



2020년 2회 정보처리기사 실기 시험 100% 합격전략집

1
일차

2
일차

3
일차

4
일차

5
일차

6
일차

7
일차

8
일차

9
일차

10
일차

11
일차

12
일차

13
일차

14
일차

15
일차

16
일차

17
일차

18
일차

19
일차

20
일차

2장 요구사항 확인

핵심 030 현행 시스템 파악 절차

핵심 031 개발 기술 환경

핵심 032 요구사항 유형

핵심 033 요구사항 개발 프로세스

핵심 034 요구사항 분석 기법

핵심 035 요구사항 확인 기법

핵심 036 UML(Unified Modeling
Language)

핵심 037 구조적(Structural) 다이어그램

핵심 038 시퀀스 다이어그램

핵심 039 커뮤니케이션 다이어그램 /
상태 다이어그램

핵심 040 유스케이스 다이어그램

핵심 041 활동 다이어그램



2020년 2회 정보처리기사 실기 대비용 핵심요약

2장 | 요구사항 확인

[핵심 030] 현행 시스템 파악 절차

- 시스템 구성 파악 : 현행 시스템의 구성은 조직의 주요 업무를 담당하는 기간 업무와 이를 지원하는 지원 업무로 구분하여 기술
- 시스템 기능 파악 : 현행 시스템의 기능은 단위 업무 시스템이 현재 제공하는 기능들을 주요 기능과 하부 기능, 세부 기능으로 구분하여 계층형으로 표시
- 시스템 인터페이스 파악 : 현행 시스템의 인터페이스에는 단위 업무 시스템 간에 주고받는 데이터의 종류, 형식, 프로토콜, 연계 유형, 주기 등을 명시
- 아키텍처 구성 파악 : 현행 시스템의 아키텍처 구성은 기간 업무 수행에 어떠한 기술 요소들이 사용되는지 최상위 수준에서 계층별로 표현한 아키텍처 구성도로 작성
- 소프트웨어 구성 파악 : 소프트웨어 구성에는 단위 업무 시스템별로 업무 처리를 위해 설치되어 있는 소프트웨어들의 제품명, 용도, 라이선스 적용 방식, 라이선스 수 등을 명시
- 하드웨어 구성 파악 : 하드웨어 구성에는 단위 업무 시스템들이 운용되는 서버의 주요 사양과 수량, 그리고 이중화의 적용 여부를 명시
- 네트워크 구성 파악 : 네트워크 구성은 업무 시스템들의 네트워크 구성을 파악할 수 있도록 서버의 위치, 서버 간의 네트워크 연결 방식을 네트워크 구성도로 작성

1. 다음 내용 중 괄호에 들어갈 가장 적합한 용어를 쓰시오.

현행 시스템의 ()에는 단위 업무 시스템 간에 주고받는 데이터의 종류, 형식, 프로토콜, 연계 유형, 주기 등을 명시한다. 데이터를 어떤 형식으로 주고받는지, 통신규약은 무엇을 사용하는지, 연계 유형은 무엇인지 등을 반드시 고려해야 한다.

답 :

정답 1. 인터페이스

[핵심 031] 개발 기술 환경

개발하고자 하는 소프트웨어와 관련된 운영체제 (Operating System), 데이터베이스 관리 시스템 (DataBase Management System), 미들웨어(Middle Ware) 등을 선정할 때 고려해야 할 사항을 기술하고, 오픈 소스 사용 시 주의해야 할 내용을 제시한다.

운영체제 (OS, Operating System)	<ul style="list-style-type: none"> • 컴퓨터 시스템의 자원들을 효율적으로 관리하며, 사용자가 컴퓨터를 편리하고 효율적으로 사용할 수 있도록 환경을 제공하는 소프트웨어 • 운영체제 관련 요구사항 식별 시 고려사항 : 가용성, 성능, 기술 지원, 주변 기기, 구축 비용
데이터베이스 관리 시스템 (DBMS)	<ul style="list-style-type: none"> • 사용자와 데이터베이스 사이에서 사용자의 요구에 따라 정보를 생성해 주고, 데이터베이스를 관리해 주는 소프트웨어 • DBMS 관련 요구사항 식별 시 고려사항 : 가용성, 성능, 기술 지원, 상호 호환성, 구축 비용
웹 애플리케이션 서버(WAS; Web Application Server)	<ul style="list-style-type: none"> • 정적인 콘텐츠 처리를 하는 웹 서버와 달리 사용자의 요구에 따라 변하는 동적인 콘텐츠를 처리하기 위해 사용되는 미들웨어 • WAS 관련 요구사항 식별 시 고려사항 : 가용성, 성능, 기술 지원, 구축 비용
오픈 소스 (Open Source)	<ul style="list-style-type: none"> • 누구나 별다른 제한 없이 사용할 수 있도록 소스 코드를 공개한 것으로 오픈 소스 라이선스를 만족하는 소프트웨어 • 오픈 소스 사용에 따른 고려사항 : 라이선스의 종류, 사용자 수, 기술의 지속 가능성

2020년 1, 2회 기사 필기

1. DBMS 관련 요구사항 식별시 고려 사항을 쓰시오.

답 :

2. 웹 애플리케이션 서버(WAS; Web Application Server)의 개념을 쓰시오.

답 :



정답 1. 가용성, 성능, 기술 지원, 상호 호환성, 구축 비용

2. 웹 애플리케이션 서버는 정적인 콘텐츠 처리를 하는 웹 서버와 달리 사용자의 요구에 따라 변하는 동적인 콘텐츠를 처리하기 위해 사용되는 미들웨어이다.

[핵심 032] 요구사항 유형

기능 요구사항	시스템이 갖추어야 할 필수적인 기능에 대한 요구사항
비기능 요구사항	필수 기능 외의 품질이나 제약사항에 관한 요구사항
사용자 요구사항	사용자 관점에서 본 시스템이 제공해야 할 요구사항
시스템 요구사항	개발자 관점에서 본 시스템 전체가 사용자 와 다른 시스템에 제공해야 할 요구사항

1. 소프트웨어가 어떤 문제를 해결하기 위해 제공하는 서비스에 대한 설명과 정상적으로 운영되는데 필요한 제약조건 등을 나타내는 요구사항의 유형 4가지를 쓰시오.

답 :

정답 1. 기능 요구사항, 비기능 요구사항, 사용자 요구사항, 시스템 요구사항

[핵심 033] 요구사항 개발 프로세스

요구사항 도출 (Requirement Elicitation)	<ul style="list-style-type: none"> 시스템, 사용자, 그리고 시스템 개발에 관련된 사람들이 서로 의견을 교환하여 요구사항이 어디에 있는지, 어떻게 수집할 것인지를 식별하고 이해하는 과정 주요 기법 : 인터뷰, 설문, 브레인스토밍, 워크샵, 프로토타이핑, 유스케이스
요구사항 분석 (Requirement Analysis)	<ul style="list-style-type: none"> 개발 대상에 대한 사용자의 요구사항 중 명확하지 않거나 모호하여 이해되지 않는 부분을 발견하고 이를 걸러내기 위한 과정 비용과 일정에 대한 제약 설정, 타당성 조사

요구사항 명세 (Requirement Specification)	요구사항을 체계적으로 분석한 후 승인될 수 있도록 문서화하는 것
요구사항 확인 (Requirement Validation)	개발 자원을 요구사항에 할당하기 전에 요구사항 명세서가 정확하고 완전하게 작성되었는지를 검토하는 활동

1. 요구사항 개발 프로세스 중 시스템, 사용자, 그리고 시스템 개발에 관련된 사람들이 서로 의견을 교환하여 요구사항이 어디에 있는지, 어떻게 수집할 것인지를 식별하고 이해하는 요구사항 개발 프로세스를 쓰시오.

답 :

정답 1. 요구사항 도출(Requirement Elicitation)

[핵심 034] 요구사항 분석 기법

- 개발 대상에 대한 사용자의 요구사항 중 명확하지 않거나 모호한 부분을 걸러내기 위한 방법이다.
- 요구사항 분류(Requirement Classification) : 요구사항을 명확히 확인할 수 있도록 요구사항을 분류함
- 개념 모델링(Conceptual Modeling)
 - 요구사항을 보다 쉽게 이해할 수 있도록 현실 세계의 상황을 단순화하여 개념적으로 표현한 것을 모델이라고 하며, 이러한 모델을 만드는 과정을 모델링이라고 한다.
 - 개념 모델 종류 : 유스케이스 다이어그램, 데이터 흐름 모델, 상태 모델, 목표기반 모델, 사용자 인터랙션, 객체 모델, 데이터 모델 등
- 요구사항 할당(Requirement Allocation) : 요구사항을 만족시키기 위한 구성 요소를 식별하는 것
- 요구사항 협상(Requirement Negotiation) : 요구사항이 서로 충돌될 경우 이를 적절히 해결하는 과정
- 정형 분석(Formal Analysis) : 구문(Syntax)과 의미(Semantics)를 갖는 정형화된 언어를 이용해 요구사항을 수학적 기호로 표현한 후 이를 분석하는 과정

잠깐만요! 자료 흐름도(DFD; Data Flow Diagram)

- 요구사항 분석에서 자료의 흐름 및 변환 과정과 기능을 도형 중심으로 기술하는 방법으로 자료 흐름 그래프, 버블 차트라고도 합니다.
- 자료 흐름도 구성 요소 표기법

프로세스 (Process)	<ul style="list-style-type: none"> 자료를 변환시키는 시스템의 한 부분(처리 과정)을 나타내며 처리, 기능, 변환, 버블이라고 함 원이나 둥근 사각형으로 표시하고 그 안에 프로세스 이름을 기입함
자료 흐름 (Data Flow)	<ul style="list-style-type: none"> 자료의 이동(흐름)이나 연관관계를 나타냄 화살표 위에 자료의 이름을 기입함
자료 저장소 (Data Store)	<ul style="list-style-type: none"> 시스템에서의 자료 저장소(파일, 데이터베이스)를 나타냄 도형 안에 자료 저장소 이름을 기입함
단말 (Terminator)	<ul style="list-style-type: none"> 시스템과 교신하는 외부 개체로, 입력 데이터가 만들어지고 출력 데이터를 받음(정보의 생산자와 소비자) 도형 안에 이름을 기입함

1. 개발 대상에 대한 사용자의 요구사항 중 명확하지 않거나 모호한 부분을 걸러내기 위한 요구사항 분석 기법 중 정형 분석(Formal Analysis)의 개념을 설명하시오.

답 :

2. 유스케이스 다이어그램, 사용자 인터랙션, 데이터 흐름 모델과 관계있는 요구사항 분석 기법을 쓰시오.

답 :

2020년 1, 2회 기사 필기

3. 데이터 흐름도(DFD)의 구성 요소를 쓰시오.

답 :

정답 1. 정형 분석은 구문(Syntax)과 의미(Semantics)를 갖는 정형화된 언어를 이용해 요구사항을 수학적 기호로 표현한 후 이를 분석하는 과정이다.

2. 개념 모델링(Conceptual Modeling)

3. 프로세스(Process), 자료 흐름(Data Flow), 자료 저장소(Data Store), 단말(Terminator)

[핵심 035] 요구사항 확인 기법

- 요구사항 개발 과정을 거쳐 문서화된 요구사항 관련 내용을 확인하고 검증하는 방법이다.
- 요구사항 검토(Requirement Reviews) : 문서화된 요구사항을 훑어보면서 확인하는 것으로 가장 일반적인 요구사항 검증 방법
- 프로토타이핑(Prototyping)
 - 초기 도출된 요구사항을 토대로 프로토타입(Prototype)을 만든 후 대상 시스템의 개발이 진행되는 동안 도출되는 요구사항을 반영하면서 지속적으로 프로토타입을 재작성하는 과정
 - 프로토타입 : 상품이나 서비스가 출시되기 전에 개발 대상 시스템 또는 그 일부분을 개략적으로 만든 원형
- 모델 검증(Model Verification) : 요구사항 분석 단계에서 개발된 모델이 요구사항을 충족시키는지 검증하는 것
- 인수 테스트(Acceptance Test) : 사용자가 실제로 사용될 환경에서 요구사항들이 모두 충족되는지 사용자 입장에서 확인하는 과정

1. 요구사항 개발 과정을 거쳐 문서화된 요구사항 관련 내용을 확인하고 검증하는 방법인 요구사항 확인 기법 중 사용자가 실제로 사용될 환경에서 요구사항들이 모두 충족되는지 사용자 입장에서 확인하는 과정이 무엇인지 쓰시오.

답 :

2. 상품이나 서비스가 출시되기 전에 개발 대상 시스템 또는 그 일부분을 개략적으로 만든 원형을 무엇이라고 하는지 쓰시오.

답 :

정답 1. 인수 테스트(Acceptance Test) 2. 프로토타입(Prototype)



[핵심 036] UML(Unified Modeling Language)

- 시스템 분석, 설계, 구현 등 시스템 개발 과정에서 시스템 개발자와 고객 또는 개발자 상호 간의 의사소통이 원활하게 이루어지도록 표준화한 대표적인 객체지향 모델링 언어이다.
- UML의 구성 요소

사물(Things)	<ul style="list-style-type: none"> • 모델을 구성하는 가장 중요한 기본 요소로, 다이어그램 안에서 관계가 형성될 수 있는 대상들 • 구조(Structural) 사물 : 시스템의 개념적, 물리적 요소를 표현 • 행동(Behavioral) 사물 : 시간과 공간에 따른 요소들의 행위를 표현 • 그룹(Grouping) 사물 : 요소들을 그룹으로 묶어서 표현 • 주해(Annotation) 사물 : 부가적인 설명이나 제약조건 등을 표현
관계(Relationships)	<ul style="list-style-type: none"> • 사물과 사물 사이의 연관성을 표현하는 것 • 연관(Association) 관계 : 2개 이상의 사물이 서로 관련되어 있음을 표현함 • 집합(Aggregation) 관계 : 하나의 사물이 다른 사물에 포함되어 있는 관계를 표현함 • 포함(Composition) 관계 : 집합 관계의 특수한 형태로, 포함하는 사물의 변화가 포함되는 사물에게 영향을 미치는 관계를 표현함 • 일반화(Generalization) 관계 : 하나의 사물이 다른 사물에 비해 더 일반적인지 구체적인지를 표현함 • 의존(Dependency) 관계 : 연관 관계와 같이 사물 사이에 서로 연관은 있으나 필요에 의해 서로에게 영향을 주는 짧은 시간 동안만 연관을 유지하는 관계를 표현함 • 실체화(Realization) 관계 : 사물이 할 수 있거나 해야 하는 기능(행위, 인터페이스)으로 서로를 그룹화 할 수 있는 관계를 표현함

다이어그램 (Diagram)

- 사물과 관계를 도형으로 표현한 것
- 정적 모델링에서는 주로 구조적 다이어그램을 사용하고 동적 모델링에서는 주로 행위 다이어그램을 사용함
 - 정적 모델링 : 사용자가 요구한 기능을 구현하는데 필요한 자료들의 논리적인 구조를 표현한 것
 - 동적 모델링 : 시스템의 내부 구성 요소들의 상태가 시간의 흐름에 따라 변화하는 과정과 변화하는 과정에서 발생하는 상호 작용을 표현한 것

1. UML의 구성 요소 3가지를 쓰시오.

답 :

2. 다음은 UML의 구성 요소들에 대한 설명이다. 괄호(①, ②)에 들어갈 알맞은 구성 요소를 쓰시오.

관계(Relationships)	사물과 사물 사이의 연관성을 표현하는 것이다.
(①)	모델을 구성하는 가장 중요한 기본 요소로, 다이어그램 안에서 관계가 형성될 수 있는 대상들이다.
(②)	사물과 관계를 도형으로 표현한 것이다.

답

① :

② :

3. UML 구성 요소 중 사물과 사물 사이의 연관성을 표현하는 것은 관계이다. 다음은 관계 종류 중 무엇에 대한 설명인지 쓰시오.

사물이 할 수 있거나 해야 하는 기능(행위, 인터페이스)으로 서로를 그룹화 할 수 있는 관계를 표현한다.

답 :

정답 1. 사물(Things), 관계(Relationships), 다이어그램(Diagram)

2. ① 사물(Things) ② 다이어그램(Diagram)

3. 실체화(Realization) 관계



[핵심 037] 구조적(Structural) 다이어그램

- 클래스(Class) 다이어그램
 - 클래스와 클래스가 가지는 속성, 클래스 사이의 관계를 표현한다.
 - UML을 이용한 정적 모델링의 대표적인 것이 클래스 다이어그램이다.
 - 클래스 다이어그램의 구성 요소

클래스(Class)	<ul style="list-style-type: none"> 각각의 객체들이 갖는 속성과 오퍼레이션(동작)을 표현 일반적으로 3개의 구획(Compartment)으로 나눠 클래스의 이름, 속성, 오퍼레이션을 표기함 속성(Attribute) : 클래스의 상태나 정보를 표현 오퍼레이션(Operation, 연산) : 클래스가 수행할 수 있는 동작으로, 함수(메소드, Method)라고도 함
제약조건	속성에 입력될 값에 대한 제약조건이나 오퍼레이션 수행 전후에 지정해야 할 조건이 있다면 이를 적음
관계(Relationships)	클래스와 클래스 사이의 연관성을 표현

- 객체(Object) 다이어그램 : 클래스에 속한 사물(객체)들, 즉 인스턴스(Instance)를 특정 시점의 객체와 객체 사이의 관계로 표현
- 컴포넌트(Component) 다이어그램 : 실제 구현 모듈인 컴포넌트 간의 관계나 컴포넌트 간의 인터페이스를 표현
- 배치(Deployment) 다이어그램 : 결과물, 프로세스, 컴포넌트 등 물리적 요소들의 위치를 표현
- 복합체 구조(Composite Structure) 다이어그램 : 클래스나 컴포넌트가 복합 구조를 갖는 경우 그 내부 구조를 표현
- 패키지(Package) 다이어그램 : 유스케이스나 클래스 등의 모델 요소들을 그룹화한 패키지들의 관계를 표현

2020년 1, 2회 기사 필기

1. Class Diagram, Object Diagram, Component Diagram, Activity Diagram 중 구조적 다이어그램이 아닌 것을 쓰시오.

답 :

2. 다음은 클래스 다이어그램의 클래스에 대한 설명이다. 괄호에 들어갈 가장 적합한 용어를 쓰시오.

클래스 다이어그램의 클래스는 각각의 객체들이 갖는 (①) 과 (②) 을 표현한다. (①) 은 클래스의 상태나 정보를 표현하고, (②) 은 클래스가 수행할 수 있는 동작으로, 함수(메소드, Method)라고도 한다.

답 :

- ① :
② :

정답 1. Activity Diagram

2. ① 속성(Attribute) ② 오퍼레이션(Operation)

[핵심 038] 시퀀스 다이어그램

- 행위(Behavioral) 다이어그램에는 동적 모델링인 시퀀스 다이어그램, 커뮤니케이션 다이어그램, 상태 다이어그램, 기능 모델링인 유스케이스(Use Case) 다이어그램, 활동(Activity) 다이어그램 등이 있다.
- 시퀀스(Sequence) 다이어그램
 - 시스템이나 객체들이 메시지를 주고받으며 시간의 흐름에 따라 상호 작용하는 과정을 그림으로 표현한 것이다.
 - 시퀀스 다이어그램의 구성 요소 : 액터, 객체, 라이프라인, 활성 상자, 메시지, 객체 소멸, 프레임 등

액터(Actor)	시스템으로부터 서비스를 요청하는 외부 요소로, 사람이나 외부 시스템
객체(Object)	<ul style="list-style-type: none"> 메시지를 주고받는 주체 콜론(:)을 기준으로 앞쪽에는 객체명을 뒤쪽에는 클래스명 기술
라이프라인(Lifeline)	객체가 메모리에 존재하는 기간으로, 객체 아래쪽에 점선을 그어 표현



활성 상자 (Activation Box)	객체가 메시지를 주고받으며 구동되고 있음을 라이프라인 상에 겹쳐 직사각형 형태로 표현
메시지(Message)	객체가 상호 작용을 위해 주고받는 메시지
객체 소멸	라이프라인 상에서 객체 소멸 표시를 만나면 해당 객체는 더 이상 메모리에 존재하지 않음을 의미
프레임(Frame)	다이어그램의 전체 또는 일부를 묶어 표현

1. 행위 다이어그램 중 동적 모델링에 해당하는 다이어그램 3 가지를 쓰시오.

답 :

2. 시스템이나 객체들이 메시지를 주고 받으며 상호 작용하는 과정을 표현하는 다이어그램이 무엇인지 쓰시오.

답 :

정답 1. 시퀀스 다이어그램(Sequence Diagram), 커뮤니케이션 다이어그램(Communication Diagram), 상태 다이어그램(State Diagram)

2. 시퀀스 다이어그램(Sequence Diagram)

[핵심 039] 커뮤니케이션 다이어그램 / 상태 다이어그램

커뮤니케이션(Communication) 다이어그램

- 시퀀스 다이어그램과 같이 동작에 참여하는 객체들이 주고받는 메시지를 표현하는데, 메시지뿐만 아니라 객체들 간의 연관까지 표현한다.
- 커뮤니케이션 다이어그램의 구성 요소 : 액터, 객체, 링크, 메시지 등

링크(Link)	<ul style="list-style-type: none"> • 객체들 간의 관계를 표현하는 데 사용 • 액터와 객체, 객체와 객체 간에 실선을 그어 표현
----------	---

상태(State) 다이어그램

- 하나의 객체가 자신이 속한 클래스의 상태 변화 혹은 다른 객체와의 상호 작용에 따라 상태가 어떻게 변화하는지를 표현한다.

- 상태 다이어그램의 구성 요소 : 상태, 이벤트, 상태 전환 등

상태(State)	<ul style="list-style-type: none"> • 객체의 상태를 둥근 시간형 안에 기술 • 시작 상태 : 상태의 시작을 속이 채워진 원(●)으로 표현 • 종료 상태 : 상태의 종료를 속이 채워진 원을 둘러싼 원(◎)으로 표현
이벤트(Event)	조건, 외부 신호, 시간의 흐름 등 상태에 변화를 주는 현상
상태 전환	상태 사이의 흐름, 변화를 화살표로 표현
프레임(Frame)	상태 다이어그램의 범위를 표현

1. 동작에 참여하는 객체들이 주고받는 메시지뿐만 아니라 객체들 간의 연관까지 표현하는 다이어그램을 쓰시오.

답 :

정답 1. 커뮤니케이션 다이어그램(Communication Diagram)

[핵심 040] 유스케이스 다이어그램

- 개발될 시스템과 관련된 외부 요소들, 즉 사용자와 다른 외부 시스템들이 개발될 시스템을 이용해 수행할 수 있는 기능을 사용자의 관점(View)에서 표현한 것이다.
- 유스케이스 다이어그램의 구성 요소 : 시스템 범위, 액터, 유스케이스, 관계

시스템 범위 (System Scope)	시스템 내부에서 수행되는 기능들을 외부 시스템과 구분하기 위해 시스템 내부의 유스케이스들을 사각형으로 묶어 시스템의 범위를 표현
액터(Actor)	시스템과 상호작용을 하는 모든 외부 요소로, 사람이나 외부 시스템을 의미
유스케이스 (Use Case)	사용자가 보는 관점에서 시스템이 액터에게 제공하는 서비스 또는 기능을 표현한 것
관계 (Relationship)	유스케이스 다이어그램에서 관계는 액터와 유스케이스, 유스케이스와 유스케이스 사이에서 나타날 수 있으며, 포함 관계, 확장 관계, 일반화 관계의 3종류가 있음



1. 유스케이스 다이어그램의 구성 요소 4가지를 쓰시오.

답 :

정답 1. 시스템 범위(System Scope), 액터(Actor), 유스케이스(Use Case), 관계(Relationship)

[핵심 041] 활동 다이어그램

- 자료 흐름도와 유사한 것으로, 사용자의 관점(View)에서 시스템이 수행하는 기능을 처리 흐름에 따라 순서대로 표현한 것이다.
- 활동 다이어그램의 구성 요소 : 액션, 액티비티, 노드, 스윘레인 등

액션(Action)/ 액티비티(Activity)	<ul style="list-style-type: none"> • 액션(Action)은 더 이상 분해할 수 없는 단일 작업 • 액티비티(Activity)는 몇 개의 액션으로 분리될 수 있는 작업
노드	<ul style="list-style-type: none"> • 시작 노드 : 액션이나 액티비티가 시작됨을 의미 • 종료 노드 : 액티비티 안의 모든 흐름이 종료됨을 의미 • 조건(판단) 노드 : 조건에 따라 제어의 흐름이 분리됨을 표현 • 병합 노드 : 여러 경로의 흐름이 하나로 합쳐짐을 표현 • 포크(Fork) 노드 : 액티비티의 흐름이 분리되어 수행됨을 표현 • 조인(Join) 노드 : 분리되어 수행되던 액티비티의 흐름이 다시 합쳐짐을 표현
스윘레인 (Swim Lane)	액티비티 수행을 담당하는 주체를 구분

1. 시스템이 어떤 기능을 수행하는지 객체의 처리 로직이나 조건에 따른 처리의 흐름을 순서에 따라 표현하는 다이어그램이 무엇인지 쓰시오.

답 :

정답 1. 활동 다이어그램(Activity Diagram)

