



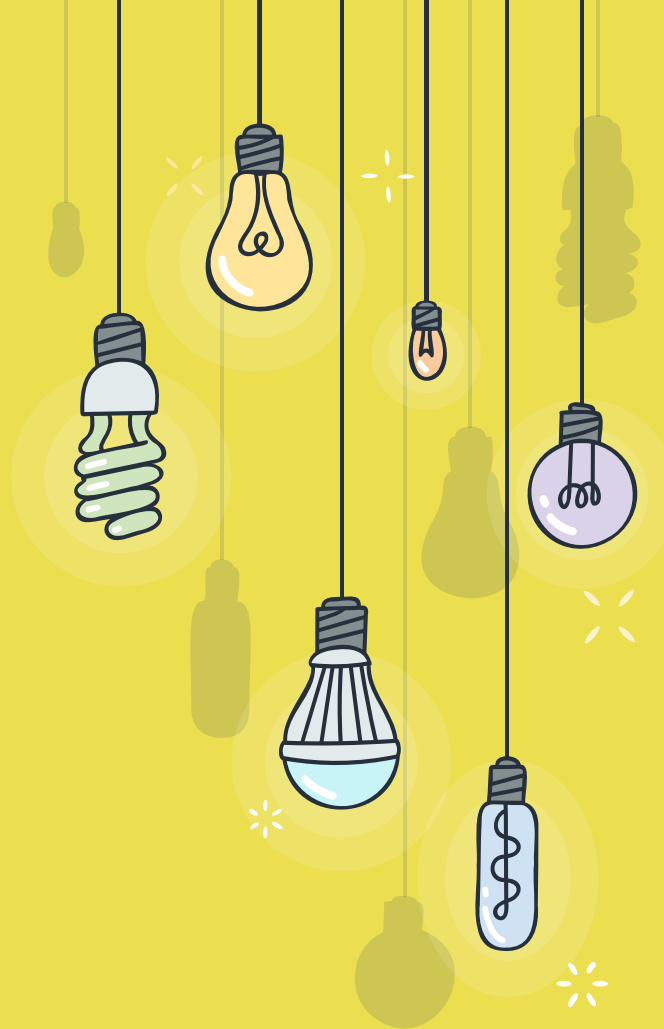
SEQUENCE DIAGRAM

1. sequence 다이어그램



1

시퀀스 다이어그램



* 상태 다이어그램과 시퀀스 다이어그램

+ 상태 다이어그램

- × 시간에 따라 각기 다른 상태의 흐름을 나타낸다

+ 시퀀스 다이어그램

- × 시스템이나 객체들끼리 주고 받는 메시지의 순서를 시간의 흐름에 따라 보여준다
- × 장점
 - ◆ 어떠한 시나리오로 흘러가는지 쉽게 표현 가능
 - ◆ 각 동작에 참여하는 시스템이나 객체들의 수행 기간 확인
 - ◆ 시스템의 동적인 구조를 보여주기 적합

* 시퀀스 다이어그램

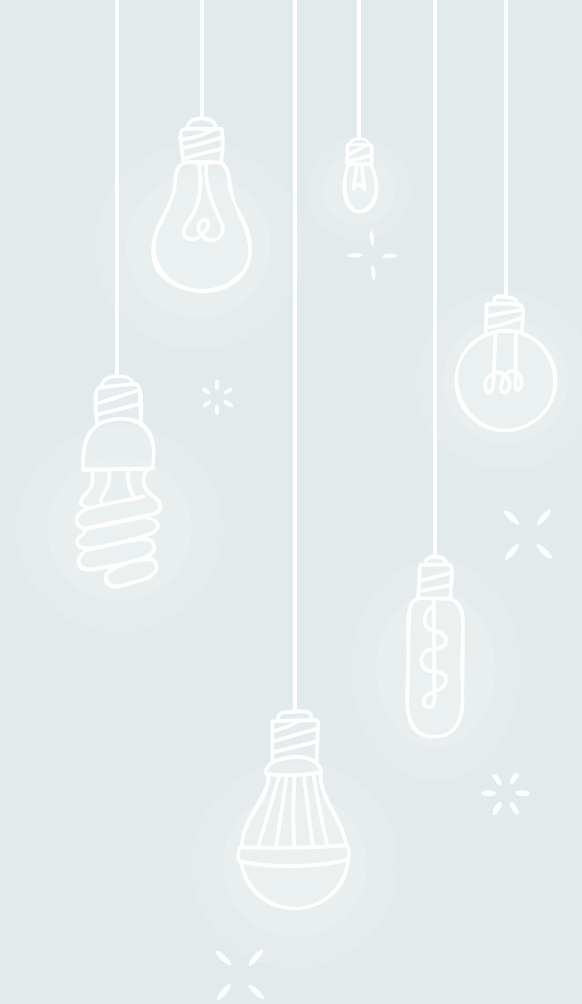
+ 생명선과 메시지

× 생명선

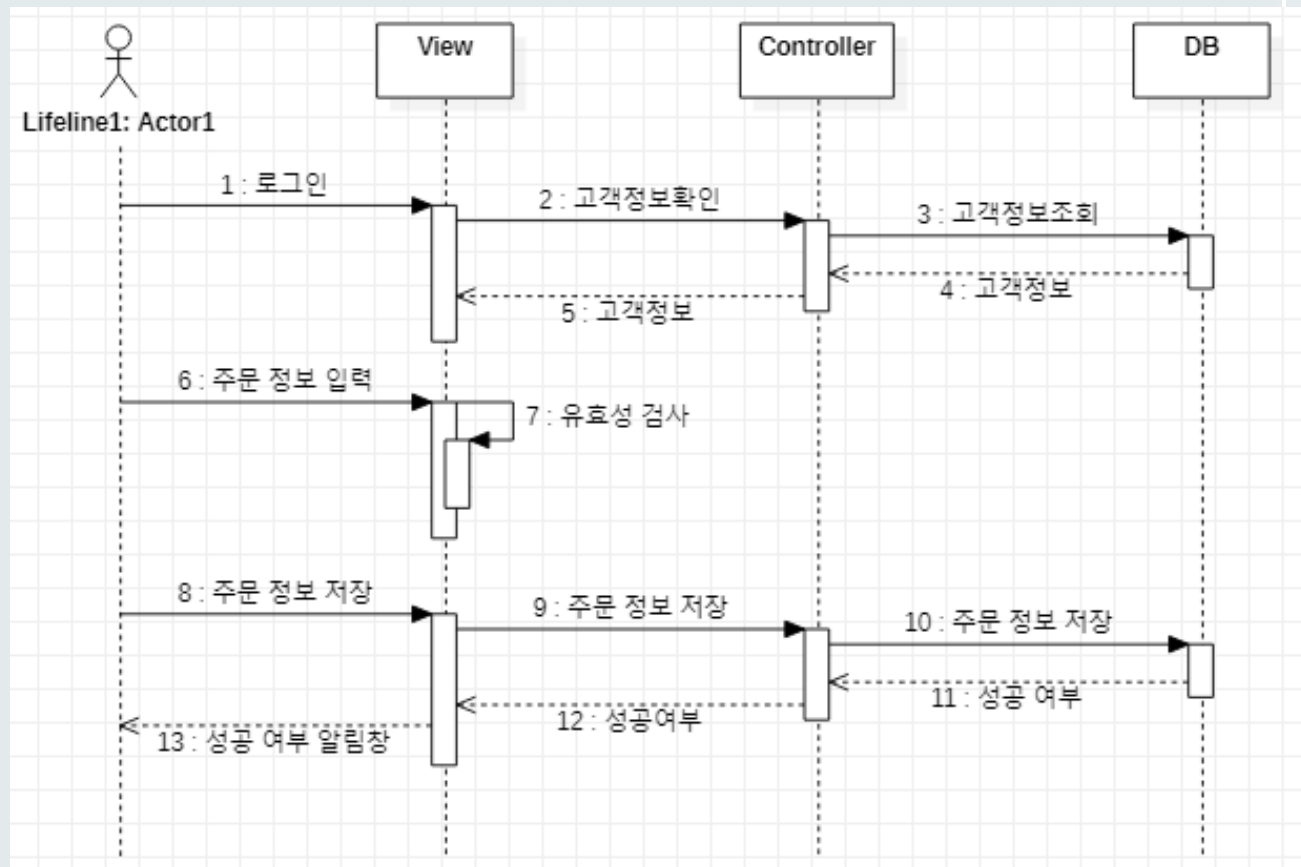
- ◆ 액터, 클래스 객체, 컴포넌트의 인스턴스 등, 상호작용에 참여하는 구체적인 대상 표현
- ◆ 생명선 끝에 X자로 끊겨 있으면 소멸(가비지 컬렉터에 넘기는 행동이 됨)

× 메시지

- ◆ 생명선 간에 전달되어 상태의 행위에 대한 호출



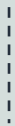



* 시퀀스 다이어그램



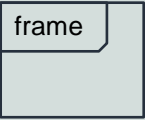


* 시퀀스 다이어그램 구성요소

+ 구성요소


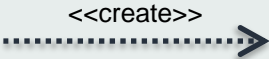

components	UML	설명
Actor		시스템으로부터 서비스를 요청하는 외부 요소로, 사람이나 외부 시스템을 뜻함
Object (객체)		메시지를 주고 받는 주체
Lifelines (생명선)		객체가 메모리에 존재하는 기간으로, 객체 아래쪽에 점선을 그어 표현함 객체 소멸(X)이 표시된 기간까지 존재함
Activations (활성 상자)		객체가 메시지를 주고받으며 구동되고 있음을 표현함

* 시퀀스 다이어그램 구성요소

components	UML	설명
Message (메시지)		객체가 상호 작용을 위해 주고받는 메시지
Destruction (객체 소멸)		해당 객체가 더이상 메모리에 존재하지 않음을 표현
Frame (프레임)		다이어그램의 전체 또는 일부를 묶어 표현한 것

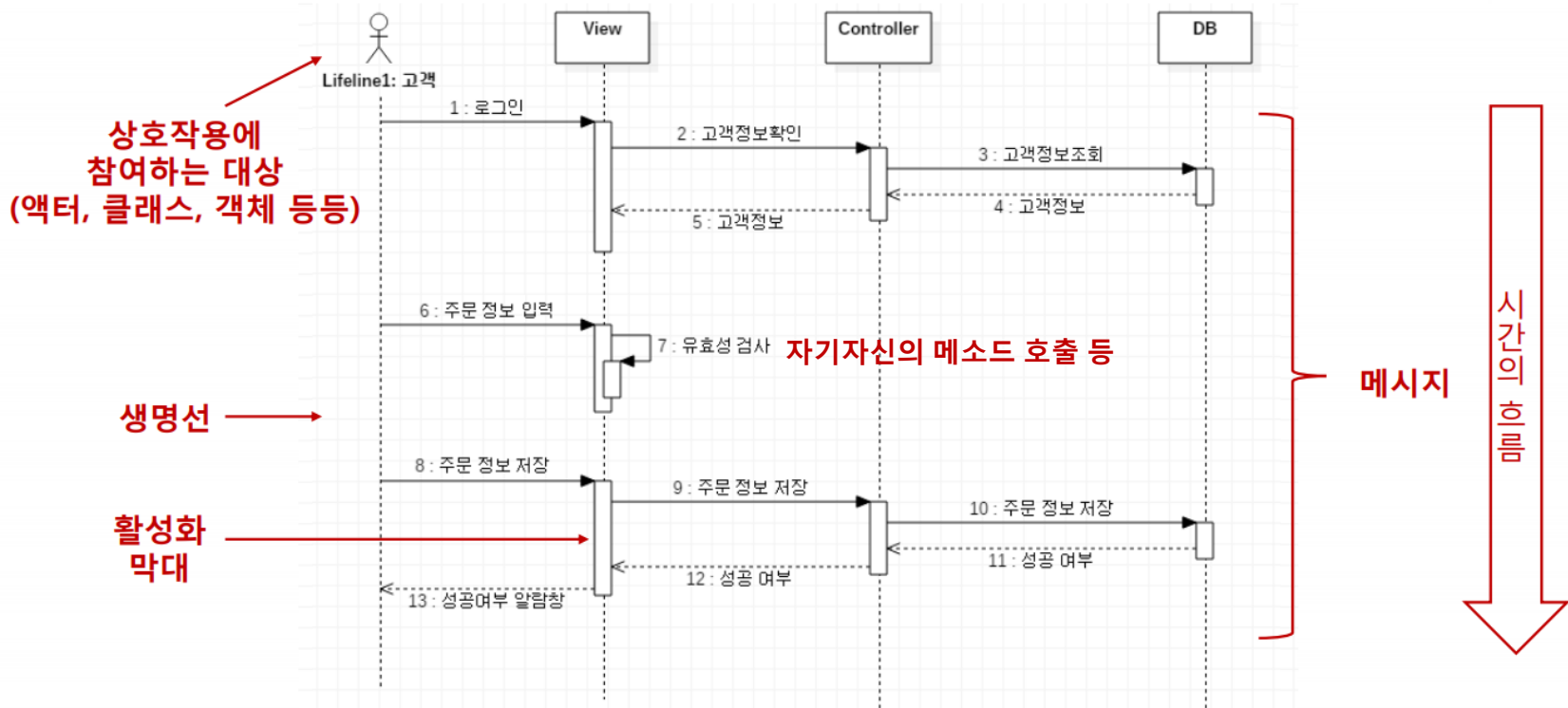
* 시퀀스 다이어그램 메시지 종류

+ 메시지 종류

메시지 유형	표현법	설명
비동기적 메시지		송신자를 대기 시키지 않음
동기적 메시지		송신자를 대기시킴
대답 메시지		동기적 메시지의 수행 결과
생성 메시지		생명선 생성
발견된 메시지		발신인을 알 수 없는 메시지
유실된 메시지		대상을 알 수 없는 메시지

* 시퀀스 다이어그램

역할 구성요소

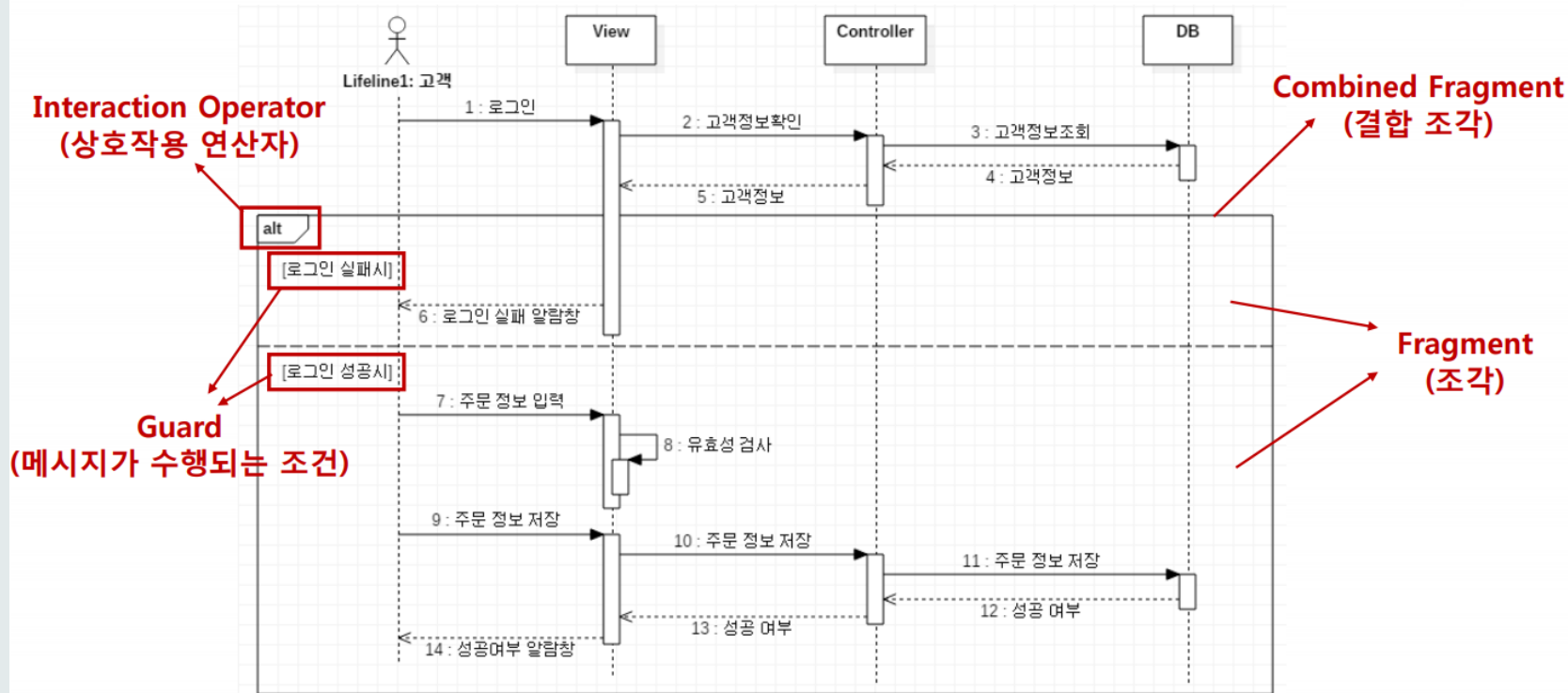


* 시퀀스 다이어그램 상호작용 연산자 종류

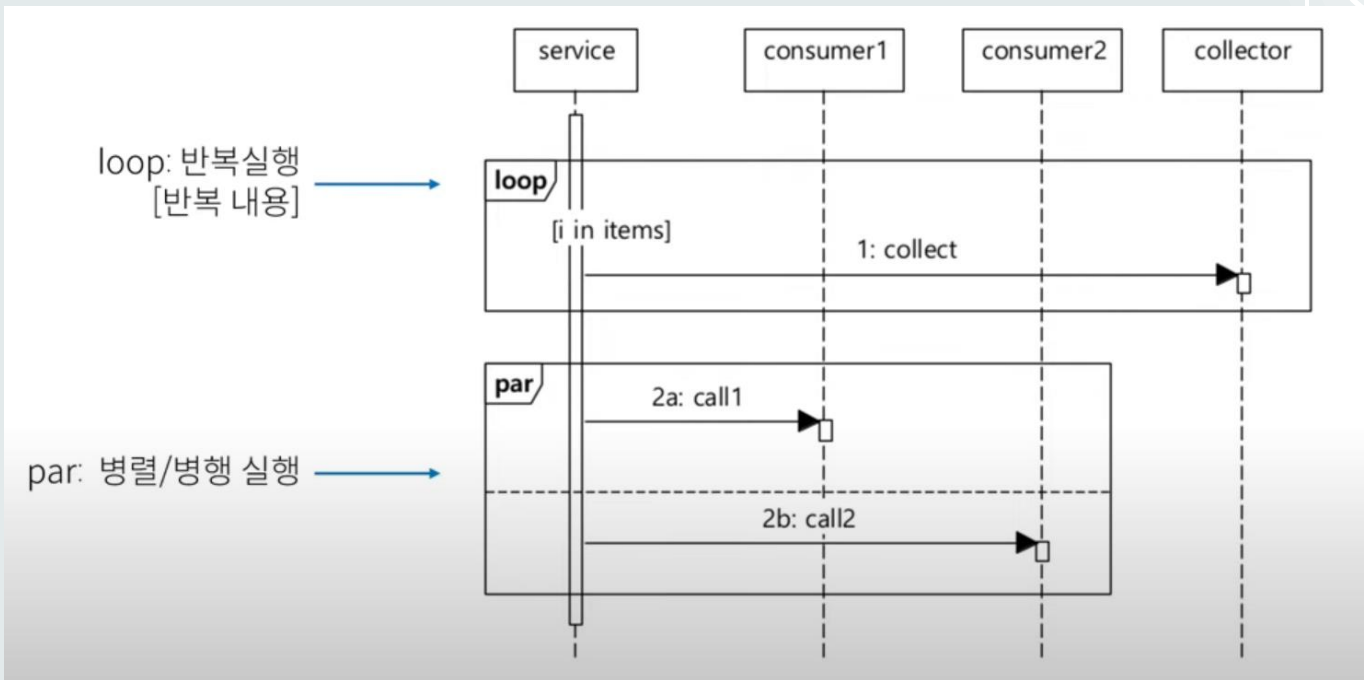
+ 상호작용의 연산자 종류

연산자 종류		설명
alt	대체	메시지의 대체 시퀀스를 포함하는 연산자이며 어떤 상황에서도 하나의 시퀀스만 발생함 -> if - else if 문에 해당하는 논리를 나타냄. 단, 모든 가드가 false이고 else가드가 존재하지 않으면 어떠한 조각도 실행되지 않음
opt	옵션	선택적 요소로 발생하거나 발생하지 않을 수 있는 시퀀스를 포함 -> if 문에 해당하는 논리를 나타냄. 대안을 하나만 제공
loop	반복	루프 상호작용 연산자는 반복적으로 실행되는 부분을 나타냄 가드 안에 단편이 실행되는 횟수를 지정할 수 있음
break	중단	보통 loop 연산자와 함께 쓰이며 조건이 true 일 경우 현재 실행을 포기하고 빠져나감

* 시퀀스 다이어그램 상호작용 제어흐름



* 시퀀스 다이어그램 상호작용 제어흐름



* 예제 1

식당

고객(Customer), 종업원(Employee), 요리사(Chef)

1. 고객이 종업원에게 음식을 주문한다(orderFood)
2. 종업원은 요리사에게 주문 받은 음식을 전달한다(demandFood)
3. 요리사는 음식을 조리한다(makeFood)
4. 요리사가 만든 음식을 종업원에게 전달한다(deliveryFood)
5. 종업원은 음식을 고객에게 전달한다(serve)

* 예제 2

카카오톡 친구관리

- 회원사용자, 서버, DB, 친구

1. 회원은 서버에 친구추가를 보낸다
2. 서버는 친구에게 친구추가 알림을 보낸다
3. 서버는 친구에게 친구 요청을 보낸다
4. 서버는 회원사용자에게 현재 친구요청상태를 보낸다
5. 회원 사용자는 친구추가를 취소 할 수 있다
6. 만약 친구가 친구수락을 하면 서버로 전송된다
 서버는 DB에 친구를 저장한다
 만약 친구가 친구거절을 하면
 서버는 거절메시지를 가지고 있다.
7. 서버는 각각 회원사용자와 친구에게 수락인지 거절인지의 알람을 전송한다
8. 친구는 회원사용자를 친구삭제를 할 수 있다
9. 회원사용자는 친구를 친구삭제할 수 있다
10. 그러면 서버는 DB의 친구목에서 삭제한다

THANKS!

+ Any questions?

