



# 11강. 액티비티 다이어그램

컴퓨터과학과 김희천 교수



# 목차

- 1 액티비티 다이어그램 개요
- 2 객체와 객체 흐름
- 3 조건과 병합 노드
- 4 병렬 처리와 확장 영역
- 5 시그널과 파티션





## Chapter. 1

---

# 액티비티 다이어그램 개요

# 1. 액티비티 다이어그램

- + 업무 흐름이나 계산 과정을 액션들의 흐름을 통해 단계적으로 표현
  - × 흐름도(또는 순서도)와 유사
  - × 순차 실행, 조건에 따른 분기, 병행 처리 등의 제어 흐름을 표현할 수 있음
- + 비즈니스 프로세스의 표현 또는 오퍼레이션의 제어 흐름 표현
  - × 둘은 추상화 수준이 다름
- + 4+1 뷰에서 프로세스 뷰를 표현하는 UML 다이어그램

## 2. 액티비티와 액션 (1/3)

### 액티비티



- 액티비티는 작업의 실행을 의미하는 것으로 일련의 액션들과 이들의 제어 흐름으로 표현됨
- 액션보다 상위 개념의 행위

#### + 액션은 액티비티를 수행하기 위해 필요한 단일 작업

- × 액티비티에 포함되는 하나의 세부 단계
- × 더 이상 분해되지 않는 단일 작업

#### + 액티비티 다이어그램의 구성 요소

- × 액션과 액티비티, 제어 노드, 화살표(제어/객체 흐름), 객체 노드 등

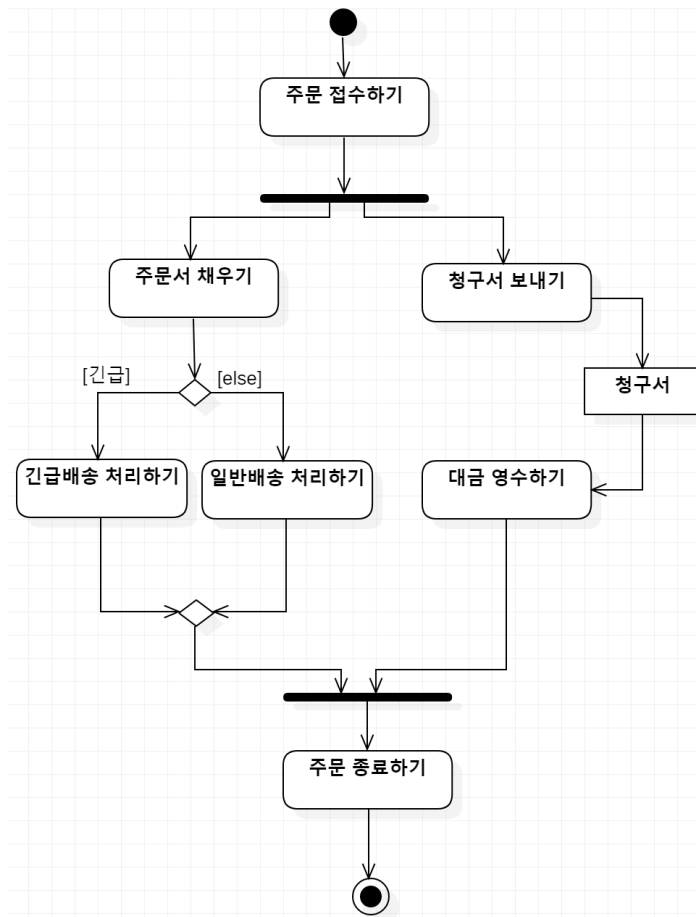
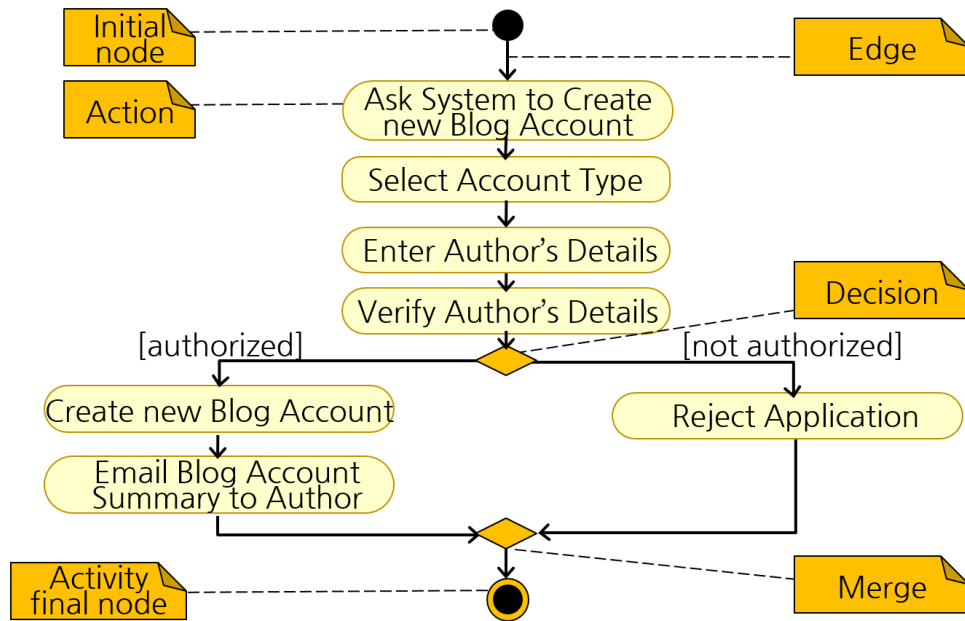
## 2. 액티비티와 액션 (2/3)

### + 표기법

- × 액티비티와 액션의 표기법은 동일하며 모서리가 둥근 사각형임
- × 액티비티 내부에 액션들이 포함되며 액티비티의 이름은 상단에 위치함
- × 액션 사이 제어 흐름은 실선 화살표(입력 흐름과 출력 흐름)로 나타냄
- × 예) '라면 끓이기'는 액티비티, '물 붓기', '물 끓이기', '라면 넣기' 는 액션



## 2. 액티비티와 액션 (3/3)



### 3. 액티비티 시작과 종료 (1/3)

#### + 시작 노드는 액티비티의 출발을 의미

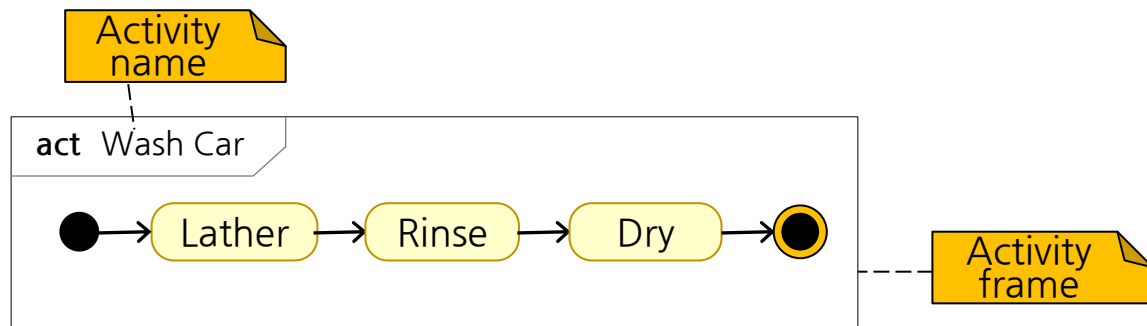
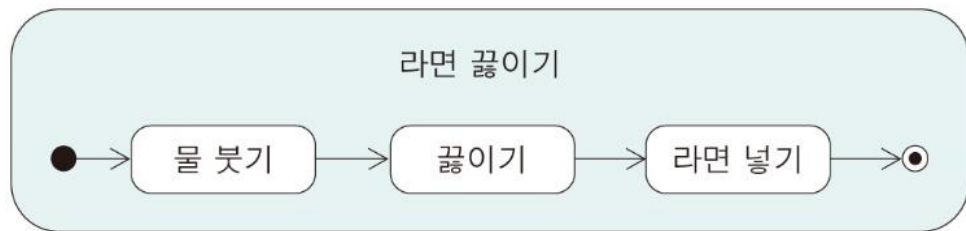
- 대개 하나의 액티비티에서 시작 노드는 하나가 존재할 수 있음
- 입력 데이터나 시그널을 받는 경우, 시작 노드가 없기도 함
- 검은 색 원으로 표시

#### × 종료 노드는 액티비티의 종료를 의미

- 정확히는 액티비티 종료 노드(activity final node)라고 함
- 액티비티 내의 모든 흐름을 종료시킴
- 일반적으로 하나가 존재하나 1개일 필요는 없음
- 검은 원을 포함한 원으로 표시



### 3. 액티비티 시작과 종료 (2/3)



### 3. 액티비티 시작과 종료 (3/3)

#### + 흐름 종료 노드

- × 해당 흐름 하나만을 종료하는 노드
- × 전체 액티비티의 종료가 아님
- × 액티비티 내부에서 병렬적 흐름이 발생할 때 특정 흐름을 종료하기 위한 것
- × x 표시를 포함한 원으로 표시

섞어주기





## Chapter. 2

# 객체와 객체 흐름

# 1. 객체와 객체 흐름

## + 객체는 액션의 입력이나 출력으로 사용되는 데이터

- 물리적인 객체를 의미할 때도 있음
- 사각형으로 표시
- 이름에 밑줄을 그어 인스턴스임을 강조할 수 있음

## × 객체 흐름

- 객체와 액션 사이의 화살표
- 객체를 제공하거나, 객체를 입력으로 받아 사용하는 액션을 표현
- 액티비티에서 데이터의 흐름을 강조하는 표현



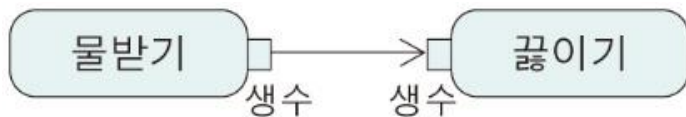
## 2. 액티비티 다이어그램과 객체

### + 액티비티 다이어그램에서 객체를 사용하는 이유

- × 액티비티 다이어그램은 객체 노드를 사용하여 데이터의 흐름을 표현
- × 객체의 생성, 사용, 변경을 명세할 수 있음
- × 액션을 통해 객체의 상태 변화를 명세함

### 3. 입력핀과 출력핀

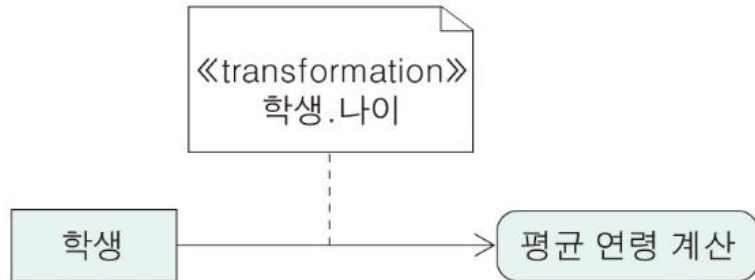
- + 입출력 핀(pin)을 사용하여 객체 흐름을 표현하기도 함
  - × 핀은 액션의 입력이나 출력 파라미터로, 액션에 입출력이 필요함을 강조하는 표현
- + 출력핀은 액션의 결과로 객체가 출력됨을 의미
- + 입력핀은 객체가 액션의 입력으로 사용됨을 의미



## 4. 객체 흐름에 스테레오 타입 사용하기 (1/2)

### + 객체 전체가 아닌 일부만 전달되는 상황을 표현할 때

- × 객체 흐름에 노트를 추가하고 《transformation》 스테레오타입을 사용
- × 복수 개의 정보를 가진 객체에서 필요한 부분만을 사용하는 것을 표현하는 것
- × 예: 학생 객체에서 나이 데이터만 사용하는 경우



## 4. 객체 흐름에 스테레오 타입 사용하기 (2/2)

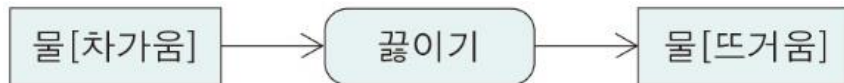
- + 특정 조건을 만족하는 객체들만 전달되는 상황을 표현할 때
  - × 객체 흐름에 노트를 추가하고 《selection》 스테레오타입을 사용
  - × 예: 학점이 3.0이 넘는 학생 객체들을 사용하는 경우





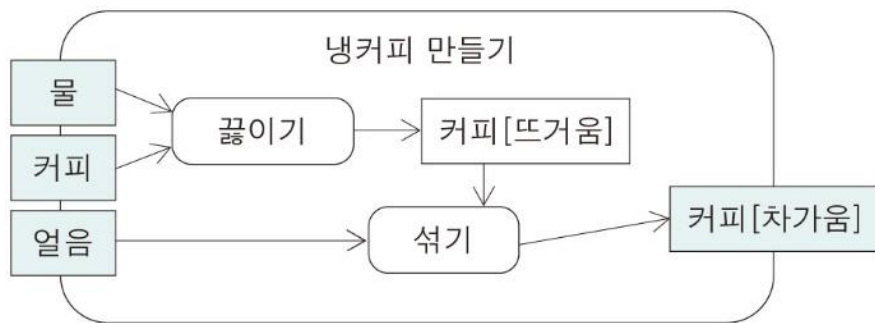
## 5. 객체의 상태 표현하기

- + 액션들의 흐름을 따라가다 객체의 상태가 바뀌는 것을 표현
- + [ ]를 사용하여 상태를 표시



## 6. 액티비티의 입출력

- + 액티비티도 입출력 표현을 위해 객체를 사용할 수 있음
  - × 액티비티에서 사용되는 입출력을 표현하기 위해 객체 노드를 사용
  - × 액티비티 파라미터라고 함
- + 액티비티의 좌우 경계 부분에 객체 노드를 위치시켜 표시함
- + 액티비티의 시작 노드와 종료 노드가 생략될 수 있음





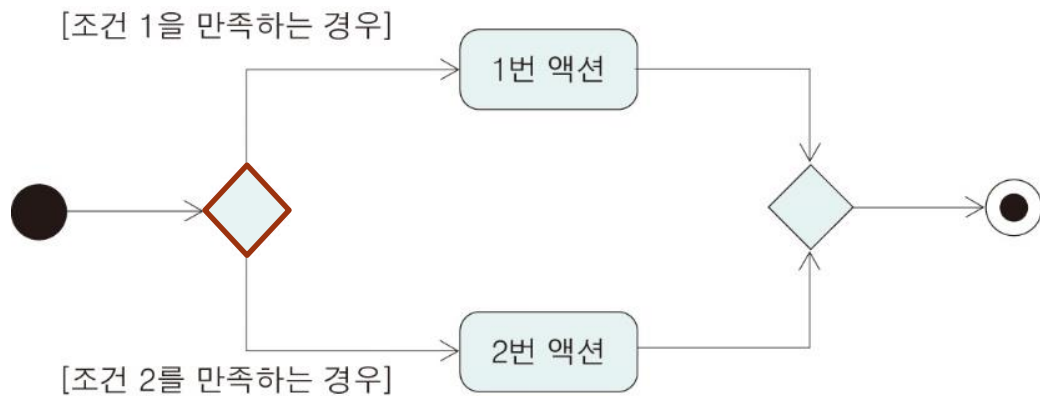
## Chapter. 3

# 조건과 병합 노드

# 1. 조건 노드

## + 조건에 따른 선택적 흐름의 실행을 표현

- ×마름모로 표시
- ×1개의 입력 화살표와 2개 이상의 분기를 가짐
- ×조건 노드에서 나오는 흐름에 조건(guard condition)을 [ ] 안에 표시함



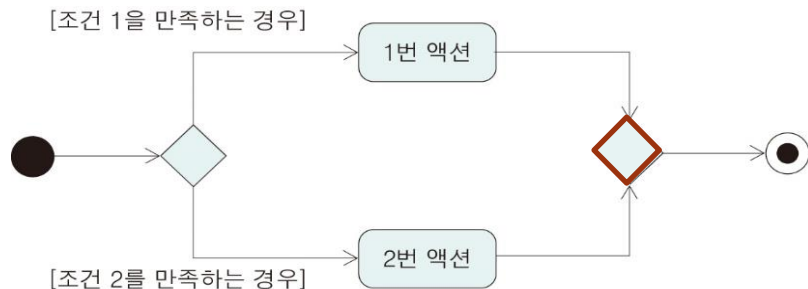
## 2. 병합 노드

### + 조건을 표시할 때 주의할 점

- × 모든 조건들은 서로 배타적이어야 함
- × 모든 경우를 조건들로 표시해야 함
- × [else]와 같은 조건을 사용할 수 있음

### + 조건에 따른 액션 흐름들은 병합 노드로 합쳐짐

- × 병합 노드의 진입 흐름 중 하나가 완료되면 진행될 수 있음





## Chapter. 4

---

# 병렬 처리와 확장 영역

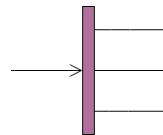
# 1. 포크와 조인 (1/2)

✚ 병렬 처리를 표현하기 위해 포크(fork)와 조인(join)을 사용

- 수직(또는 수평)의 굵은 막대로 표시

## ✕ 포크

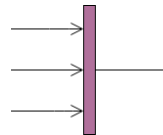
- 포크 뒤의 흐름은 2개 이상으로 나뉘어 동시에 진행됨



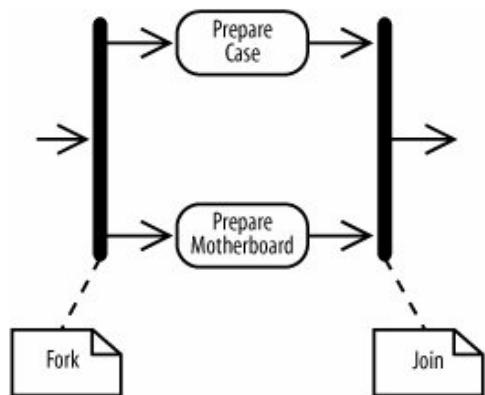
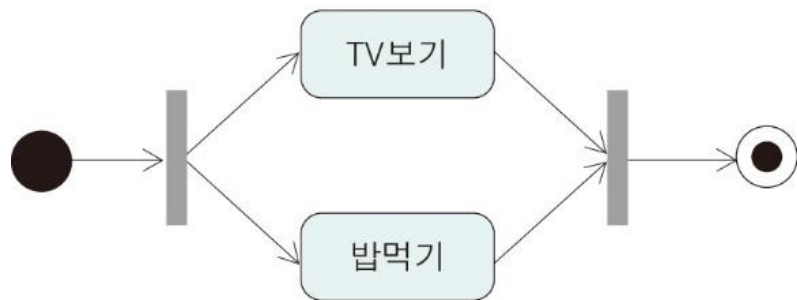
## ✕ 조인

- 동시 진행되었던 흐름들은 조인을 통해 동기화됨

- 동시 수행되었던 (조인의 진입) 흐름들이 모두 종료되어야 조인을 통해 다음 흐름으로 이어짐(동기적)



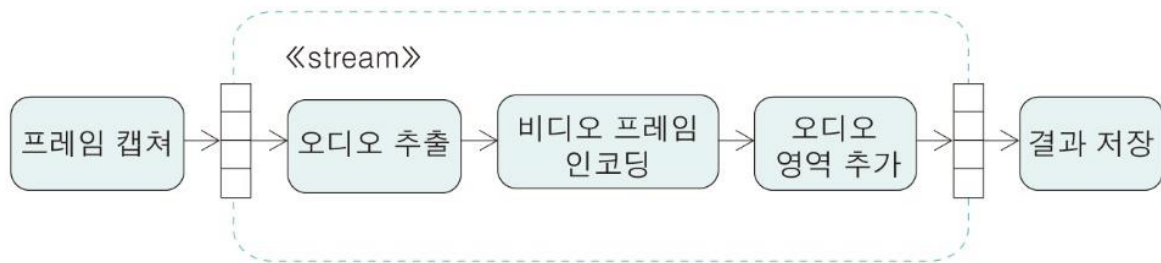
# 1. 포크와 조인 (2/2)





## 2. 확장 영역

- + 입력 객체가 컬렉션(collection)일 때, 이것의 모든 요소들에 수행되는 액션들을 하나의 그룹으로 표현
  - × 확장 영역으로 액션들을 포함하는 모서리가 둥근 사각형을 점선으로 표시
  - × 개별 요소들로 나뉘어 연속적으로 처리된 후 다시 조합됨
  - × 양쪽에 연결된 4개의 사각형(확장/객체 노드)을 각각 위치시켜 입력과 출력을 표현함



### 3. 확장 영역의 종류

#### + 《parallel》

× 요소들을 병행 처리함

#### + 《iterative》

× 요소들을 하나씩 순차적으로 처리함. 하나의 요소가 모든 액션들에서 처리된 후, 다음 요소의 처리를 시작함

#### + 《stream》

× 스트림으로 처리함. 첫 번째 액션이 하나의 요소를 처리한 후, 바로 다음 요소를 처리하는 것



## Chapter. 5

---

# 시그널과 파티션

# 1. 시그널

## + 시그널 주고받기는 외부의 사람, 외부의 시스템 또는 프로세스와의 상호 작용을 의미

- 이벤트의 발생으로 메시지를 외부로 보내거나 외부로부터 받는 것
- 메시지는 소프트웨어 설계자 관점이며 시그널은 시스템 설계자 관점

## ×예

- 신용 카드 결제에서 카드의 검증을 위해 카드 회사와 상호 작용하는 경우, 소프트웨어가 카드 회사에 요청을 보내고 응답을 받는 것
- 버튼을 클릭할 때, 해당 이벤트 처리가 수행되어야 함
- 고객에게 배송의 지연을 알려줌

## 2. 시그널의 송수신 (1/2)

### + 시그널 수신 액션

- × 시그널 발생을 기다리는 액션
- × 오목 오각형으로 표시하며 액션의 이름이 시그널의 이름
- × 시그널을 수신하면 연결된 다음 액션을 실행시킴
- × 시그널을 받으면 어떻게 반응해야 하는지 알고 있으나 언제 올지는 모름
- × 입력 흐름이 없는 시그널 수신은 액티비티의 시작점이 됨



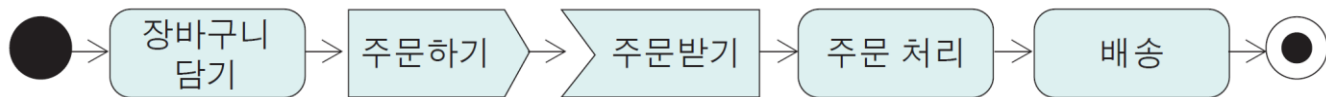
## 2. 시그널의 송수신 (2/2)

### + 시그널 송신 액션

- 시그널을 생성하여 전송(블록 오각형)
- 외부의 참여자에게 시그널을 보내면, 외부 참여자가 적절히 반응할 것으로 기대하나 응답을 기다리지 않음

### × 시그널의 송수신은 비동기적

- 시그널을 보내고 응답이 오기를 기다리지 않고 다음 액션을 바로 수행할 수 있음
- 시그널을 동기화하려면 시그널 송신과 수신을 나란히 위치시킴



### 3. 인터럽트 (1/2)

- + 시그널을 이용하여 인터럽트 상황을 표현할 수 있음

- × 특정 사건이 발생하면, 영역 내 모든 프로세스가 종료되는 상황을 표현할 때 사용함

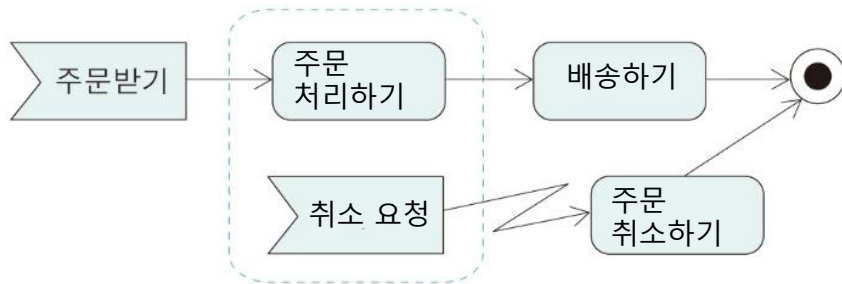
- + 인터럽트 영역

- × 인터럽트가 발생할 수 있는 영역으로 취소가 가능한 액션들을 포함
  - × 모서리가 둥근 점선의 사각형으로 영역을 표시

### 3. 인터럽트 (2/2)

#### + 인터럽트의 표시

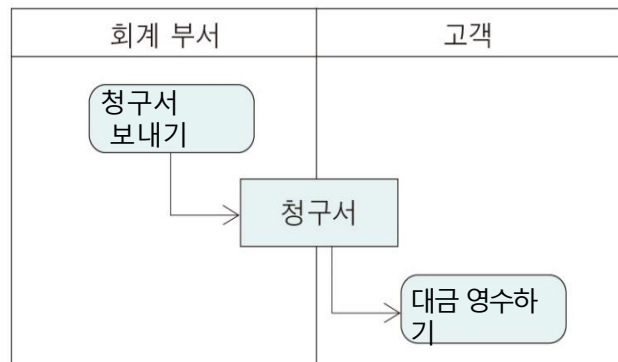
- × 인터럽트를 발생시키는 이벤트에서 인터럽트 발생 시 실행할 액션으로 지그재그 모양의 화살표가 나감
- × 예) 모든 주문이 처리되고 배송된 후에 종료됨. 만약 주문 처리 중에 취소 요청이 발생하면 영역 내 전체 프로세스가 종료됨. 단, 배송이 시작되면 주문 취소를 할 수 없음.





## 4. 파티션

- + 액션에 책임을 지는 참여자를 표현하기 위해 파티션 또는 레인으로 구분함
  - × 액티비티에 포함된 액션들을 수행하는 주체는 하나 이상이 될 수 있음
- + **스웜레인(swimlane)**이라고도 함
  - × 세로 또는 가로로 구역을 구분하여 담당 주체나 역할을 표시함

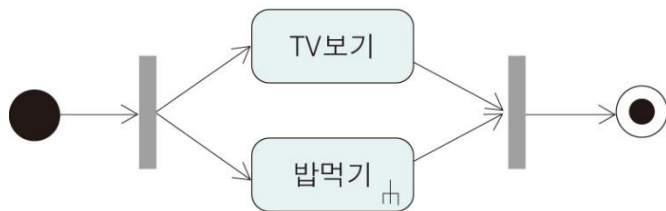


## 5. 액티비티 호출과 재사용성 (1/2)

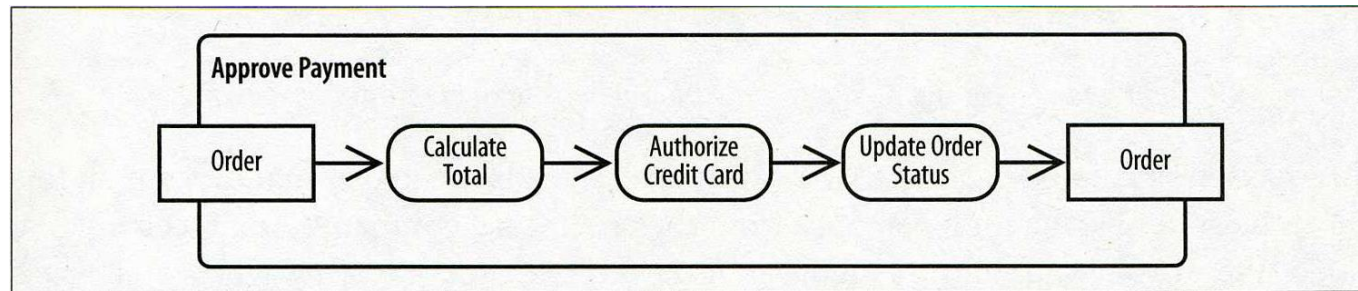
### + 호출 액티비티 액션

- × 다른 액티비티를 호출하는 액션
- × 가독성과 재사용성을 높이기 위해 액티비티를 별도의 다이어그램으로 분리
- × 액티비티 다이어그램이 너무 커질 때 또는 자주 사용되는 일련의 액션들이 중복되어 표현될 때 별도의 다이어그램을 그림
- × 분리된 다이어그램은 다른 액티비티 다이어그램 내에서 호출될 수 있음

### + 액션 노드 내에 포크 모양의 아이콘을 표시



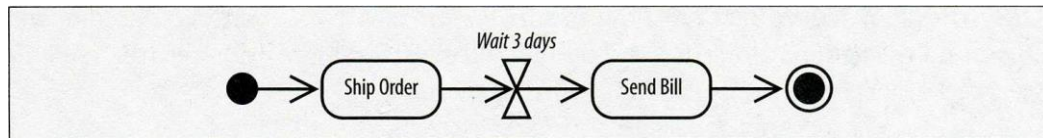
## 5. 액티비티 호출과 재사용성 (2/2)



## 6. 타임 이벤트 - 교재에 없음

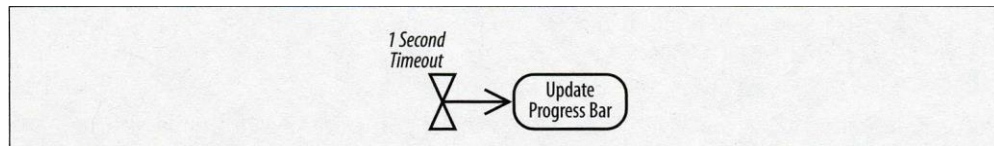
### + 액티비티의 실행 흐름에서 시간적 요인을 표현해야 하는 경우

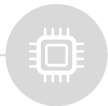
- × 모래시계 기호로 타임 이벤트의 발생을 표시함
- × 정해진 시간을 기다려야 하는 상황
- × 입력 흐름을 가진 타임 이벤트는 한번만 일어남



### + 규칙적 시간 간격을 두고 반복되는 상황을 표현

- × 입력 흐름이 없는 타임 이벤트는 주기적으로 반복됨을 의미
- × 시작 노드를 대신할 수 있음





다음강의

## 12강. 상호작용 다이어그램

