

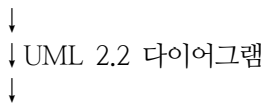
1. UML 개요

UML : 통합 모델링 언어(unified modeling language) 약어이다.

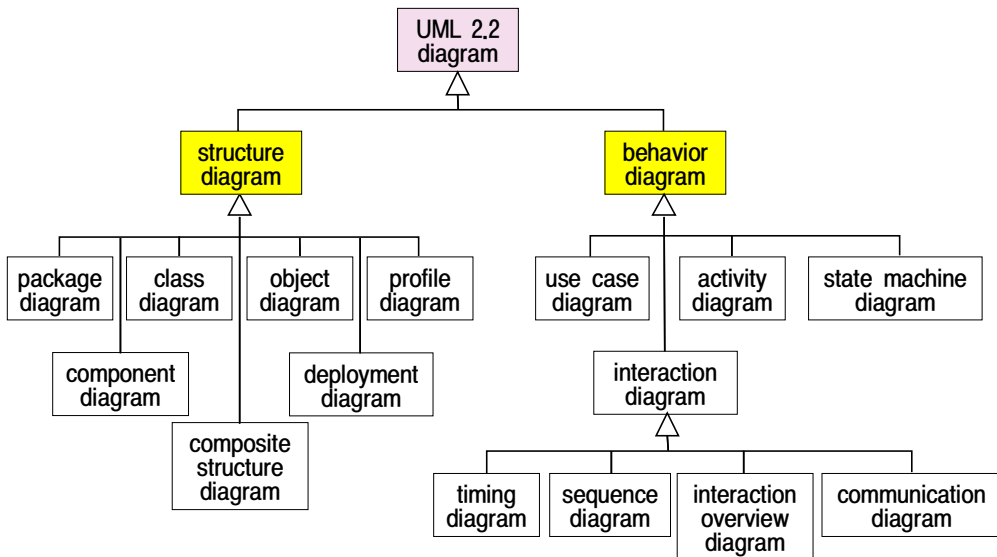
UML은 단어 그대로 **통합 모델링 언어**이다. 시스템을 **시각적으로 표현**하는 것이다.

<UML 다이어그램>

- UML은 버전이 증가하면서 다이어그램 이름이 변경된 것도 있고, 수가 증가되고 있다.
- 예 : 상태(state) 다이어그램은 상태기계(state machine) 다이어그램으로 변경되었다.



<UML 2.2 다이어그램>



UML 2.2 다이어그램 : 구조 다이어그램 + 행위 다이어그램 = 7 + 7 = 14(개)

-
- 구조 다이어그램(structural diagram) : 시스템의 정적인 구조를 나타냄
 - 행위 다이어그램(behavioral diagram) : 시스템을 구성하는 객체들의 동적인 관계

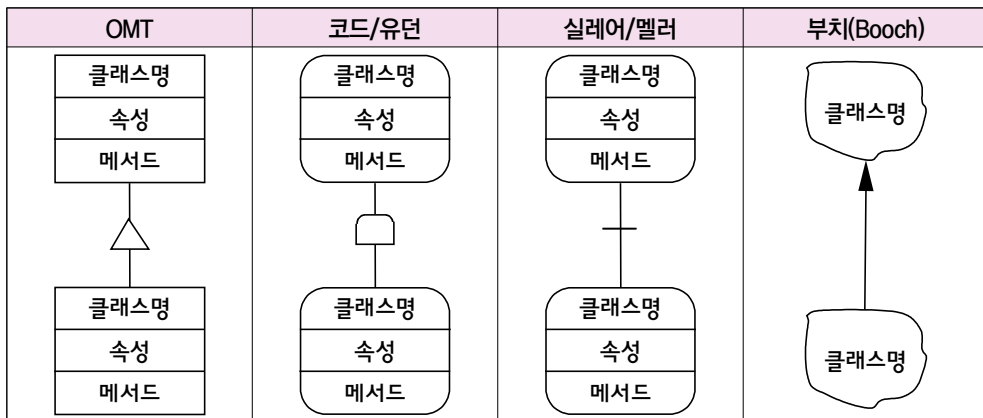
〈UML 정리〉

- UML은 소프트웨어 공학에서 사용되는 표준화된 **범용 모델링 언어**이다.
- 시스템의 유형과 규모에 따라서 필요한 다이어그램을 선택하여 사용한다.
- UML은 시스템 개발 활동의 전 과정에서 사용될 수 있다.
- UML은 객체지향 소프트웨어 개발에서 산출물 명세화, 시각화, 문서화할 때 사용한다.
- UML은 시스템 개발에 사용되는 **개발 방법론과 무관**하다.
- UML은 **객체지향 개발 방법론**이 아니다.
- 하지만, UML은 객체지향 설계 표현 방법의 표준으로 받아들여지고 있다.
- UML은 시스템 개발에 사용되는 CASE 도구에 무관하다.
- UML은 시스템 개발에 사용되는 **프로그래밍 언어에 무관**하다.
- UML로 모델링 한 것은 C++, Java 등의 원시코드로 변환할 수 있다.

◆ 객체지향 개발 방법론

부치(booch)의 OOD	<ul style="list-style-type: none"> • OOD는 Object Oriented Design 약어이다. • 정적 모델과 동적 모델로 표현
코드(coad)/유던(yourdon) OOAD	<ul style="list-style-type: none"> • OOAD는 Object-oriented analysis and design 약어이다. • 객체지향 = 객체 + 분류화 + 상속 + 객체 통신
실라이어(shlaer)/멜러(mellor) OOSA	<ul style="list-style-type: none"> • OOSA는 Object Oriented Structured Analysis 약어이다. • 시스템, 도메인, 서브시스템, 객체, 상태, 프로세스의 6계층으로 분석 • 시스템을 최상위 수준의 시스템부터 최하위 프로세스 수준까지 표현
럼보(rumbaugh)의 OMT	<ul style="list-style-type: none"> • OMT는 Object Modeling Technique 약어이다. • 객체 모델링 / 동적 모델링 / 기능 모델링 등을 이용

◆ 객체지향 개발 방법론에서 상속을 표기하는 방법



- 객체지향 개발 방법론에 따라 표기법 서로 다르다.
- 레셔닐사가 객체지향 개발의 표기법만이라도 통일하자고 제안하였다.
- 1997년, UML은 OMG(Object Management Group)에 의해 표준으로 채택되었다.

1. 구조 다이어그램(structural diagram) : 시스템의 정적인 구조를 나타냄

① 클래스 다이어그램(class diagram)

- 시스템을 구성하는 클래스들 사이의 관계를 표현한다.
- 시스템 분석 단계와 설계 단계에 걸쳐 여러 번 작성된다.

② 객체 다이어그램(object diagram)

- 객체 다이어그램은 특정 시점의 시스템에 포함된 객체들의 모습(snapshot)을 기술한 것이다.
- 객체 다이어그램은 클래스의 객체(instance)들의 관계를 기술한 것이다.(instance diagram)
- 객체 다이어그램은 클래스 다이어그램보다 더 구체적이다.

③ 컴포넌트 다이어그램(component diagram)

- 컴포넌트 구조 사이의 관계를 표현한다.
- 컴포넌트가 이용하는 인터페이스와 내부 구성요소 사이의 관계를 표현한다.
- 소프트웨어 컴포넌트 사이의 의존관계를 묘사한다.
- 하나 이상의 클래스를 구현하여 컴포넌트를 구성한다.
- 컴포넌트기반 개발의 가장 큰 장점은 재사용이다.

④ 복합체 구조 다이어그램(composite structure diagram)

- 복합체 구조 다이어그램은 UML 2에 새로 포함되었다. 컴포넌트기반개발 지원(CBD)
- 클래스 다이어그램으로만 표현이 가능했던 시스템 내부구조를 보다 상세하게 설계할 수 있다.
- 복합체 구조의 클래스와 컴포넌트 내부구조를 표현한다.
- 복잡한 객체를 분할하여 여러 부분으로 나누어 표현할 수 있다.

⑤ 배치 다이어그램(deployment diagram)

- 소프트웨어, 하드웨어, 네트워크를 포함한 실행 시스템의 물리구조를 표현한다.
- 시스템을 구성하는 물리적인 노드와 통신경로, 그리고 컴포넌트의 수행 환경을 표시한다.

⑥ 패키지 다이어그램(package diagram)

- 클래스나 유스케이스 등을 포함한 여러 모델 요소들을 그룹화해 패키지를 구성한다.
- 시스템의 컴파일 시 계층적구조를 기술하는 다이어그램이다.
- 패키지들 사이의 관계를 표현한다.

⑦ 프로필 다이어그램(profile diagram)

- 스테레오 타입 «profile»로 표시한다.
- 스테레오 타입 클래스, 프로파일 등의 고정 관념을 표시한다.
- 프로필만 간단하게 구성된 다이어그램이다.

2. 행위 다이어그램(behavior diagram) : 시스템을 구성하는 객체들의 동적인 관계

① 사용예 다이어그램(use-case diagram) - 사용자 측면에서 본 시스템의 요구사항

- 사용예(use-case, 유스케이스)와 행위자(actor) 사이의 관계를 연결한다.
- 사용예 다이어그램은 시스템 설계에서 분석단계에 해당한다.

② 활동 다이어그램(activity diagram) - 제어 및 자료흐름(순서도)

- 단계별 작업흐름(workflow)을 표현한다.
- 업무 처리 과정이나 연산이 수행되는 과정을 표현한다.

③ 상태 머신 다이어그램(state machine diagram) - 이벤트에 대한 객체들의 상태 변화

- 객체의 상태 변화를 그림으로 나타낸 것이다.(객체의 생명주기를 표현)
- 객체 상태는 이벤트 또는 메시지를 주고받음으로써 변화될 수 있다.
- 개별 클래스 행동이나 여러 클래스의 특정 통신 행동을 하나의 머신으로 지정할 수 있다.

④ 순차 다이어그램(sequence diagram) - 객체들 사이의 메시지 교환(시간흐름)

- 시간흐름에 따른 객체 사이의 상호작용을 표현한다.
- 객체들 사이의 메시지 교환을 시간흐름을 기준으로 시각화하여 나타낸 것이다.
- 시스템 실행 시 생성, 소멸되는 객체를 표기하고 객체끼리 주고받는 메시지를 나타낸다.

⑤ 상호작용 개요 다이어그램(interaction overview diagram)

- 활동 다이어그램과 순차 다이어그램을 혼합(접목)한 다이어그램이다.
- 활동 다이어그램에서 객체 사이의 시간흐름을 기지고 있는 부분을 순차 다이어그램을 이용하여 부분적으로 표현한 다이어그램이다.
- 액티비티들이 작은 시퀀스로 분리되어 표현

⑥ 통신 다이어그램(communication diagram)

- 객체 사이의 관계를 중심으로 상호작용을 표현한다.
- 시스템을 구성하는 요소들을 서로 연결하여 시스템 전체의 목적을 구성해 나간다.
- 참여 요소들 간의 메시지 송수신 관계를 파악하기에 좋은 상호작용 다이어그램이다.
- 통신 다이어그램과 순차 다이어그램과 다른 점

통신 다이어그램	통신 다이어그램은 참여 요소들 간의 상호작용 관계에 초점을 둔다.
순차 다이어그램	순차 다이어그램은 메시지들의 흐름과 순서에 초점을 둔다.

⑦ 타이밍 다이어그램(timing diagram)

- 객체 상태 변화와 시간 제약을 명시적으로 표현한다.
- 객체가 한 상태에서 얼마나 오랜 시간을 지체하는지를 명시한다.

기출문제 분석

1. 다음 중 UML 다이어그램이 아닌 것은? [2007년 국가 7급]

- ① 클래스 다이어그램 (class diagram)
- ② 속성 다이어그램 (attribute diagram)
- ③ 사용사례 다이어그램 (use-case diagram)
- ④ 순차 다이어그램 (sequence diagram)

☞ UML(Unified Modeling Language)

-
- 속성 다이어그램 (attribute diagram)(×)
→ 현재, UML(Unified Modeling Language)에 속성 다이어그램은 없다.
 - UML은 소프트웨어 공학에서 사용되는 표준화된 범용 모델링 언어이다.
 - 시스템의 유형과 규모에 따라서 필요한 다이어그램을 선택하여 사용한다.
 - UML은 시스템 개발 활동의 전 과정에서 사용될 수 있다.
 - UML은 객체지향 소프트웨어 개발에서 산출물 명세화, 시각화, 문서화할 때 사용한다.
-

정답 : ②

2. UML은 시스템의 정적인 부분과 동적인 부분을 표현하기 위하여 여러 다이어그램을 제공한다. 정적인 부분을 표현하는 다이어그램들은 시스템의 구조를 나타내기 위해서 사용되며, 동적인 부분을 표현하는 다이어그램들은 시스템의 행위를 나타내기 위해 사용된다. 다음 중 성격이 다른 다이어그램은? [2011년 국가 7급]

- ① 클래스 다이어그램(class diagram)
- ② 협동 다이어그램(collaboration diagram)
- ③ 상태 다이어그램(state diagram)
- ④ 활동 다이어그램(activity diagram)

☞ UML 다이어그램

-
- 클래스 다이어그램은 구조 다이어그램이며 나머지는 행위 다이어그램이다.
 - 구조 다이어그램(structural diagram) : 시스템의 정적인 구조를 나타냄
 - 행위 다이어그램(behavioral diagram) : 시스템을 구성하는 객체들의 동적인 관계
-

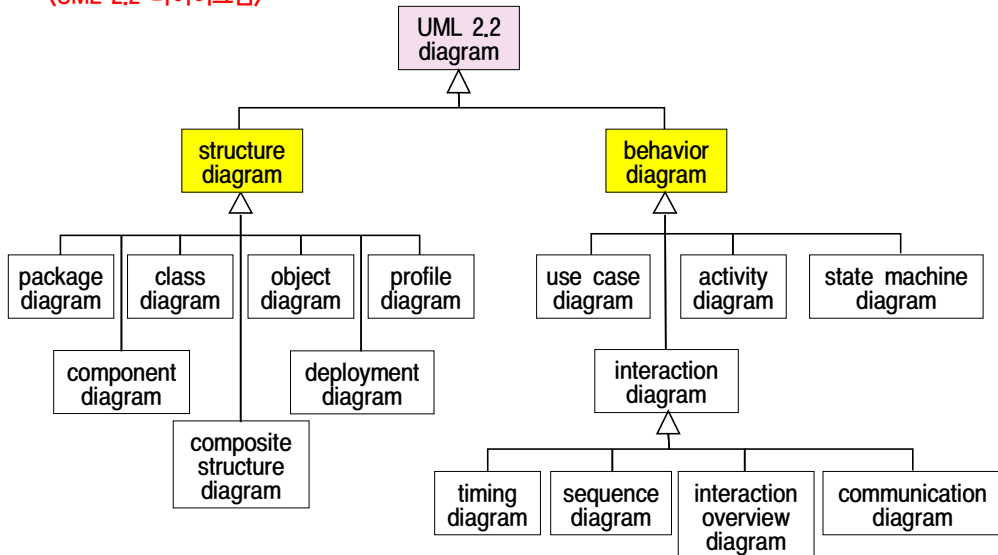
정답 : ①

3. UML 다이어그램 중 행위(behavior) 다이어그램에 해당하지 않는 것은? [2020년 서울 7급]

- ① 활동(activity) 다이어그램
- ② 상태(state) 다이어그램
- ③ 유스케이스(use case) 다이어그램
- ④ 컴포넌트(component) 다이어그램

☞ UML 다이어그램

<UML 2.2 다이어그램>



- 구조 다이어그램(structural diagram) : 시스템의 정적인 구조를 나타냄
- 행위 다이어그램(behavioral diagram) : 시스템을 구성하는 객체들의 동적인 관계

// UML 정리

- UML은 소프트웨어 공학에서 사용되는 표준화된 범용 모델링 언어이다.
- 시스템의 유형과 규모에 따라서 필요한 다이어그램을 선택하여 사용한다.
- UML은 시스템 개발 활동의 전 과정에서 사용될 수 있다.
- UML은 객체지향 소프트웨어 개발에서 산출물 명세화, 시각화, 문서화할 때 사용한다.
- UML은 시스템 개발에 사용되는 **개발 방법론과 무관**하다.
- 하지만, UML은 객체지향 설계 표현 방법의 표준으로 받아들여지고 있다.
- UML은 시스템 개발에 사용되는 CASE 도구에 무관하다.
- UML은 시스템 개발에 사용되는 **프로그래밍 언어에 무관**하다.
- UML로 모델링 한 것은 C++, Java 등의 원시코드로 변환할 수 있다.

4. 다음은 UML2.0 다이어그램에 대한 내용이다. <보기 1>과 <보기 2>를 바르게 연결한 것은?
[2020년 국가 7급]

-----<보기 1>-----

- ㄱ. 시스템의 컴파일 시 계층적구조를 기술하는 다이어그램이다.
- ㄴ. 시퀀스 다이어그램과 액티비티 다이어그램을 혼합한 다이어그램이다.
- ㄷ. 하나의 클래스 실행 시 내부구조를 상세하게 표현하고자 할 때 사용하는 복합구조를 표현하는 다이어그램이다.

-----<보기 2>-----

- A. 인터랙션 오버뷰 다이어그램(interaction overview diagram)
- B. 컴포지트 스트럭처 다이어그램(composite structure diagram)
- C. 패키지 다이어그램(package diagram)

ㄱ	ㄴ	ㄷ
① A	B	C
② C	A	B
③ A	C	B
④ C	B	A

♣ UML2.0 다이어그램

A. 인터랙션 오버뷰 다이어그램(interaction overview diagram)

- 시퀀스 다이어그램과 액티비티 다이어그램을 혼합한 다이어그램이다.
- 여러 상호작용 다이어그램 사이의 제어흐름을 표현한다.
- 상위 수준의 상호작용인 순차, 통신, 타이밍 다이어그램 등으로 표현된 여러 개의 상호작용간의 관계를 표현한 다이어그램이다.

B. 컴포지트 스트럭처 다이어그램(composite structure diagram)

- 하나의 클래스 실행 시 내부구조를 상세하게 표현하고자 할 때 사용하는 복합구조를 표현하는 다이어그램이다.
- 복합체 구조 다이어그램은 UML 2에 새로 포함되었다. 컴포넌트기반개발 지원(CBD)
- 클래스 다이어그램으로만 표현이 가능했던 시스템 내부구조를 보다 상세하게 설계할 수 있다.
- 복합체 구조의 클래스와 컴포넌트 내부구조를 표현한다.
- 복잡한 객체를 분할하여 여러 부분으로 나누어 표현할 수 있다.

C. 패키지 다이어그램(package diagram)

- 시스템의 컴파일 시 계층적구조를 기술하는 다이어그램이다.
- 클래스나 유스케이스 등을 포함한 여러 모델 요소들을 그룹화해 패키지를 구성한다.
- 그리고, 패키지들 사이의 관계를 표현한다.

5. UML 다이어그램의 설명으로 옳지 않은 것은? [2008년 국가 7급]

- ① 사용사례 다이어그램(use-case diagram) : 시스템의 기능을 모델링
- ② 상태 다이어그램(state diagram) : 클래스 사이의 메시지 교환을 시간흐름에 따라 표현
- ③ 클래스 다이어그램(class diagram) : 시스템의 정적인 구조를 나타냄
- ④ 활동 다이어그램(activity diagram) : 시스템의 동적 특징을 나타냄

☞ UML 다이어그램

◆ 상태 다이어그램

- 객체의 상태 변화를 그림으로 나타낸 것이다.(객체의 생명주기를 표현)
- 이벤트에 대한 반응으로 객체의 상태가 어떻게 변화되는지를 나타낸다.
- 여기서, 상태는 객체가 가지는 특정한 값이다.

◆ 순차 다이어그램(sequence diagram)

- 시간흐름에 따른 객체 사이의 상호작용을 표현한다.
 - 객체들 사이의 메시지 교환을 시간흐름을 기준으로 시각화하여 나타낸 것이다.
 - 시스템 실행 시 생성, 소멸되는 객체를 표기하고 객체끼리 주고받는 메시지를 나타낸다.
-

정답 : ②

6. 객체지향 소프트웨어에 대한 설명으로 옳지 않은 것은? [2012년 지방 9급]

- ① 동일한 클래스의 객체들은 서로 다른 클래스 변수 값을 가진다.
- ② 클래스의 객체에 대한 연산을 정의하는 부프로그램을 메서드라 한다.
- ③ 클래스가 하나의 부모클래스를 갖는 것을 단일상속이라 한다.
- ④ UML(Unified Modeling Language)은 객체지향 소프트웨어의 정적모델과 동적모델을 표현할 수 있다.

☞ 객체지향

- 동일한 클래스의 객체들은 서로 다른 클래스 변수 값을 가진다.(x)
→ 동일한 클래스의 객체들은 동일한 클래스의 변수 값을 가진다.
 - A a = new A(); //a는 클래스 A의 객체
-

정답 : ①

7. 프로젝트 관리를 위한 도구 및 기법이 아닌 것은? [2012년 국가 7급]

- ① Use-case Diagram
- ② PERT Chart
- ③ GANTT Chart
- ④ Work Breakdown Structure

☞ 프로젝트 관리를 위한 도구 및 기법

● UML의 Use-case Diagram(사용예 다이어그램)

- 사용예 다이어그램은 시스템의 요구사항을 식별하고, 정의하는데 사용한다.
- 사용예 다이어그램은 시스템 개발에서 분석단계에 해당한다.
- 사용예 다이어그램은 사용자 관점에서 볼 수 있는 시스템의 기능이 무엇인지를 기술한다.
→ 시스템 기능이 어떻게 구현되는지는 기술하지 않는다.

● 프로젝트 관리 정의

- 성공적인 프로젝트 관리에 필수적인 일정, 조직, 지휘, 통제를 제공하는 절차와 기술
- 주어진 기간, 예산, 자원으로 좋은 품질의 소프트웨어 개발을 위한 기술적, 관리적 업무
- 시작과 끝점을 가지며, 최종 산출물을 만들어 내는 작업 계획을 관리하는 것

● 프로젝트 관리를 위한 도구 및 기법

- 작업분해구조(WBS, Work Breakdown Structure)
- PERT/CPM(Program Evaluation and Review Technique/Critical Path Method)
- 간트 차트(GANTT Chart)

● PERT/CPM 전개 과정

- 프로젝트에서 수행되어야 할 모든 작업(activity) 파악
↓
- 작업간의 선행 관계를 결정
↓
- 각 작업간의 선행 관계를 네트워크로 표시
↓
- 작업에 소요되는 시간 추정
↓
- 프로젝트의 최단완료시간과 임계경로 발견

8. 요구사항 모델링 기법과 가장 관련이 적은 것은? [2013년 국가 7급]

- ① UML(Unified Modeling Language)
- ② SDL(Specification and Design Language)
- ③ CPM(Critical Path Method)
- ④ SCR(Software Cost Reduction)

☞ 요구사항 모델링 기법

· CPM은 프로젝트 활동의 계획을 짜기 위한 도구이다.(요구사항 모델링이 아님)

● UML의 유스케이스 다이어그램

- 사용자 요구사항이 제대로 도출 되었는지를 검증하기 위한 용도로 사용
- 개발 시스템의 기능들을 제대로 이해하였는지를 파악하기 위한 용도로 사용
- 유스케이스 다이어그램은 객체지향 기반보다는 기능 중심에 초점을 둔 다이어그램이다.

● SDL(Specification and Design Language) - 설명 및 디자인 언어

- SDL은 시스템을 기술하기 위한 형식 언어이다.
 - SDL 기반으로 시스템의 기능을 분석하고, 설계할 수 있다.
- SDL은 시스템의 상태전이에 대한 이벤트에 초점을 맞추어 요구사항을 분석한다.
- SDL을 이용하여 시스템의 요구사항을 충족하는 모델을 설계할 수 있다.
 - 예 : SDL을 이용하여 프로토콜 요구사항을 충족하는 프로토콜 모델을 설계할 수 있다

● CPM(Critical Path Method) - 임계경로법

- CPM은 프로젝트 활동의 계획을 짜기 위한 도구이다.
- CPM은 효과적인 프로젝트 관리에 중요한 도구이다.

● SCR(Software Cost Reduction) - 소프트웨어 원가 인하

- SCR은 정형 명세(formal specification)에 사용되는 도구이다.
 - 정형 명세는 요구사항 명세 기법으로 요구사항 기술을 수학 및 논리학에 기반을 둔다.
- SCR은 시스템의 요구사항을 좀 더 정형적으로 분석하는데 사용된다.
- 요구사항을 좀 더 정형적으로 분석하기 위해 UML 모델을 SCR 표기법으로 전환 가능

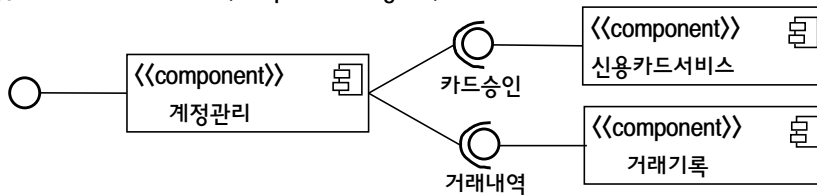
9. UML 다이어그램에 대한 설명으로 옳지 않은 것은? [2021년 국가 7급]

- ① 상태 다이어그램(state diagram)은 객체의 상태 변화를 나타낸다.
- ② 클래스 다이어그램(class diagram)은 클래스와 클래스 사이의 관계를 나타내는 정적인 모델이다.
- ③ 컴포넌트 다이어그램(component diagram)은 물리적인 노드와 커뮤니케이션 경로를 각각 육면체와 선으로 표시한다.
- ④ 활동 다이어그램(activity diagram)은 업무의 흐름을 표현하거나 유스케이스의 구체적인 흐름을 나타내기 위해 사용한다.

☞ UML 다이어그램

- 컴포넌트 다이어그램(component diagram)은 물리적인 노드와 커뮤니케이션 경로를 각각 육면체와 선으로 표시한다.(x) → 배치 다이어그램

// 컴포넌트 다이어그램(component diagram)



- 소프트웨어 컴포넌트 사이의 의존관계를 묘사한다.
- 컴포넌트가 이용하는 인터페이스와 내부 구성요소 사이의 관계를 표현한다.
- 컴포넌트 기반 개발의 가장 큰 장점은 재사용이다.

정답 : ③