**문제1**

**문제**: 어떤 시스템의 설계 단계에서 초기 사용자 요구를 시각화하고 빠른 피드백을 얻기 위해 화면 정의 작업을 수행하고자 한다. 이 때 가장 적절한 도구 또는 산출물을 고르시오.  
**보기**:  
① UI 프로토타입  
② 와이어프레임  
③ 화면 흐름도  
④ 반응형 디자인  
⑤ 표준 컴포넌트  
**정답**: ①  
**해설**: 초기 사용자 피드백을 얻기 위한 시각화 도구로는 UI 프로토타입이 가장 적절하다.

**문제2**

**문제**: 스크럼 개발 방법론에서 매일 팀이 모여 작업 진행 상황을 공유하고, 문제를 논의하는 회의는 무엇인가?  
**보기**:  
① 스프린트 회고  
② 제품 백로그 회의  
③ 데일리 스크럼  
④ 스프린트 계획  
⑤ 리파인먼트 미팅  
**정답**: ③  
**해설**: 데일리 스크럼은 스크럼 팀이 매일 모여 진행 상황을 공유하고 문제를 논의하는 짧은 회의이다.

**문제3**

**문제**: 시스템 간 연계 인터페이스를 설계할 때, 비동기 방식으로 메시지를 전달하고 수신을 보장하기 위해 사용하는 시스템은?  
**보기**:  
① API Gateway  
② 메시지 브로커  
③ Web Server  
④ File Server  
⑤ Load Balancer  
**정답**: ②  
**해설**: 메시지 브로커는 비동기 통신을 위한 시스템으로, 송신자와 수신자가 동시에 연결되어 있지 않아도 메시지 전달을 보장한다.

**문제4**

**문제**: 소프트웨어 형상관리에서 변경 요청에 대해 승인하거나 거절하는 기능은 무엇인가?  
**보기**:  
① 형상 감사  
② 형상 식별  
③ 형상 통제  
④ 형상 보고  
⑤ 형상 계획  
**정답**: ③  
**해설**: 형상 통제는 형상 항목에 대한 변경 요청을 검토하고 승인 또는 거절하는 활동이다.

**문제5**

**문제**: 사용자가 메뉴를 클릭했을 때 시스템이 응답 중임을 보여주는 UI 원칙은?  
**보기**:  
① 일관성  
② 직관성  
③ 가시성  
④ 피드백  
⑤ 유연성  
**정답**: ④  
**해설**: 피드백은 사용자의 행위에 대해 시스템이 상태나 결과를 즉각적으로 알려주는 것을 의미한다.

**문제6**

**문제**: 소프트웨어 생명주기(SDLC)의 단계 중 구현 단계 직전에 수행되어야 하는 활동은?  
**보기**:  
① 요구사항 정의  
② 테스트 계획 수립  
③ 설계  
④ 유지보수  
⑤ 배포  
**정답**: ③  
**해설**: 소프트웨어 구현은 설계 후에 수행되며, 설계는 요구사항을 기반으로 시스템 구조를 정의하는 단계이다.

**문제7**

**문제**: REST API를 설계할 때 가장 먼저 고려해야 하는 요소는 무엇인가?  
**보기**:  
① 요청/응답 포맷  
② API 인증 방식  
③ 리소스 URI 설계  
④ HTTP 상태코드  
⑤ 응답 시간  
**정답**: ③  
**해설**: RESTful 설계에서 가장 먼저 고려해야 하는 것은 URI를 리소스 중심으로 설계하는 것이다.

**문제8**

**문제**: API 인증에서 OAuth2 방식은 어떤 방식으로 접근 토큰을 부여하는가?  
**보기**:  
① 사용자 직접 인증  
② API Key 발급  
③ 권한 위임을 통한 인증 서버 발급  
④ 세션 쿠키 부여  
⑤ 보안코드 직접 삽입  
**정답**: ③  
**해설**: OAuth2는 권한 위임 방식으로 인증 서버가 접근 토큰을 발급해주는 구조이다.

**문제9**

**문제**: 업무 프로세스를 분석할 때 현재 수행 중인 프로세스를 도식화하는 기법은?  
**보기**:  
① TO-BE 모델링  
② 시나리오 기법  
③ 유스케이스 다이어그램  
④ AS-IS 모델링  
⑤ 상태 다이어그램  
**정답**: ④  
**해설**: AS-IS 모델링은 현재 상태의 업무 프로세스를 분석하고 시각화하는 데 사용된다.

**문제10**

**문제**: 로그인 기능 테스트에서 사용자 입력 없이 로그인 버튼을 클릭했을 때 오류 메시지가 출력되는지 확인하는 테스트 유형은?  
**보기**:  
① 경계값 분석  
② 정상 시나리오  
③ 예외 시나리오  
④ 통합 테스트  
⑤ 회귀 테스트  
**정답**: ③  
**해설**: 예외 시나리오는 사용자 오류 또는 예외 조건에 대한 시스템의 처리 여부를 검증하는 테스트 방식이다.

**문제11**

**문제**: 사용자의 접근 빈도가 높은 데이터를 빠르게 제공하기 위해 캐싱을 도입하고자 한다. 이 때 가장 적절한 캐시 전략은?  
**보기**:  
① LRU 캐시  
② FIFO 캐시  
③ LFU 캐시  
④ Random 캐시  
⑤ ARC 캐시  
**정답**: ①  
**해설**: LRU(Least Recently Used)는 가장 오랫동안 사용되지 않은 데이터를 제거하는 전략으로 일반적인 캐시에서 효과적이다.

**문제12**

**문제**: 마이크로서비스 아키텍처(MSA)에서 서비스 간 독립성을 유지하기 위해 데이터베이스를 어떻게 구성하는 것이 가장 적절한가?  
**보기**:  
① 단일 통합 DB 사용  
② 모놀리식 DB 분할  
③ 서비스별 독립 DB 구성  
④ 클라우드 공유 DB 사용  
⑤ 읽기 전용 DB 사용  
**정답**: ③  
**해설**: MSA에서는 각 서비스가 독립적으로 배포 및 운영될 수 있도록 자체 DB를 갖도록 설계한다.

**문제13**

**문제**: 요구사항 명세 시 고객의 니즈를 정확히 반영하기 위한 활동으로 가장 적절한 것은?  
**보기**:  
① 기능 설계  
② 요구사항 인터뷰  
③ 테스트 시나리오 작성  
④ 모듈 분해  
⑤ 성능 측정  
**정답**: ②  
**해설**: 요구사항 인터뷰는 고객과 직접 커뮤니케이션을 통해 명확한 요구사항을 도출하는 기법이다.

**문제14**

**문제**: 소프트웨어 테스트 단계에서 결함 수정 후 기존 기능이 정상적으로 동작하는지 확인하는 테스트는?  
**보기**:  
① 회귀 테스트  
② 단위 테스트  
③ 통합 테스트  
④ 부하 테스트  
⑤ 탐색 테스트  
**정답**: ①  
**해설**: 회귀 테스트는 버그 수정 등 변경 이후 기존 기능의 이상 유무를 확인하기 위한 테스트이다.

**문제15**

**문제**: 시스템의 무결성과 일관성을 유지하기 위해 변경 이력과 접근을 통제하는 활동은?  
**보기**:  
① 보안 정책 설정  
② 형상관리  
③ 변경 요청  
④ 로깅 시스템 도입  
⑤ 감사 로그 검토  
**정답**: ②  
**해설**: 형상관리는 소프트웨어 구성요소의 변경 이력을 추적하고 통제하는 프로세스를 포함한다.

**문제16**

**문제**: 테스트 케이스 설계 시 정상적인 입력만을 고려하는 방식은?  
**보기**:  
① 경계값 분석  
② 예외 시나리오 설계  
③ 기능 기반 설계  
④ 정상 시나리오 설계  
⑤ 오류 추정  
**정답**: ④  
**해설**: 정상 시나리오는 일반적인 상황에서 예상되는 입력을 기준으로 테스트를 설계한다.

**문제17**

**문제**: 웹 시스템 설계 시, 다양한 해상도와 디바이스에서도 일관된 화면을 제공하기 위한 접근 방식은?  
**보기**:  
① 서버 사이드 렌더링  
② 정적 페이지  
③ 반응형 웹 디자인  
④ 하이브리드 앱  
⑤ 데스크톱 전용 설계  
**정답**: ③  
**해설**: 반응형 웹 디자인은 다양한 화면 크기에 맞춰 자동으로 레이아웃이 조정되도록 설계된다.

**문제18**

**문제**: 시스템 간 실시간 데이터 연계를 위한 가장 적절한 통신 방식은?  
**보기**:  
① 일괄 처리  
② FTP  
③ 파일 공유  
④ REST API  
⑤ 테이블 복제  
**정답**: ④  
**해설**: REST API는 경량의 HTTP 기반 통신으로 실시간 연계에 적합하다.

**문제19**

**문제**: 애플리케이션 배포 자동화를 위해 Jenkins와 같은 도구를 사용하는 이유는?  
**보기**:  
① 데이터 분석  
② 코드 컴파일  
③ 빌드 자동화 및 배포  
④ 모니터링  
⑤ 트래픽 분석  
**정답**: ③  
**해설**: CI/CD 도구는 코드 빌드부터 테스트, 배포까지 자동화된 파이프라인을 제공한다.

**문제20**

**문제**: 다음 중 UI 설계 시 직관성을 높이기 위한 가장 적절한 방법은?  
**보기**:  
① 색상 최소화  
② 사용자 흐름 단순화  
③ 기능 숨기기  
④ 복잡한 버튼 배치  
⑤ 상호작용 숨김  
**정답**: ②  
**해설**: 사용자 흐름이 단순할수록 사용자는 빠르게 UI를 이해하고 이용할 수 있다.

**문제21**

**문제**: 대규모 분산 시스템에서 장애 복구 시, 서비스 중단을 최소화하기 위해 설계에 포함해야 할 전략은?  
**보기**:  
① 단일 서버 아키텍처  
② 수동 백업 시스템  
③ 이중화 및 페일오버 구성  
④ 로컬 로그 저장  
⑤ 암호화 적용  
**정답**: ③  
**해설**: 이중화 및 페일오버는 장애 발생 시 다른 노드로 자동 전환되어 서비스 연속성을 보장한다.

**문제22**

**문제**: UI 요소가 서로 다른 기능을 수행하지만 동일한 색상이나 아이콘을 사용하는 경우 발생할 수 있는 문제는?  
**보기**:  
① 접근성 향상  
② 피드백 강화  
③ 혼란 유발  
④ 보안성 저하  
⑤ 일관성 향상  
**정답**: ③  
**해설**: 시각적으로 동일한 요소가 다른 기능을 수행하면 사용자의 혼란을 초래한다.

**문제23**

**문제**: 비기능 요구사항 중 시스템이 동시에 처리할 수 있는 사용자 수와 관련된 항목은?  
**보기**:  
① 보안성  
② 신뢰성  
③ 확장성  
④ 응답 시간  
⑤ 동시 처리성  
**정답**: ⑤  
**해설**: 동시 처리성은 시스템이 동시 사용자 요청을 얼마나 감당할 수 있는지를 의미한다.

**문제24**

**문제**: 업무 프로세스 개선을 위해 현재 프로세스를 분석한 후 미래 프로세스를 설계하는 접근 방법은?  
**보기**:  
① BPMN 작성  
② AS-IS 모델링  
③ TO-BE 모델링  
④ 워크플로우 분석  
⑤ 요구사항 수집  
**정답**: ③  
**해설**: TO-BE 모델링은 향후 도입될 개선된 프로세스를 정의하는 기법이다.

**문제25**

**문제**: MSA 환경에서 서비스 간 통신 시, 메시지 구조를 정의하고 공유하는 데 사용하는 형식은?  
**보기**:  
① YAML  
② Swagger  
③ JSON Schema  
④ Markdown  
⑤ PDF 문서  
**정답**: ③  
**해설**: JSON Schema는 API 요청 및 응답 메시지 구조를 명확히 정의하는 데 사용된다.

**문제26**

**문제**: UI 테스트에서 실제 사용자 흐름을 자동화하여 검증하는 테스트 기법은?  
**보기**:  
① 단위 테스트  
② 통합 테스트  
③ 회귀 테스트  
④ 시나리오 테스트  
⑤ 코드 리뷰  
**정답**: ④  
**해설**: 시나리오 테스트는 실제 사용자 행동을 따라가며 UI 동작을 검증한다.

**문제27**

**문제**: 운영 환경에 배포된 시스템의 이상 동작을 실시간으로 감지하고자 한다. 이 때 가장 적절한 방법은?  
**보기**:  
① 수동 점검  
② 주기적 백업  
③ 로그 분석 도구 활용  
④ 테스트 코드 작성  
⑤ 형상관리  
**정답**: ③  
**해설**: 로그 분석 도구는 실시간 모니터링과 경고 시스템을 통해 이상 동작을 감지한다.

**문제28**

**문제**: REST API에서 리소스를 식별하는 가장 기본적인 방식은?  
**보기**:  
① 쿼리 파라미터  
② 세션 토큰  
③ 쿠키 값  
④ URI 경로  
⑤ 응답 헤더  
**정답**: ④  
**해설**: REST 설계의 핵심은 리소스를 URI로 식별하는 것이다.

**문제29**

**문제**: 스프린트 종료 후 팀이 개선점을 도출하고 다음 작업에 반영하기 위해 수행하는 회의는?  
**보기**:  
① 스프린트 계획 회의  
② 데일리 스크럼  
③ 스프린트 리뷰  
④ 스프린트 회고  
⑤ 리파인먼트  
**정답**: ④  
**해설**: 스프린트 회고는 프로세스 개선을 위한 내부 피드백 회의이다.

**문제30**

**문제**: 백엔드 서버가 동시에 처리할 수 있는 요청 수를 증가시키기 위해 가장 먼저 고려해야 할 요소는?  
**보기**:  
① UI 색상 조정  
② 데이터베이스 인덱스  
③ 로깅 정책  
④ 스레드/이벤트 루프 모델  
⑤ REST 응답 포맷  
**정답**: ④  
**해설**: 서버는 스레드 기반 또는 이벤트 루프 방식으로 병렬 처리를 하며 성능에 직접적인 영향을 준다.

**문제31**

**문제**: 대용량 트래픽을 효율적으로 처리하기 위해 서버 앞단에 위치하여 요청을 분산 처리하는 구성 요소는?  
**보기**:  
① 메시지 큐  
② 데이터 웨어하우스  
③ API 게이트웨이  
④ 로드 밸런서  
⑤ 프록시 서버  
**정답**: ④  
**해설**: 로드 밸런서는 요청을 여러 서버에 분산시켜 부하를 분산하고 가용성을 높이는 역할을 한다.

**문제32**

**문제**: UI 프로토타입을 설계할 때 가장 중요한 고려 요소는?  
**보기**:  
① 정확한 코드 구현  
② 미적인 요소  
③ 사용자의 니즈 반영  
④ 데이터베이스 구조  
⑤ 브라우저 호환성  
**정답**: ③  
**해설**: UI 프로토타입은 사용자의 니즈와 기대를 빠르게 반영하고 피드백을 받기 위한 도구이다.

**문제33**

**문제**: 다음 중 인터페이스 정의서에 포함되어야 할 핵심 항목이 아닌 것은?  
**보기**:  
① 인터페이스 식별자  
② 전송 방식  
③ 오류 처리 방법  
④ 데이터베이스 인덱스  
⑤ 데이터 포맷  
**정답**: ④  
**해설**: 인터페이스 정의서는 시스템 간 통신 사양을 정의하며, DB 인덱스는 포함 대상이 아니다.

**문제34**

**문제**: 단위 테스트에서 주로 사용되는 테스트 대상은 무엇인가?  
**보기**:  
① 전체 시스템  
② 사용자 시나리오  
③ 개별 모듈/함수  
④ 통신 네트워크  
⑤ UI 화면 구성  
**정답**: ③  
**해설**: 단위 테스트는 함수나 메서드와 같은 최소 코드 단위를 대상으로 수행한다.

**문제35**

**문제**: 다음 중 마이크로서비스 아키텍처에서 발생할 수 있는 단점은?  
**보기**:  
① 빠른 배포 가능  
② 모듈 간 낮은 결합도  
③ 서비스 간 통신 복잡성 증가  
④ 유연한 기술 스택  
⑤ 독립적 확장 가능  
**정답**: ③  
**해설**: MSA는 독립성이 장점이지만 서비스 간 통신이 복잡해지는 단점이 있다.

**문제36**

**문제**: REST API 응답에서 리소스를 성공적으로 생성했을 때 주로 사용하는 HTTP 상태 코드는?  
**보기**:  
① 200 OK  
② 204 No Content  
③ 201 Created  
④ 400 Bad Request  
⑤ 500 Internal Server Error  
**정답**: ③  
**해설**: 201 Created는 POST 요청으로 리소스가 성공적으로 생성되었음을 나타낸다.

**문제37**

**문제**: 개발자가 Git을 사용할 때 브랜치를 활용하는 주된 이유는?  
**보기**:  
① 로그 저장  
② 코드 컴파일  
③ 독립적 기능 개발  
④ 리소스 감소  
⑤ 트래픽 분산  
**정답**: ③  
**해설**: 브랜치는 독립적인 기능 개발이나 버그 수정을 위해 사용되며, 병합을 통해 통합된다.

**문제38**

**문제**: 테스트 자동화의 주요 장점 중 하나는 무엇인가?  
**보기**:  
① 개발 시간 증가  
② 수동 오류 유발  
③ 반복 작업 자동화  
④ 테스트 생략  
⑤ 복잡도 증가  
**정답**: ③  
**해설**: 자동화는 반복적인 테스트를 빠르고 정확하게 수행할 수 있게 해준다.

**문제39**

**문제**: UI/UX 설계에서 피드백이 필요한 가장 대표적인 예는?  
**보기**:  
① 정적 화면  
② 동영상 배경  
③ 버튼 클릭 후 상태 변화  
④ 동일한 색상 테마  
⑤ 페이지 로딩 시간  
**정답**: ③  
**해설**: 사용자 행동에 대한 시스템의 응답을 시각적으로 제공하는 것이 피드백이다.

**문제40**

**문제**: 유지보수 단계에서 가장 중요한 활동 중 하나는?  
**보기**:  
① 코드 주석 제거  
② 기능 삭제  
③ 버그 수정 및 개선  
④ 데이터베이스 재설계  
⑤ 요구사항 축소  
**정답**: ③  
**해설**: 유지보수는 기존 시스템의 오류를 수정하고 성능을 향상시키는 것이 핵심이다.

**문제41**

**문제**: 인터페이스 설계에서 시스템 간 데이터 연계 시 필수적으로 고려해야 할 항목은?  
**보기**:  
① UI 디자인  
② 응답 시간  
③ 데이터 포맷  
④ 폰트 스타일  
⑤ 소스코드 구조  
**정답**: ③  
**해설**: 데이터 포맷은 서로 다른 시스템이 데이터를 정확히 주고받기 위해 반드시 정의되어야 한다.

**문제42**

**문제**: 다음 중 UI 설계 원칙에 해당하지 않는 것은?  
**보기**:  
① 일관성  
② 단순성  
③ 가시성  
④ 보안성  
⑤ 직관성  
**정답**: ④  
**해설**: 보안성은 시스템 설계 관점이고, UI 설계 원칙에는 포함되지 않는다.

**문제43**

**문제**: 마이크로서비스 간 통신 방식으로 REST 대신 비동기 메시징을 사용하는 주요 이유는?  
**보기**:  
① 사용자 인터페이스 단순화  
② 서버 부하 증가  
③ 실시간 응답 제공  
④ 느린 응답 처리  
⑤ 느슨한 결합 유지  
**정답**: ⑤  
**해설**: 비동기 메시징은 서비스 간 강한 연결 없이 메시지만 주고받아 느슨한 결합을 가능하게 한다.

**문제44**

**문제**: 요구사항 분석 단계에서 작성된 유스케이스 다이어그램은 주로 어떤 내용을 포함하는가?  
**보기**:  
① UI 컴포넌트 목록  
② 시스템 내부 클래스 구조  
③ 사용자와 시스템 간 상호작용  
④ 테이블 명세서  
⑤ 네트워크 구성  
**정답**: ③  
**해설**: 유스케이스 다이어그램은 사용자의 행위와 시스템의 응답 간 상호작용을 표현한다.

**문제45**

**문제**: API 명세서에 가장 일반적으로 포함되지 않는 항목은?  
**보기**:  
① 요청 방식 (GET/POST 등)  
② 파라미터 설명  
③ 응답 코드 설명  
④ 보안 로그 기록  
⑤ 예제 요청/응답  
**정답**: ④  
**해설**: 보안 로그는 시스템 운영과 관련된 항목으로, API 명세서에는 포함되지 않는다.

**문제46**

**문제**: 단위 테스트 수행 시 테스트 대상이 외부 의존성을 가지는 경우 가장 효과적인 해결 방법은?  
**보기**:  
① 실제 DB 연결  
② API 호출 생략  
③ 목(Mock) 객체 사용  
④ 테스트 건너뛰기  
⑤ 무작위 값 삽입  
**정답**: ③  
**해설**: 목(Mock)은 외부 의존성을 대체하여 단위 테스트가 독립적으로 수행될 수 있게 한다.

**문제47**

**문제**: 백엔드 아키텍처에서 트랜잭션의 원자성을 보장하기 위한 기법은?  
**보기**:  
① SQL Injection  
② 캐시 적중  
③ ACID 원칙  
④ 데이터 파이프라인  
⑤ 로깅 미들웨어  
**정답**: ③  
**해설**: 트랜잭션의 원자성, 일관성, 고립성, 지속성을 보장하는 원칙이 ACID이다.

**문제48**

**문제**: 운영 환경에서 배포 전 기능 이상 유무를 확인하기 위해 실제와 유사한 환경에서 수행하는 테스트는?  
**보기**:  
① 단위 테스트  
② 회귀 테스트  
③ 시스템 테스트  
④ 통합 테스트  
⑤ 스테이징 테스트  
**정답**: ⑤  
**해설**: 스테이징 테스트는 실제 운영 환경과 유사한 조건에서 수행하여 배포 전 이상 여부를 점검한다.

**문제49**

**문제**: REST API 응답에서 요청이 잘못되었음을 나타내는 HTTP 상태 코드는?  
**보기**:  
① 200 OK  
② 201 Created  
③ 204 No Content  
④ 400 Bad Request  
⑤ 500 Internal Server Error  
**정답**: ④  
**해설**: 400 Bad Request는 클라이언트의 요청이 잘못되었을 때 반환된다.

**문제50**

**문제**: 인터페이스 요구사항 중 데이터 전송에 있어 ‘속도’와 가장 밀접하게 관련된 비기능 요구사항은?  
**보기**:  
① 보안성  
② 신뢰성  
③ 응답성  
④ 확장성  
⑤ 가용성  
**정답**: ③  
**해설**: 응답성은 시스템이 요청에 얼마나 빠르게 응답하는지를 의미한다.

**문제51**

**문제**: 다음 중 요구사항 정의 단계에서 주로 작성하는 산출물은 무엇인가?  
**보기**:  
① 테스트 케이스  
② 클래스 다이어그램  
③ UI 디자인 시안  
④ 요구사항 명세서  
⑤ 운영 매뉴얼  
**정답**: ④  
**해설**: 요구사항 명세서는 사용자 요구를 정리하고 시스템이 제공해야 할 기능을 명확히 문서화한 결과물이다.

**문제52**

**문제**: 화면 설계 시 '일관성' 원칙을 가장 잘 설명한 것은?  
**보기**:  
① 모든 기능은 하나의 색상으로 표시한다.  
② 동일한 기능은 동일한 위치와 표현으로 배치한다.  
③ 최대한 많은 정보를 제공한다.  
④ 기능별로 매번 다른 디자인을 적용한다.  
⑤ 애니메이션 효과를 최소화한다.  
**정답**: ②  
**해설**: 일관성은 사용자가 시스템을 예측하고 쉽게 사용할 수 있도록 동일한 규칙을 적용하는 것이다.

**문제53**

**문제**: MSA 환경에서 공통 기능(인증, 로깅 등)을 처리하는 API Gateway의 역할 중 가장 적절한 것은?  
**보기**:  
① 데이터 분석  
② 사용자 인터페이스 처리  
③ 서비스별 로직 분기  
④ 공통 기능 집중 처리  
⑤ 데이터 마이그레이션  
**정답**: ④  
**해설**: API Gateway는 인증, 로깅, 라우팅 등 공통 기능을 처리하여 마이크로서비스 간 복잡성을 낮춰준다.

**문제54**

**문제**: UI 요소 간의 상호작용 흐름을 시각적으로 표현한 다이어그램은?  
**보기**:  
① 클래스 다이어그램  
② 상태 다이어그램  
③ 시퀀스 다이어그램  
④ 화면 흐름도  
⑤ 유스케이스 다이어그램  
**정답**: ④  
**해설**: 화면 흐름도는 UI 간 전환 및 사용자의 동선을 시각적으로 표현하는 도구이다.

**문제55**

**문제**: 다음 중 테스트 시 ‘정상 입력값 경계’를 검증하는 데 적절한 테스트 기법은?  
**보기**:  
① 동등 분할  
② 경계값 분석  
③ 상태 전이  
④ 탐색 테스트  
⑤ 회귀 테스트  
**정답**: ②  
**해설**: 경계값 분석은 입력값의 상한, 하한 경계에서 시스템이 올바르게 작동하는지를 확인하는 기법이다.

**문제56**

**문제**: 서비스 장애 발생 시 자동으로 복구되도록 설정하는 기능은?  
**보기**:  
① 리포트 기능  
② Failover  
③ 리팩토링  
④ 세션 클러스터링  
⑤ 데이터 암호화  
**정답**: ②  
**해설**: Failover는 장애 발생 시 자동으로 대체 시스템으로 전환하여 서비스를 지속할 수 있게 하는 기능이다.

**문제57**

**문제**: 테스트 계획 수립 시 가장 먼저 정의해야 할 항목은?  
**보기**:  
① 테스트 데이터  
② 테스트 도구  
③ 테스트 범위 및 목표  
④ 버그 추적 방식  
⑤ 리포트 양식  
**정답**: ③  
**해설**: 테스트의 전체 방향성과 목적을 결정하기 위해 가장 먼저 범위 및 목표를 설정한다.

**문제58**

**문제**: 비즈니스 로직 설계 시 가장 우선적으로 고려해야 할 것은?  
**보기**:  
① 데이터 암호화  
② 사용자 인터페이스  
③ 업무 흐름 및 규칙  
④ 보고서 양식  
⑤ 로깅 방식  
**정답**: ③  
**해설**: 비즈니스 로직은 업무 규칙을 반영해야 하므로, 가장 먼저 해당 흐름과 규칙을 고려해야 한다.

**문제59**

**문제**: 요구사항이 불완전하거나 모호할 경우 가장 바람직한 대응은?  
**보기**:  
① 그대로 개발 진행  
② 유사 프로젝트 참고  
③ 사용자와 추가 인터뷰 진행  
④ 개발자 추측으로 판단  
⑤ 문서 생략  
**정답**: ③  
**해설**: 불완전한 요구사항은 반드시 사용자와의 추가 커뮤니케이션을 통해 명확히 해야 한다.

**문제60**

**문제**: 단위 테스트를 자동화하기 위한 가장 일반적인 도구는?  
**보기**:  
① Jenkins  
② Selenium  
③ JUnit  
④ Grafana  
⑤ Git  
**정답**: ③  
**해설**: JUnit은 Java 기반 단위 테스트 자동화를 위한 대표적인 프레임워크이다.