

## 섹션 1.

클라우드, CI/CD, 클래스/객체/인스턴스,  
static 키워드, 오버로딩/오버라이딩,  
추상화, 컴파일러/인터프리터

25/01/22  
김동규

# 1부 타임라인

30분 - 공유 세션

5분 - 쉬는 시간

15분 - 시험

10분 - 채점 및 토의

# 공유 세션

섹션 1 후반부에 상당히 많은 내용이 있었습니다.

한 번 복습 해봅시다.

**클라우드(Cloud)**

# #1 가상머신

전통적인 배포 방식 vs 가상화 배포방식

# #1 가상머신

가상화 배포방식의 장점으로서는 어떤 것들이 있을까요?

단점은 무엇이 있을까요?

## #2 오프프레미스, 온프레미스

클라우드 서비스는 오프프레미스 방식일까요, 온프레미스 방식일까요?

## #2 오프프레미스, 온프레미스

온프레미스 방식의 장점으로 어떤 것들이 있을까요?





<https://www.digitaltoday.co.kr/news/articleView.html?idxno=247374>

# **#3 IaaS, PaaS, SaaS**

IaaS, PaaS, SaaS 는 각각 어떤 단어의 약자일까요?

# #3 IaaS, PaaS, SaaS

IaaS (Infrastructure-as-a-Service)

“컴퓨터와 네트워크를 빌려주는 서비스”

# **#3 IaaS, PaaS, SaaS**

PaaS (Platform-as-a-Service)

“개발할 수 있는 플랫폼 제공 서비스”

# **#3 IaaS, PaaS, SaaS**

SaaS (Software-as-a-Service)

“소프트웨어 빌려쓰는 서비스”

# **#3 IaaS, PaaS, SaaS**

IaaS 와 PaaS 를 비교해봅시다.

유연성 / 이식성 / 운영비 효율성 의 관점에서.

# #3 IaaS, PaaS, SaaS

	IaaS	PaaS
유연성	높음	낮음
이식성	높음	낮음
운영비 효율성	낮음	높음

# #4 컨테이너와 도커

컨테이너란 무엇인가?



## #4 컨테이너와 도커

가상 머신 방식에서는 Hypervisor 위에 가상 머신을 여러 개 두고 OS를 각각 깔았었다.

컨테이너 방식의 경우, OS를 하나만 깔다.

그 OS 위에 Docker 가 있고, Docker 위에 컨테이너 별로 구분이 되어 있다.

## #4 컨테이너와 도커

도커(Docker)란 컨테이너 배포에 필요한 거의 모든 기능을 제공하는 플랫폼

이식성, 유연성, 운영비 효율성 세 마리 토끼를 모두 잡는 것

# #4 컨테이너와 도커

도커가 컨테이너를 만드는 과정

도커 파일 -> 도커 이미지 -> 도커 컨테이너

**CI / CD**

# CI / CD

CI / CD 는 어떤 단어의 약자일까요?

# CI / CD

필요성

파이프라인이란

**클래스 & 객체 & 인스턴스**

# 클래스 & 객체 & 인스턴스

클래스란 무엇인가요?

객체란 무엇인가요?

인스턴스란 무엇인가요?



# 클래스 & 객체 & 인스턴스

객체란 변수와 메서드들이 하나로 묶여있는 것

클래스란 객체를 만드는 설계도(blueprint)

인스턴스는 이러한 설계도를 이용해 생성된 객체

**static 키워드**

# static 키워드

클래스의 인스턴스가 아닌 클래스에 속하며 클래스의 변수, 메서드 등을 공유하는 데 사용  
모든 객체가 동일한 메모리를 참조하므로 메모리 사용량 줄어 들 수 있다.

# **static 키워드**

static 키워드의 단점은 무엇이 있을까요?

# 오버로딩 & 오버라이딩

# 오버로딩 & 오버라이딩

오버로딩(overloading)에 대해 설명해 보세요.

오버라이딩(overriding)에 대해 설명해 보세요.

# 오버로딩 & 오버라이딩

오버로딩이 성립하기 위한 조건

1. 메서드 이름이 같아야 한다.
2. 매개변수의 개수 또는 타입이 달라야 한다.
3. 반환 타입은 관계 없다.

# 오버로딩 & 오버라이딩

## 오버라이딩 (Overriding)

조상 클래스의 메서드를 자손 클래스에서 오버라이딩할 때

1. 선언부가 조상 클래스의 메서드와 일치해야 한다.
2. 접근 제어자를 조상 클래스의 메서드보다 좁은 범위로 변경할 수 없다.
3. 예외는 조상 클래스의 메서드보다 많이 선언할 수 없다.



# 오버로딩 & 오버라이딩

```
class Parent {  
    void parentMethod() {}  
}  
  
class Child extends Parent {  
    void parentMethod() {} // 오버라이딩  
    void parentMethod(int i) {} // 오버로딩  
  
    void childMethod() {}  
    void childMethod(int i) {} // 오버로딩  
    void childMethod() {} // ERROR. 중복 정의 되었음 (already defined in Child)  
}
```

추상화

# 추상화

추상화(Abstraction)란?

추상화의 목적

# 추상화

데이터 추상화 & 프로세스 추상화

**컴파일러 언어 & 인터프리터 언어**

# 컴파일러 언어 & 인터프리터 언어

컴파일러란?

인터프리터란?

# 컴파일러 언어 & 인터프리터 언어

	컴파일러	인터프리터
변환 단위	전체 변환	한 줄씩 변환
속도	컴파일 과정 자체는 시간 소요, 변환된 코드는 직접 실행돼 빠름	초기 시작은 빠르지만, 전체 코드 실행 시간은 느려질 수 있음
사용 예	C, C++, Go, Rust 등	Python 등
컴파일 과정 여부	코드를 수정할 때 컴파일 필요	코드를 수정했을 때 컴파일 필요 없음

# 컴파일러 언어 & 인터프리터 언어

JIT 컴파일러 (Just In Time Compiler)

코드 분석 -> 동적 컴파일 -> 최적화 -> 실행



# 컴파일러 언어 & 인터프리터 언어

JIT 컴파일러 (Just In Time Compiler)

장점과 단점에 대해 말해보세요.

# 쉬는 시간

시험 보기 전 잠시 5분 휴식 취하겠습니다.

# 시험 시간

15분 동안 시험 보겠습니다.

# 채점 및 토의 시간 (10분)

시험 문제는 좀 어땠나요?

헛갈리는 문제가 있었다면 한 번 얘기해 봅시다.

# 1부 종료

감사합니다