

[정답 및 해설] [인터페이스 구현>인터페이스 설계서 확인]

인터페이스 설계서 확인 1. ① 사용자 ② 시스템

- * 인터페이스 요구사항 유형
 - 사용자 인터페이스 요구사항
 - 시스템 인터페이스 요구사항

인터페이스 설계서 확인 2. 인터페이스 설계서(정의서)

인터페이스 설계서(정의서)에 대한 설명이다.

인터페이스 설계서 확인 3. ① 시스템 ② 상세 기능별

- * 인터페이스 설계서(정의서)
 - 시스템 인터페이스 정의서
 - 상세 기능별 인터페이스 정의서

인터페이스 설계서 확인 4. ① 인터페이스 목록 ② 인터페이스 명세

- * 인터페이스 정의서
 - 인터페이스 목록
 - 인터페이스 명세

인터페이스 설계서 확인 5. (¬)

- * 정적/동적 모형, 데이터 포맷 형태에 따른 인터페이스 설계서
 - 정적, 동적 모형을 통한 인터페이스 설계서
 - 데이터 정의를 통한 인터페이스 설계서

인터페이스 설계서 확인 6. (¬)→(≡)→(⊂)

- * 외부, 내부 모듈 간 공통 기능 및 데이터 인터페이스 확인 순서
 - 1단계: 인터페이스 설계서의 외부 및 내부 모듈의 기능을 확인한다.
 - 2단계: 인터페이스 설계서의 외부 및 내부 모듈을 기반으로 공통적으로 제공되는 기능을 확인한다.
 - 3단계: 인터페이스 설계서의 외부 및 내부 모듈을 기반으로 각 데이터의 인터페이스를 확인한다.

인터페이스 설계서 확인 7. 인터페이스 데이터 표준

외부 및 내부 모듈 간 데이터를 교환하고 상호 호환이 되게 하기 위해서 인터페이스 데이터 표준을 정의하고 이를 관리하여야 한다.

인터페이스 설계서 확인 8. ① 외부 ② 내부

- * 외부, 내부 모듈 연계를 위한 인터페이스 기능 식별 순서
 - 1단계: 개발하고자 하는 응용소프트웨어와 관련된 외부 및 내부 모듈 간 연계된 기능을 식별한다.
 - 2단계: 외부 및 내부 모듈 간 연계된 기능에 따른 인터페이스 기능을 식별한다.

[정답 및 해설] [인터페이스 구현>인터페이스 설계서 확인]

인터페이스 설계서 확인 9. $(\neg) \rightarrow (\rightleftharpoons) \rightarrow (\perp) \rightarrow (\sqsubset)$

* 외부 및 내부 모듈 간 인터페이스 데이터 표준 확인
순서

- 1단계: 데이터 인터페이스의 의미 파악
- 2단계: 데이터 인터페이스 의미 파악을 통한 데이터 표준 확인
- 3단계: 인터페이스 데이터 표준 식별
- 4단계: 데이터 표준 및 조정해야 할 항목을 검토하고 최종적으로 확인

[정답 및 해설] [인터페이스 구현>인터페이스 기능 구현]

인터페이스 기능 구현 1. ① 컴포넌트 명세서 ② 인터페이스 명세서

- 컴포넌트 명세서: 컴포넌트의 개요 및 내부 클래스의 동작, 인터페이스를 통해 외부와 통신하는 명세를 정의한다.
- 인터페이스 명세서: 컴포넌트 명세서에 명시된 인터페이스 클래스의 세부적인 조건 및 기능을 명시한 명세서이다.

인터페이스 기능 구현 2. 인터페이스

- * 인터페이스 기능 구현 과정
- 1단계: 모듈 간 세부 설계서를 확인한다.
- 2단계: 일관되고 정형화된 인터페이스 기능 구현을 정의한다.

인터페이스 기능 구현 3. ① JSON ② XML

- JSON: 속성-값 쌍으로 이루어진 데이터 오브젝트를 전달하기 위해 사용하는 개방형 표준 포맷이다.
- XML: 웹브라우저 간 HTML 문법이 호환되지 않는 문제와 SGM의 복잡함을 해결하기 위하여 개발된 다목적 마크업 언어이다.

인터페이스 기능 구현 4. 엔티티

- * 인터페이스 엔티티를 통한 인터페이스 구현
- 인터페이스가 필요한 시스템 사이에 별도의 인터페이스 엔티티를 두어서 상호 연계한다.

인터페이스 기능 구현 5. 기능 구현

- * 인터페이스 구현 과정
- 1단계: 사전에 정의된 기능 구현을 분석한다.
- 2단계: 인터페이스를 구현한다.

인터페이스 기능 구현 6. ① Write ② Read

- JSON: 속성-값 쌍으로 이루어진 데이터 오브젝트를 전달하기 위해 사용하는 개방형 표준 포맷이다.
- XML: 웹브라우저 간 HTML 문법이 호환되지 않는 문제와 SGM의 복잡함을 해결하기 위하여 개발된 다목적 마크업 언어이다.

인터페이스 기능 구현 7. AJAX

- 송신 측에서 인터페이스 객체를 송신 시 예외를 처리하는 방안: AJAX 호출 후 반환 값을 받아 어떻게 처리할지를 호출하는 부분에서 사전 정의

인터페이스 기능 구현 8. ① (¬) ② (┌) ③ (┐)

- 인터페이스 객체 송신 시 예외가 발생하는 케이스를 정의하고 각 예외 케이스마다 예외 처리 방안을 정의한다. 예 외가 발생하는 경우는 시스템의 환경과 송신하는 데이터, 프로그램 자체의 원인이 있을 수 있다.

[정답 및 해설] [인터페이스 구현>인터페이스 기능 구현]

인터페이스 기능 구현 9. ① 송신 ② 수신

- 인터페이스 엔티티를 사용한 인터페이스에서의 예외 처리 방안은 송신 인터페이스 테이블에서 인터페이스 예외를 처리하는 방안과 수신 인터페이스 테이블에서 인터페이스 예외를 처리하는 방안이 있다.

인터페이스 기능 구현 10. ① 스니핑(Sniffing) ② 스니퍼(Sniffer)

- 데이터 통신 시 데이터 탈취 위협: 데이터 통신 내역을 중간에서 감청하여 기밀성을 훼손할 수 있는 기법으로 스니핑이라고 불리기도 한다. 스푸핑처럼 공격 대상을 직접적으로 속이는 해킹이 아닌 수동적 해킹 공격 기법이며, 도청할 수 있도록 중간에 설치되는 도구를 스니퍼라고 한다. 주로 패킷 분석기 같은 툴을 통해서 진행된다.

인터페이스 기능 구현 11. 시큐어 코딩

- 시큐어 코딩: 대표적인 웹 애플리케이션의 보안 취약점 발표 사례인 OWASP Top 10을 참고하여 KISA에서 SW 보안 약점 가이드를 발표하였고 SW 보안 취약점, 약점 및 대응 방안이 구체적으로 서술되어 있다.

인터페이스 기능 구현 12.

- 데이터베이스 암호화 알고리즘: 대칭 키 암호 알고리즘, 해시 알고리즘, 비대칭 키 알고리즘

인터페이스 기능 구현 13. ① API ② Plug-in

- 데이터베이스 암호화 기법에는 애플리케이션에서 암호화를 수행하는 API 방식과 데이터베이스에서 암호화를 수행하는 Plug-in 방식, API 방식과 Plug-in 방식을 혼합한 Hybrid 방식이 있다.

인터페이스 기능 구현 14. 보안 취약점

- * 인터페이스 보안 기능 적용 과정
 - 1단계: 인터페이스의 보안 취약점을 분석한다.
 - 2단계: 분석된 보안 취약점을 근거로 인터페이스 보안 기능을 적용한다.

인터페이스 기능 구현 15. AH(인증 헤더)

- AH(인증 헤더): 출발지 인증, 데이터 무결성은 제공하지만 기밀성은 제공하지 않는다.

인터페이스 기능 구현 16. IKE

- IKE: 인터넷 표준 암호 키 교환 프로토콜

인터페이스 기능 구현 17. ① 전송 모드 ② 터널 모드

- 전송 모드(transport mode): 종단 호스트 간의 보안 터널을 제공 (컴퓨터-컴퓨터)
- 터널 모드(tunnel mode): 보안 게이트웨이 간의 보안 터널을 제공 (LAN-LAN)

[정답 및 해설] [인터페이스 구현>인터페이스 기능 구현]

인터페이스 기능 구현 18. 인터페이스 테이블

- 인터페이스 테이블: 데이터베이스 테이블을 사용하여 데이터를 교환하는 시스템과의 통합을 위한 옵션

인터페이스 기능 구현 19. ① (ㄷ) ② (ㄱ) ③ (ㄴ)

- 데이터베이스 보안 고려사항으로는 데이터베이스 접근 권한, 악의적 코드 삽입 금지, 민감 데이터 관리, 악의적 시도 시 에러 처리 등이 있다.

[정답 및 해설] [인터페이스 구현>인터페이스 구현 검증]

인터페이스 구현 검증 1. xUnit, STAF, FitNesse, NTAF, Selenium, watir

* 인터페이스 구현 검증 도구

- xUnit
- STAF
- FitNesse
- NTAF
- Selenium
- watir

인터페이스 구현 검증 2. APM

애플리케이션 모니터링 툴(APM: Application Performance Management)에 대한 설명이다.

인터페이스 구현 검증 3. ① (ㄴ) ② (ㄷ) ③ (ㄱ)

* 감시 도구 요건

1. 송신 측에서 인터페이스 대상 선택 전송
 - 데이터베이스 SQL 모니터링
 - 조회 Transaction 모니터링
 - JSON 생성 객체 모니터링
2. 인터페이스 객체 전송
 - 통신 암호화 모니터링
 - 패킷정보 모니터링
 - 연결된 Transaction 변수 모니터링
3. 수신 후 수신 측 트랜잭션과 결과 반환
 - 객체 입력, 출력값 모니터링
 - 객체 동작 성공, 실패 여부

인터페이스 구현 검증 4. ① (ㄴ) ② (ㄷ) ③ (ㄱ)

* 인터페이스 오류 처리 방법

- 사용자 화면에서 오류를 인지하는 방법
- 인터페이스 오류 로그 생성
- 인터페이스 관련 테이블에 오류 사항 기록

인터페이스 구현 검증 5. ① (ㄴ) ② (ㄷ) ③ (ㄱ)

* 인터페이스 오류 처리 보고 시기에 따른 구분

- 최초 발생 보고
- 오류 처리 경과 보고
- 완료 보고

인터페이스 구현 검증 6. ① (ㄱ) ② (ㄱ) ③ (ㄷ) ④ (ㄴ) ⑤ (ㄴ) ⑥ (ㄷ)

* 주기적으로 인터페이스 오류 발생 확인 방법

- 인터페이스 오류 로그 확인
- 인터페이스 오류 테이블 확인
- 인터페이스 감시 도구(monitoring) 활용