	H/W 변경에 대한 확장성 취약
실행성능	코드 실행 부분을 S/W로 처리하여 코드 부분의 실행 속도 느림	코드 실행 부분을 H/W로 처리하여 복잡한 계산 <b>실행 속도</b> 빠름
컨텐츠 개발	검증과정이 불필요하여 컨텐츠 개발 및 제공 용이	검증과정이 필요하므로 컨텐츠 시험에 많은 시간 소요
용도	안정성이 확보되고 개발이 용이하므로 실행속도에 문제가 없는 게임 등과 같은 <b>End-User용 콘텐츠</b> 개발에 적합	안정성 확보가 어렵지만 실행성능이 보장되므로 동영상 Codec 등과 같은 <b>전문 솔루션</b> 개발에 적합