

# 1과목 : 소프트웨어 설계

## Chapter 1: 요구사항 확인

### Section 1: 소프트웨어 생명주기

Q. 애자일 방법론에 해당하지 않는 것은?

1. 기능 중심 개발
2. 스크럼
3. 익스트림 프로그래밍
4. 모듈 중심 개발

정답 :

해설 :

애자일 모형

‘민첩한’, ‘기민한’ 이라는 의미로 고객의 요구사항 변화에 유연하게 대응할 수 있도록  
일정한 주기를 반복하면서 개발과정을 진행하는 개발 방법론

애자일 모형

스크럼(Scrum)

XP(eXtreme Programming)

칸반(Kanban)

Lean

크리스탈(Crystal)

기능중심개발(FDD: Feature Driven Development)

DSDM(Dynamic System Development Method)

DAD(Disciplined Agile Delivery)

### Section 2: 스크럼(Scrum) 기법

Q. 다음의 스크럼(Scrum) 개발 과정을 진행 순서에 맞게 올바르게 나열한 것은?

- ㄱ. 스프린트(sprint)
- ㄴ. 스프린트 회고(Spring Retrospective)
- ㄷ. 일일 스크럼 회의(Daily Scrum Meet)
- ㄹ. 스프린트 검토 회의
- ㅁ. 스프린트 계획 회의

1. ㄱ - ㄷ - ㄱ - ㄴ - ㄹ
2. ㄱ - ㄱ - ㄷ - ㄹ - ㄴ
3. ㄱ - ㄷ - ㄱ - ㄹ - ㄴ
4. ㄱ - ㄹ - ㄱ - ㄴ - ㄷ

정답 :

해설 : 계획한 내용을 토대로 일정 기간 동안 스프린트를 수행하면서 진행 상황을 매일 점검하고 하나의 스프린트가 끝나면 검토한 후 진행을 되돌아본다.

## Section 3: XP(eXtreme Programming) 기법

Q. XP(extreme Programming)의 기본 원리로 볼 수 없는 것은?

1. Linear Sequential Method
2. Pair Programming
3. Collective Ownership
4. Continuous Integration

정답 :

해설 : XP의 5가지 핵심가치(의사소통, 단순성, 용기, 존중, 피드백),  
XP의 주요 실천방법 - 짝 프로그래밍, 공동 코드 소유, 테스트 주도 개발  
전체 팀, 지속적인 통합, 리팩토링, 소규모 릴리즈 등  
문제의 Linear Sequential Method: 순차적 방법으로 기본 원리가 아니다  
Pair Programming: 개발 코드에 대한 권한과 책임을 공동으로 소유  
Collective Ownership: 소스에 대한 팀의 공통책임, 누구든 수정 가능함  
Continuous Integration: 컴포넌트 또는 모듈 단위로 지속적으로 통합,

## Section 9: UML(Unified Modeling Language)

Q. 럼바우(Rumbaugh) 객체지향 분석 기법에서 동적 모델링에 활용되는 다이어그램은?

1. 객체 다이어그램(Object Diagram)
2. 패키지 다이어그램(Package Diagram)
3. 상태 다이어그램(State Diagram)
4. 자료 흐름도(Data Flow Diagram)

정답 :

해설 : 럼바우 객체지향 분석 기법과 관련된 다이어그램 두가지

- 정적모델링에 활용되는 구조적 다이어그램 : 객체 다이어그램(Object Diagram)
- 동적 모델링에 활용되는 행위 다이어그램 : 상태(State Diagram)

Q. UML(Unified Modeling Language)에 대한 설명 중 틀린 것은?

1. 기능적 모델은 사용자 측면에서 본 시스템 기능이며 UML에서는 Use case Diagram을 사용한다.

2. 정적 모델은 객체, 속성, 연관관계, 오퍼레이션의 시스템 구조를 나타내며, UML에서는 **Class Diagram**을 사용한다.
3. 동적 모델은 시스템의 내부 동작을 말하며, UML에서는 **Sequence D**다.
4. 정적 모델은 객체, 속성, 연관관계, 오퍼레이션의 시스템 구조를 나타내며, UML에서는 **Class Diagram**을 사용한다.

## Section 10: 주요 UML 다이어그램

**Q. UML에서 시퀀스 다이어그램의 구성 항목에 해당하지 않는 것은?**

1. 생명선
2. 실행
3. 확장
4. 메시지

정답 :

해설 : 시퀀스 다이어그램(순차 다이어그램) - 시간의 흐름에 따른 상호작용 과정을 표현

구성요소 - 액터(Actor) : 서비스를 요청하는 외부요소, 사람이나 외부 시스템

객체(object) : 메시지를 주고받는 주체

생명선(Lifeline) : 객체가 메모리에 존재하는 기간, 객체 아래쪽에 점선을 그어 표현

실행 상자(Active Box): 객체가 메시지를 주고받으며 구동되고 있음을 표현함

메시지(Message): 객체가 상호 작용을 위해 주고받는 메시지

## Chapter 2: 화면 설계

### Section 13: UI 설계 도구

**Q. 다음 내용이 설명하는 UI 설계 도구는?**

- 디자인, 사용 방법 설명, 평가 등을 위해 실제 화면과 유사하게 만든 정적인 형태의 모형
- 시각적으로만 구성 요소를 배치하는 것으로 일반적으로 실제로 구현되지는 않음

1. 스토리보드(Storyboard)
2. 목업(Mockup)
3. 프로토타입(Prototype)
4. 유스케이스(Usecase)

정답 :

해설 : 핵심은 “시각적으로만 배치하는 것으로 실제 기능은 구현되지 않는다”는 것

- 와이어프레임 : 기획 초기에 제작하는 것으로 페이지에 대한 개략적인 레이아웃, UI 요소 등에 대한 뼈대를 설계하는 단계
- 스토리보드 : 와이어 프레임에 콘텐츠에 대한 설명, 페이지 간 이동 흐름 등을 추가한 문서
- 목업: 디자인, 사용방법 설명, 평가 등을 위해 와이어프레임보다 좀 더 실제화면과 유사하게 만든 정적인 형태의 모형
- 프로토타입 : 와이어프레임이나 스토리보드 등에 “인터랙션”을 적용함으로써 실제 구현된 것처럼 테스트 가능한 동적인 형태의 모형이다.
- 유스케이스 : 사용자 측면에서의 요구사항으로, 사용자가 원하는 목표를 달성하기 위해 수행할 내용을 기술한다.

## Section 15: 품질 요구 사항

**Q. ISO/IEC 9126의 소프트웨어 품질 특성 중 기능성(Functionality)의 하위 특성으로 옳지 않은 것은?**

1. 학습성
2. 적합성
3. 정확성
4. 보안성

정답 :

해설 : 기능성의 하위 특성에는 적합성, 정확성, 상호 운용성, 보안성, 준수성 등이 있다.

**Q. 패키지 소프트웨어의 일반적인 제품 품질 요구사항 및 테스트를 위한 국제 표준은?**

1. ISO/IEC 2192
2. IEEE 19554
3. ISO/IEC 12119
4. ISO/IEC 14959

정답 :

해설 : “테스트 절차”가 포함된 국제 표준은 ISO/IEC 12119이다.

**Q.** 소프트웨어 품질 목표중 하나 이상의 하드웨어 환경에서 운용되기 위해 쉽게 수정될 수 있는 시스템 능력을 의미하는 것은?

1. Portability
2. Efficiency
3. Usability
4. Correctness

정답 :

해설 : Portability - 이식성, Efficiency - 효율성, Usability - 유용성, Correctness - 정확성

## Chapter 3: 애플리케이션 설계

### Section 21: 소프트웨어 아키텍처

**Q.** 다음 ( ) 안에 들어갈 내용으로 옳은 것은?

컴포넌트 설계 시 “( )에 의한 설계”를 따를 경우 해당 명세서에는

- (1) 컴포넌트의 오퍼레이션 사용전에 참이 되어야할 선행조건
- (2) 사용 후 만족 되어야 할 결과조건
- (3) 오퍼레이션이 실행되는 동안 항상 만족되어야 할 불변조건 등이 포함되어야 한다.

1. 협약(Contract)
2. 프로토콜(Protocol)
3. 패턴(Pattern)
4. 관계(Relation)

정답 :

해설 : 협약(Contract)에 의한 설계 - 컴포넌트를 설계할 때 클래스에 대한 여러 가정을 공유할 수 있도록 명세한 것으로, 소프트웨어 컴포넌트에 대한 정확한 인터페이스를 명세한다.  
협약에 의한 설계 시 명세에 포함될 조건에는 선행 조건, 결과 조건, 불변 조건이 있다.

**Q.** 소프트웨어 아키텍처 설계에서 시스템 품질 속성이 아닌 것은?

1. 가용성(Availability)
2. 독립성(Isolation)

3. 변경 용이성(Modifiability)

4. 사용성(Usability)

정답 :

해설 : 시스템 품질 속성에는 성능, 변경 용이성, 사용성, 기능성, 가용성, 확장성, 보안성 등이 있다.

**Q. 아키텍처 설계 과정이 올바른 순서로 나열된 것은?**

(가) 설계 목표 설정

(나) 시스템 타입 결정

(다) 스타일 적용 및 커스터마이징

(라) 서브시스템의 기능, 인터페이스 동작 작성

(마) 아키텍처 설계 검토

1. (가) - (나) - (다) - (라) - (마)

2. (마) - (가) - (나) - (라) - (다)

3. (가) - (마) - (나) - (라) - (다)

4. (가) - (나) - (다) - (마) - (라)

정답 :

해설 : 먼저 **목표**를 설정하고 **타입**을 결정한 후 **스타일(패턴)**을 적용한다. 이어서 **서브시스템**을 구체화하고 아키텍처를 최종 **검토**한다.

## Section 22: 아키텍처 패턴

**Q. 서브시스템이 입력 데이터를 받아 처리하고 결과를 다른 시스템에 보내는 작업이 반복되는**

**아키텍처 스타일은?**

1. 클라이언트 서버 구조

2. 계층 구조

3. MVC 구조

4. 파이프 필터 구조

정답 :

해설 : 시스템이 파이프처럼 연결되어 있어서 앞 시스템의 처리 결과물을 파이프를 통해 전달받아 처리한 후 다음 시스템으로 넘겨주는 패턴을 반복하는 아키텍처 스타일을 파이프-필터 구조라고한다.

**Q.** 네트워크 프로토콜의 **OSI** 참조 모델과 가장 관련이 깊은 아키텍처 모델은?

1. Peer-To-Peer Model
2. Mvc Model
3. Layers Model
4. Client-Server Model

정답 :

해설 : OSI 참조 모델은 네트워크 프로토콜을 계층(Layer)별로 구분한 모델이다.

## Section 24: 객체지향 분석 및 설계

**Q.** 객체지향 분석 방법론 중 **Coad-Yourdon** 방법에 해당하는 것은?

1. E-R 다이어그램을 사용하여 객체의 행위를 데이터 모델링하는데 초점을 둔 방법이다.
2. 객체, 동적, 기능 모델로 나누어 수행하는 방법이다.
3. 미시적 개발 프로세스와 거시적 개발 프로세스를 모두 사용하는 방법이다.
4. Use-Case를 강조하여 사용하는 방법이다.

정답 :

해설 : E-R 다이어그램은 Coad-Yourdon, 미시적과 거시적은 Booch,  
Use-Case 는 Jacobson, 객체-동적-기능 모델은 Rumbaugh

## Section 25: 모듈

**Q.** 다음 중 가장 강한 응집도(Cohesion)는?

1. Sequential Cohesion
2. Procedural Cohesion
3. Logical Cohesion
4. Coincidental Cohesion

정답 :

해설 : 응집도가 강한것부터(기능적 - 순차적 - 교환(통환)적 - 절차적 - 시간적 - 논리적 - 우연적)  
Functional - Sequential - Communication - Procedural - Temporal - Logical

결합도의 경우 약한것부터(자료-스탬프(검인) - 제어 - 외부 - 공통(공유) - 내용)

Data - Stamp - Control - External - Common - Content

## Section 28: 디자인 패턴

**Q. GoF(Gang of Four)** 디자인 패턴을 생성, 구조, 행동 패턴의 세그룹으로 분류할 때,

구조 패턴이 아닌 것은?

1. Adapter 패턴
2. Bridge 패턴
3. Builder 패턴
4. Proxy 패턴

정답 :

해설 : Builder는 '생성' 패턴이다.

## Chapter 4: 인터페이스 설계

### Section