

소프트웨어공학

1. 애자일 방법론에 대한 설명으로 옳지 않은 것은?

- ① 요구사항의 변화가 적은 대규모 시스템 개발에 적합하다.
- ② 포괄적인 문서보다 실행 가능한 소프트웨어를 중시한다.
- ③ 프로세스와 도구보다 개인과의 상호 소통을 중시한다.
- ④ 익스트림 프로그래밍(XP), 동적 시스템 개발 방법론(DSDM) 등이 있다.

2. 객체지향 설계 원칙에 대한 설명으로 옳지 않은 것은?

- ① 확장에는 열려 있어야 하고 변경에는 닫혀 있어야 한다.
- ② 상위 클래스의 객체는 하위 클래스의 객체로 교체될 수 있어야 한다.
- ③ 클라이언트는 추상 클래스나 인터페이스가 아닌 구체(concrete) 클래스에 의존해야 한다.
- ④ 하나의 범용 인터페이스보다는 여러 개의 특화된 인터페이스로 나누어야 한다.

3. 요구사항 수집 기법에 대한 설명으로 옳지 않은 것은?

- ① 인터뷰는 사용자 또는 관련자들과의 직접적인 대화를 통하여 요구사항을 추출한다.
- ② 워크스루는 업무에서 사용하고 있는 다양한 종류의 문서를 분석하는 기법이다.
- ③ JAD(Joint Application Development)는 프로젝트 관리자, 사용자, 개발자 등이 모여 상호 토론하는 방법이다.
- ④ 관찰은 개발될 소프트웨어의 사용자가 수행하는 활동을 살펴보는 기법이다.

4. (가), (나)에 해당하는 용어를 바르게 연결한 것은?

- (가) 모듈과 모듈 사이의 관계에서 관련 정도를 나타내는 용어이다. 모듈 간에는 관련이 적을수록 상호 의존성이 줄어들어 모듈의 독립성이 높아지고 독립성이 높아지면 모듈 간에 영향이 적어져 좋은 설계가 된다.
- (나) 모듈 내부에 존재하는 구성 요소 사이의 밀접한 정도를 나타내는 용어이다. 밀접한 정도가 높을수록 필요한 구성 요소만 모여 있고, 낮을수록 서로 관련성이 적은 구성 요소들이 모여 있다.

(가)

- ① 결합도(coupling)
- ② 결함도(defect)
- ③ 응집도(cohesion)
- ④ 응집도(cohesion)

(나)

- 응집도(cohesion)
- 성숙성(maturity)
- 결합도(coupling)
- 성숙성(maturity)

5. 컴포넌트 기반 개발에 대한 설명으로 옳지 않은 것은?

- ① 컴포넌트는 정적으로 바인딩되며 실행시간에는 인터페이스를 통해 접근할 수 없다.
- ② 재사용 가능한 컴포넌트를 기반으로 소프트웨어를 개발하는 방법론이다.
- ③ 컴포넌트는 독립적이고 교체 가능한 부품이다.
- ④ 컴포넌트는 인터페이스를 통해 다른 컴포넌트와 연결된다.

6. V 모델에 대한 설명으로 옳은 것만을 모두 고르면?

- ㄱ. V 모델은 인수 테스트를 포함한다.
- ㄴ. 폭포수 모델의 변형으로 테스트 단계를 추가 확장한 것이다.
- ㄷ. 폭포수 모델이 산출물 중심이라면, V 모델은 각 개발 단계를 검증하는 데 초점을 둔다.

- ① ㄱ, ㄴ
- ② ㄱ, ㄷ
- ③ ㄴ, ㄷ
- ④ ㄱ, ㄴ, ㄷ

7. CMMI(Capability Maturity Model Integration)의 단계에 대한 설명으로 옳지 않은 것은?

- ① 관리(managed) – 프로젝트 진행 비용, 일정, 기능에 대한 추적이 가능한 단계
- ② 정의(defined) – 소프트웨어 개발과정이 임시방편이거나 무질서하여 이를 정의해야 하는 단계
- ③ 최적화(optimizing) – 정량적인 피드백과 기술을 통해 소프트웨어 개발 프로세스가 지속적으로 개선되는 단계
- ④ 정량적 관리(quantitatively managed) – 소프트웨어 프로세스와 제품의 품질에 대해 정량적 측정이 이루어지는 단계

8. 다음 설명에 해당하는 아키텍처 패턴은?

- 분리된(decoupled) 컴포넌트들이 원격 서비스를 호출해 상호작용하는 시스템을 구조화할 때 유용하다.
- OMG(Object Management Group)의 CORBA는 이 아키텍처의 대표적인 사례이다.

- ① MVC(Model-View-Controller)
- ② 계층(layered)
- ③ 브로커(broker)
- ④ 파이프 필터(pipe-and-filter)

9. K 대학은 외국인 신입생의 증가에 대응하기 위해 다음과 같이 정보 시스템을 유지보수하려고 한다. (가), (나)에 해당하는 유지보수 유형을 바르게 연결한 것은?

- (가) DBMS가 다국어 지원이 가능한 제품으로 변경되어 이와 관련된 프로그램을 수정하고자 한다.
- (나) 기존 비대면 수업 등록 기능에서 발생한 오류 사항을 수정하고자 한다.

(가)

- ① 수정(corrective)
- ② 예방(preventive)
- ③ 완전(perfective)
- ④ 적응(adaptive)

(나)

- 예방(preventive)
- 수정(corrective)
- 적응(adaptive)
- 수정(corrective)

10. 프로그램의 내부 구조를 참조하지 않고, 요구사항 명세서를 기반으로 입력 데이터를 선택한 후 출력 결과를 확인하는 테스트 기법에 해당하지 않는 것은?

- ① 경계값 분석(boundary value analysis)
- ② 동등분할(equivalence partitioning)
- ③ 원인-결과 그래프(cause-effect graph)
- ④ 자료 흐름(data flow)

11. 비기능적 요구사항에 해당하는 것은?

- ① 전자레인지 시스템은 쿠키킹 중일 때 남은 쿠키킹 시간을 계산하여 표시한다.
- ② 프로그램은 400 MB 이내의 메모리를 사용해야 한다.
- ③ 입력한 문서의 음절, 단어, 문장, 줄, 문단, 문장당 단어의 수를 분석하여 표 형태로 보여 줄 수 있어야 한다.
- ④ 예약 시스템은 특정일에 예약한 고객의 리스트를 출력한다.

12. 요구사항 검증 항목에 대한 설명으로 옳지 않은 것은?

- ① 완전성 – 사용자 요구사항이 빠짐없이 완전히 반영되었는가?
- ② 일관성 – 요구사항 간의 모순이나 충돌이 존재하는가?
- ③ 추적성 – 개발 결과와 요구사항 일치를 검증할 수 있는가?
- ④ 명확성 – 서술된 명세서의 내용이 모호하지 않고 간결하고 명쾌한가?

13. 학사관리시스템에서 파악된 다음 코드 악취를 해결하기 위한 리팩토링 기법은?

- enrol() 메서드의 코드가 4,000라인이 넘어 가독성이 낮다.
- study()와 credit() 메서드는 각각 200라인 정도지만, 두 메서드는 100라인 정도가 중복되어 있다.

- ① 메서드 추출(extract method)
- ② 위임 숨기기(hide delegate)
- ③ 중개자 제거(remove middle man)
- ④ 클래스 인라인화(inline class)

14. 나선형(spiral) 모델에서 수행하는 활동과 가장 거리가 먼 것은?

- ① 개발(develop)
- ② 리스크 식별 및 해결(identify, resolve risks)
- ③ 목표 정의(determine objectives)
- ④ 유지보수(maintenance)

15. 다음 설명에 해당하는 테스트는?

시스템을 변경한 후 변경된 부분이 기존 시스템에 부작용을 일으키지 않는지를 조사하기 위해 이미 사용한 테스트 케이스를 재사용하여 실행하는 테스트

- ① 베타(beta) 테스트
- ② 스트레스(stress) 테스트
- ③ 신뢰성(reliability) 테스트
- ④ 회귀(regression) 테스트

16. 스크럼(Scrum)에 대한 설명으로 옳지 않은 것은?

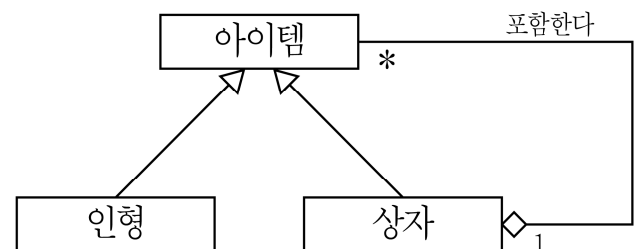
- ① 매일 15분 정도의 짧은 일일 스크럼 회의를 한다.
- ② 스프린트(sprint)는 인덱스 카드를 이용하여 기능을 간략히 서술한 것이다.
- ③ 제품 백로그(product backlog)는 개발할 제품에 대한 요구사항 목록을 포함한다.
- ④ 소프트웨어를 반복적으로 개발하면서 사용자의 요구사항을 지속적으로 반영한다.

17. 다음 자바 코드에 적용된 디자인 패턴은?

```
public class DesignManager {
    private DesignManager() { }
    private static DesignManager _instance = null;
    public synchronized
        static DesignManager Instance() {
        if (_instance == null)
            _instance = new DesignManager();
        return _instance;
    }
}
```

- ① 데코레이터(decorator)
- ② 반복자(iterator)
- ③ 싱글톤(singleton)
- ④ 옵서버(observer)

18. 다음 클래스 다이어그램에 대한 설명으로 옳지 않은 것은?



- ① 인형은 상자를 포함할 수 없다.
- ② 상자와 인형은 아이템의 종류이다.
- ③ 상자는 인형 또는 상자를 하나 이상 포함해야 한다.
- ④ 상자가 인형을 포함한 상자를 포함할 수 있다.

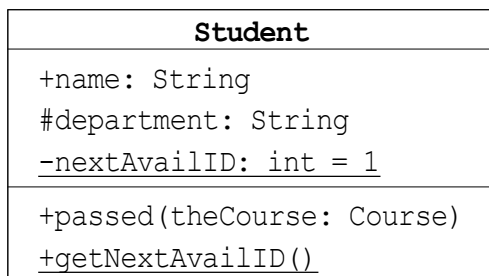
19. 기능 점수(function point)에 대한 설명으로 옳지 않은 것은?

- ① 2022년 SW사업 대가산정 가이드에서 기능 점수 산정의 정통법은 소프트웨어 기능을 도출한 후 각 기능에 평균 복잡도를 적용한다.
- ② 계획 단계뿐 아니라 분석, 설계, 구현 단계에서도 사용할 수 있어 단계가 진행됨에 따라 더 정확한 기능 점수를 측정할 수 있다.
- ③ 구현 기술, 구현 언어, 개발 도구, 개발자의 능력에 상관없이 소프트웨어의 규모 측정에 일관성을 제공한다.
- ④ 소프트웨어의 기능을 정량화해 개발 비용 산정에 활용한다.

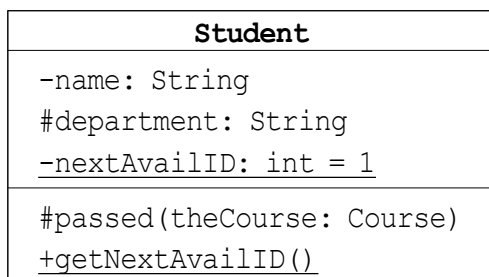
20. 다음 자바 코드를 클래스 다이어그램으로 바르게 표현한 것은?

```
class Student {
    private String name;
    protected String department;
    private static int nextAvailID = 1;
    protected void passed(Course theCourse) {...}
    public static void getNextAvailID() {...}
}
```

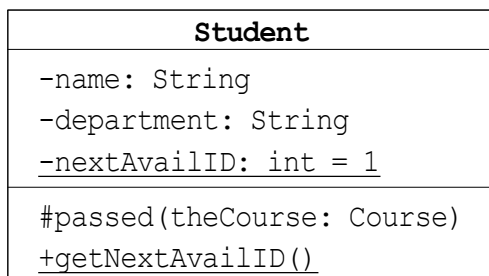
①



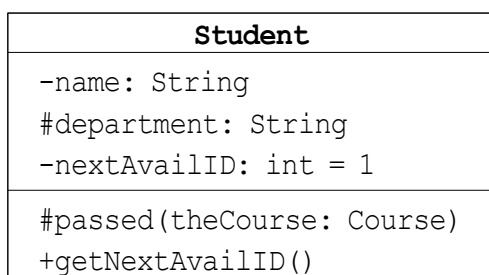
②



③



④



21. 다음 소스 코드에 대해 조건(condition) 커버리지와 분기(branch) 커버리지를 각각 100 % 만족하는 테스트 케이스 집합은? (단, 단축 연산(short-circuit evaluation)은 수행하지 않는다)

```
if (x >= 0 || y < 4)
    z = x + y;
else
    z = x - y;
```

- ① {(x = -1, y = 2), (x = 1, y = 6)}
- ② {(x = -1, y = 2), (x = 2, y = 2)}
- ③ {(x = 1, y = 6), (x = -2, y = 6)}
- ④ {(x = 2, y = 2), (x = -2, y = 6)}

22. 다음 설명에 해당하는 UML 다이어그램은?

- 객체가 수행하는 작업의 흐름을 표현한다.
- 비즈니스 프로세스와 같은 고수준의 흐름을 표현한다.
- 병행 처리를 표현하기 위해 포크와 조인을 이용한다.

- ① 배치(deployment) 다이어그램
- ② 액티비티(activity) 다이어그램
- ③ 컴포넌트(component) 다이어그램
- ④ 패키지(package) 다이어그램

23. 작업 분할 구조도(WBS, Work Breakdown Structure)에 대한 설명으로 옳지 않은 것은?

- ① 프로젝트에서 수행해야 하는 모든 작업을 포함해야 한다.
- ② 작업 프로세스 중심으로 작성할 수도 있고, 산출물 위주로 작성할 수도 있다.
- ③ 작업 패키지(work package)는 계층 구조 상위에 있는 여러 작업을 묶은 관리 단위이다.
- ④ 프로젝트 목표를 달성하고 결과물을 산출하기 위해 수행해야 할 작업을 계층적으로 분할한 것이다.

24. 클래스 B 객체를 생성하면서 전달한 인자의 값을 상속을 이용하여 화면에 출력하기 위한 C++ 코드 중 오류가 있는 코드의 라인 번호만을 모두 고르면? (단, 밑줄 친 라인 번호의 코드 이외에는 오류가 없다)

라인 번호	C++ 코드
1	#include<iostream>
2	using namespace std;
3	
4	class A {
5	private:
6	int size;
7	public:
8	A(int size) { size = size; }; //오류 확인
9	virtual void print() { cout << size; };
10	};
11	
12	class B : public A {
13	public:
14	B(int size) : A(size) { };
15	void print() { cout << size; }; //오류 확인
16	};
17	
18	int main() {
19	A *b = new B(10);
20	b.print(); //오류 확인
21	return 0;
22	}

- ① 15
- ② 8, 15
- ③ 8, 20
- ④ 8, 15, 20

25. 다음은 프로젝트 P의 계획 단계에서 수립한 작업, 소요기간, 각 작업의 선행작업이다. 프로젝트 수행 중 개발자들이 독감에 걸려 작업 T5와 T7이 각각 7일, 5일이 지연되어 프로젝트 최소 완료 시간이 40일로 되었다면 (가)에 들어갈 작업 소요기간(일)은? (단, T5와 T7을 제외한 다른 작업은 계획대로 수행되었다)

작업	소요기간(일)	선행작업
T1	7	—
T2	8	T1
T3	3	T1
T4	5	T2, T3
T5	6	T3
T6	10	T5
T7	(가)	T4
T8	5	T6, T7

- ① 8
- ② 10
- ③ 13
- ④ 15