

소프트웨어공학

문 1. 통합 프로세스 모델(UP: Unified Process)의 특징을 설명한 것으로 옳지 않은 것은?

- ① 반복적(iterative)이고, 점진적(incremental)인 개발 방법이다.
- ② 유스케이스를 기반으로 한다.
- ③ 아키텍처 중심의 개발을 지향한다.
- ④ 위험 관리는 포함되어 있지 않다.

문 2. 소프트웨어 테스트에 대한 설명으로 옳지 않은 것은?

- ① 동료 검토(peer review)는 개발한 원시 코드나 여러 가지 산출물에 대한 검토를 의뢰하여 오류를 찾는 방법이다.
- ② 기본 경로 테스트(basic path test)는 원시 코드의 독립적인 경로가 최소한 한 번은 실행되는 테스트 케이스를 찾아 테스트를 수행한다.
- ③ 회귀 테스트(regression test)는 한 모듈의 수정이 다른 부분에 미치는 영향을 최소화하기 위해 필요하다.
- ④ 블랙 박스 테스트(black box test)는 입력 값에 대한 예상 출력 값을 정해 놓고 그대로 결과가 나오는지 원시 코드를 보며 확인한다.

문 3. 요구분석명세서를 작성할 때 고려해야 할 사항으로 옳지 않은 것은?

- ① 시스템의 기능을 구현하는 방법에 대하여 중점적으로 기술한다.
- ② 제안된 시스템에 영향을 주는 제약조건을 기술한다.
- ③ 시스템의 인수를 위한 테스트 기준을 제공한다.
- ④ 소프트웨어의 품질 기준에 대한 우선순위를 정한다.

문 4. 소프트웨어 유지보수의 형태에 대한 설명으로 옳지 않은 것은?

- ① 수정 유지보수(corrective maintenance)는 개발된 소프트웨어를 사용자가 인도받은 후 사용하면서 발견되는 오류를 잡는 것이다.
- ② 예방 유지보수(preventive maintenance)는 미리 예상되거나 예측되는 오류를 찾아 수정하는 것이다.
- ③ 적응 유지보수(adaptive maintenance)는 개발 과정에서 바로 잡지 못한 오류를 유지보수 단계에서 해결하는 것이다.
- ④ 완전 유지보수(perfective maintenance)는 결함으로 인해 요청된 변경뿐만 아니라 시스템의 일부 측면을 향상시키기 위한 변경을 포함하고 있다.

문 5. 허가받지 않은 사용자가 데이터 접근을 통해 변경을 시도했을 때 보호할 수 있는지 정도를 나타내는 품질 특성에 해당하는 것은?

- ① 무결성(integrity)
- ② 융통성(flexibility)
- ③ 안전성(safety)
- ④ 안정성(stability)

문 6. 애자일(agile) 선언문에 대한 설명으로 옳지 않은 것은?

- ① 프로세스와 도구보다 개인과 그들의 협업에 더 가치를 둔다.
- ② 포괄적인 문서화보다 제대로 작동하는 소프트웨어에 더 가치를 둔다.
- ③ 고객과의 협력보다는 계약 협상에 더 가치를 둔다.
- ④ 계획에 따르기보다는 변화에 대응하는 것에 더 가치를 둔다.

문 7. 다음에서 설명하는 디자인 패턴에 해당하는 것은?

이미 만들어져 있는 클래스를 사용하고 싶지만 인터페이스가 원하는 방식과 일치하지 않을 때, 또는 관련성이 없거나 예측하지 못한 클래스들과 협동하는 재사용 가능한 클래스를 생성하기를 원할 때 사용한다.

- ① Bridge
- ② Adapter
- ③ Composite
- ④ Facade

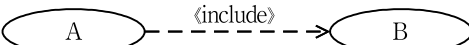
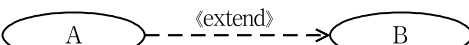
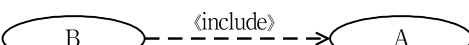
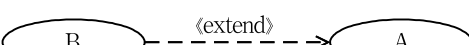
문 8. 비기능 요구사항에 대한 설명으로 옳지 않은 것은?

- ① 예산의 제약, 조직의 정책, 다른 소프트웨어와 하드웨어 시스템과의 상호 운영성, 안정성 규칙과 프라이버시 보호법과 같은 사용자의 필요에 의해 발생한다.
- ② 요구사항과 목표를 혼합한 문서를 사용하여 비기능적 요구사항을 표현한다.
- ③ 시스템에서 제공되는 서비스나 기능에 대한 제약이다.
- ④ 시스템이 제공해야 하는 서비스와 시스템이 특정 입력에 대해 어떻게 반응하는지, 시스템이 특정 상황에서 어떻게 동작해야 하는지에 관한 사항이다.

문 9. 객체지향 개념에 대한 설명으로 옳지 않은 것은?

- ① 정보은닉(information hiding)은 모듈 안에 있는 자료구조와 메서드에 사용된 알고리즘을 외부에서 직접 변경할 수 없고, 공개 인터페이스로 정의된 메서드를 통해서만 접근할 수 있는 것을 의미한다.
- ② 전체 객체에 전속되어 독립된 객체로 존재할 수 없는 부분 객체도 있는데 이와 같은 관계를 특별히 집합 관계(aggregation relationship)라고 한다.
- ③ 리스코프 교체 원칙(LSP: Liskov Substitution Principle)은 부모 클래스와 자식 클래스 사이의 행위가 일관성이 있어야 한다는 의미이다.
- ④ 다형성(polymorphism)은 같은 오퍼레이션이 다른 클래스에서 다르게 동작하는 것을 말하며, 하나의 함수 이름이나 연산자가 여러 목적으로 사용될 수 있는 것을 의미한다.

문 10. 유스케이스 다이어그램에서 A 유스케이스를 수행하는 도중에 특정 조건을 만족하면 B 유스케이스를 수행한다. A 유스케이스와 B 유스케이스 간의 관계로 옳은 것은?

- ① 
- ② 
- ③ 
- ④ 

문 11. 객체지향 시스템 테스트의 설명으로 옳지 않은 것은?

- ① 클래스에 대한 테스트 시 필요로 하는 객체 상태를 고려하지 않는다.
- ② 클래스 클러스터 차원에서 상속, 캡슐화, 동적 바인딩, 다형성 등의 이유로 관계가 맺어진 클래스들의 집합은 한 덩어리로 간주하여 연결 관계를 확인해야 한다.
- ③ 서브클래스를 테스트할 때 부모클래스의 테스트 케이스를 재사용할 수 있다.
- ④ 클래스 차원에서 메서드 상호 간, 메서드와 데이터 멤버 간의 연결 관계의 정확성을 확인해야 한다.

문 12. 소프트웨어 재사용에 대한 설명으로 옳지 않은 것은?

- ① 사용자 인터페이스와 같은 표준은 표준화된 재사용 가능한 컴포넌트의 집합으로 구현될 수 있다.
- ② 응용시스템, 컴포넌트, 객체, 함수는 재사용이 가능하다.
- ③ 소프트웨어 재사용은 기존의 소프트웨어를 활용하므로 수정하거나 이해하기 위한 추가의 노력이 들지 않는다.
- ④ 동일한 작업을 수차례 반복하는 대신 전문가들의 지식이 요약된 재사용 가능한 소프트웨어를 개발, 활용할 수 있다.

문 13. 형상관리에 대한 설명으로 옳은 것은?

- ① 컴포넌트 기반의 소프트웨어 개발 프로젝트에서는 형상관리가 필요하지 않다.
- ② 베이스라인은 단위 테스트를 통과한 원시 코드의 집합을 의미한다.
- ③ ISO 9000에서는 형상관리에 대한 언급이 없으나 CMMI에서는 형상관리 프로세스를 요구한다.
- ④ 형상관리는 소프트웨어 품질 보증을 위한 주요한 요소이며, 주 임무는 변경의 통제이다.

문 14. 다음에서 설명하는 CMMI의 성숙도 단계에 해당하는 것은?

표준 프로세스를 면밀히 검토하여 보완하고 최신 기술들을 반영하여 지속적으로 프로세스를 개선하고 이 개선된 프로세스를 전 조직이 사용하도록 한다.

- ① 관리(managed) 단계
- ② 정의(defined) 단계
- ③ 정량적 관리(quantitatively managed) 단계
- ④ 최적화(optimizing) 단계

문 15. 인수 테스트(acceptance test)에 대한 설명으로 옳지 않은 것은?

- ① 인수 테스트의 목적은 사용자에게 소프트웨어가 개발되어 사용될 준비가 되었다는 확신을 주기 위한 것이다.
- ② 알파 테스트는 선택된 사용자가 사용자 환경에서 수행하는 인수 테스트이다.
- ③ 사용자 스토리를 작성하면서 함께 작성한 테스트 시나리오에 따라 고객이 직접 테스트한다.
- ④ 개발자팀이 소프트웨어를 사용자에게 배포하여 사용자가 자신의 컴퓨터 환경 또는 실제 상황에서 수행하는 테스트이다.

문 16. McCabe의 순환복잡도(CC : Cyclometric Complexity)에 대한 설명으로 옳지 않은 것은?

- ① 프로그램의 논리적인 복잡도를 정량적으로 측정하기 위한 척도(metric)이다.
- ② 프로그램의 분기 노드의 수보다 항상 큰 값을 갖는다.
- ③ 프로그램의 모든 경로를 최소 한 번 이상 실행하도록 테스트하기 위한 입력의 개수에 해당한다.
- ④ 플로우 그래프(flow graph)에서 간선(edge)의 수를 E, 노드의 수를 N이라고 하면 $CC = E - N + 2$ 이다.

문 17. COCOMOII의 서브모델이 아닌 것은?

- ① 초기 설계 모델(early design model)
- ② 임베디드 모델(embedded model)
- ③ 응용 결합 모델(application composition model)
- ④ 재사용 모델(reuse model)

문 18. 객체지향 설계 기법 중 “클라이언트는 자신이 사용하지 않는 메서드에 의존 관계를 맺으면 안 된다.”라는 설계 원칙으로 옳은 것은?

- ① 단일 책임의 원칙
- ② 개방 폐쇄의 원칙
- ③ 의존 관계 역전의 원칙
- ④ 인터페이스 분리의 원칙

문 19. 소프트웨어 품질에 대한 설명으로 옳지 않은 것은?

- ① ISO/IEC 12119 - 패키지 소프트웨어의 일반적인 제품 품질 요구사항 및 테스트를 위한 국제 표준 규격이다.
- ② 매트릭(metric) - 품질 기준별로 측정방법과 스케일 등을 정의하여 정확히 품질 기준을 측정할 수 있게 한다.
- ③ 인스펙션(inspection) - 체크리스트를 가지고 본인이 개발한 코드와 산출물 등을 검토하는 것이다.
- ④ 품질 기준(criteria) - 개발자 측면의 내부 관점을 나타내는 것이다.

문 20. 요구사항 품질 속성에 대한 설명으로 옳지 않은 것은?

- ① 기능성(functionality) - 요구분석명세서가 ‘어떻게’ 보다 ‘무엇을’에 관점을 두고 기술되어야 함을 의미한다.
- ② 명확성(unambiguity) - 요구사항이 모든 이해관계자들에 의해 한 가지 의미로 해석되어야 한다.
- ③ 완전성(completeness) - 요구사항을 바로 구현할 수 있도록 자료구조나 알고리즘이 명시되어야 한다.
- ④ 일관성(consistency) - 요구사항들이 서로 모순되지 않아야 한다.