

- 문 11. 배열, 레코드, 구조체 등을 매개변수로 사용하는 모듈 사이의 결합도는?
 ① 데이터 결합도(Data Coupling)
 ② 스탬프 결합도(Stamp Coupling)
 ③ 제어 결합도(Control Coupling)
 ④ 내용 결합도(Content Coupling)

문 12. 요구분석 단계를 순서대로 바르게 나열한 것은?

ㄱ. 요구사항 검증
 ㄴ. 요구사항 명세화
 ㄷ. 타당성 조사
 ㄹ. 요구사항 추출 및 분석

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

문 13. 시스템을 구성하는 물리적인 노드와 통신 경로, 그리고 컴포넌트의 수행 환경을 표시하는 UML 다이어그램은?

- ① 배치(deployment) 다이어그램
 ② 통신(communication) 다이어그램
 ③ 타이밍(timing) 다이어그램
 ④ 컴포지트(composite) 다이어그램

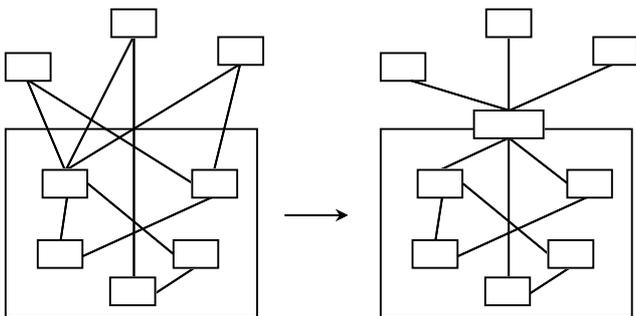
문 14. 프로그램 인스펙션(inspection)을 통해 결함을 검출하고자 할 때, 결함 유형에 따른 검사항목을 옳게 짝지은 것은?

- ① 데이터 결함 - 모든 반복문은 확실히 종료되는가?
 ② 제어 결함 - 버퍼 오버플로우의 가능성이 있는가?
 ③ 인터페이스 결함 - 매개변수들의 순서가 올바른가?
 ④ 메모리 관리 결함 - 예상하지 않은 입력이 문제를 일으킬 수 있는가?

문 15. Iterator 패턴에 대한 설명으로 옳지 않은 것은?

- ① 집합 객체의 요소들에 대해 순서대로 접근하는 방법을 제공한다.
 ② 배열(Array), 배열리스트(ArrayList), 해시 테이블과 같은 객체를 처리하는 데 사용하는 패턴이다.
 ③ 서로 다른 인터페이스를 사용하기 때문에 각각의 객체를 참조하기 위한 다형성 코드(polymorphic code)를 개발하는 것이 불가능하다.
 ④ 서로 다른 집합 객체 구조에 대해 동일한 방법으로 순회할 수 있다.

문 16. 그림과 같이 서브시스템 사이의 의사소통 및 종속성을 최소화하기 위하여 단순화된 하나의 인터페이스를 제공하는 디자인 패턴은?

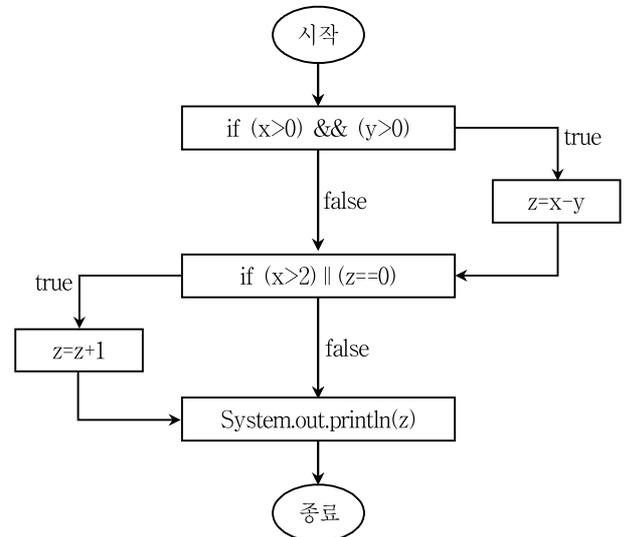


- ① Adapter 패턴
 ② Bridge 패턴
 ③ Decorator 패턴
 ④ Facade 패턴

문 17. 통합 테스트에 대한 설명으로 옳지 않은 것은?

- ① 회귀 테스트는 복잡하고 시간이 중요한 프로젝트에 적용하면 효율적이다.
 ② 점증적인 통합은 빅뱅 방식에 비해 인터페이스를 완전하게 테스트할 가능성이 더 높다.
 ③ 객체지향 시스템의 드라이버는 사용자 인터페이스를 대체하여 사용함으로써 인터페이스의 구현 이전에 기능성 테스트를 수행할 수 있다.
 ④ 객체지향 시스템의 스텝은 하나 이상의 협동 클래스들이 아직 완전히 구현되지 않은 상황에서 사용할 수 있다.

문 18. 다음 제어 흐름 그래프에 나타난 프로그램을 테스트할 때, 옳지 않은 것은?



- ① 분기 커버리지를 만족하는 최소의 테스트 케이스는 2개이다.
 ② 기본 경로의 개수는 3개이다.
 ③ 문장 커버리지를 만족하는 최소의 테스트 케이스는 1개이다.
 ④ {(x:1, y:2, z:0), (x:5, y:0, z:0)}은 분기 커버리지를 만족하지 못한다.

문 19. 오픈 소스 도구와 기능을 연결한 것으로 옳지 않은 것은?

- ① Cucumber - 행위 주도 개발(Behavior Driven Development) 지원 도구
 ② Maven - 프로젝트 관리 도구
 ③ Git - 분산 버전 관리 도구
 ④ JaCoCo - 자바 소스 보안 취약점 분석 도구

문 20. 국제표준과 관련된 내용으로 옳지 않은 것은?

- ① IEEE 830 - 소프트웨어 요구명세서 표준
 ② IEEE 1219 - 소프트웨어 유지보수 표준
 ③ IEC 62304 - 항공기 소프트웨어 안전성 표준
 ④ ISO 26262 - 자동차 기능 안전성 표준